UNIVERSIDADE FEDERAL DO CEARÁ

CENTRO DE TECNOLOGIA

PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE TELEINFORMÁTICA

DOUTORADO EM ENGENHARIA DE TELEINFORMÁTICA

JOSÉ MARIA PIRES DE MENEZES JÚNIOR

CONTRIBUIÇÕES AO PROBLEMA DE PREDIÇÃO RECURSIVA DE SÉRIES TEMPORAIS UNIVARIADAS USANDO REDES NEURAIS RECORRENTES

FORTALEZA

2012

JOSÉ MARIA PIRES DE MENEZES JÚNIOR

CONTRIBUIÇÕES AO PROBLEMA DE PREDIÇÃO RECURSIVA DE SÉRIES TEMPORAIS UNIVARIADAS USANDO REDES NEURAIS RECORRENTES

Tese apresentada ao Curso de Doutorado em Engenharia de Teleinformática do Programa de Pós-Graduação em Engenharia de Teleinformática do Centro de Tecnologia da Universidade Federal do Ceará, como requisito parcial à obtenção do título de doutor em Engenharia de Teleinformática. Área de Concentração: Sinais e Sistema

Orientador: Prof. Dr. Guilherme de Alencar Barreto

FORTALEZA

2012

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Biblioteca Universitária
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

M511c Menezes Júnior, José Maria Pires de.
Contribuições ao Problema de Predição Recursiva de Séries Temporais Univariadas Usando Redes Neurais Recorrentes / José Maria Pires de Menezes Júnior. – 2012.
188 f. : il. color.

Tese (doutorado) – Universidade Federal do Ceará, Centro de Tecnologia, Programa de Pós-Graduação em Engenharia de Teleinformática, Fortaleza, 2012.
Orientação: Prof. Dr. Guilherme de Alencar Barreto.

1. Séries Temporais. 2. Predição Recursiva. 3. Redes Neurais Recorrentes. 4. Rede NARX. 5. Máquina de Aprendizado Extremo. I. Título.

CDD 621.38

JOSÉ MARIA PIRES DE MENEZES JÚNIOR

CONTRIBUIÇÕES AO PROBLEMA DE PREDIÇÃO RECURSIVA DE SÉRIES TEMPORAIS UNIVARIADAS USANDO REDES NEURAIS RECORRENTES

Tese submetida à Coordenação do Curso de Pós-Graduação em Engenharia de Teleinformática, da Universidade Federal do Ceará, como requisito parcial para obtenção de grau de Doutor em Engenharia de Teleinformática, área de concentração Sinais e Sistemas.

Aprovada em: 02 / 03 / 2012.

BANCA EXAMINADORA

_____
Prof. Dr. Guilherme de Alencar Barreto (Orientador)
Universidade Federal do Ceará (UFC)

_____
Prof. Dr. Giovanni Cordeiro Barroso
Universidade Federal do Ceará (UFC)

_____
Prof. Dr. Arthur Plinio de Souza Braga
Universidade Federal do Ceará (UFC)

_____
Prof. Dr. José Manoel de Seixas
Universidade Federal do Rio de Janeiro (UFRJ)

_____
Prof. Dr. André Ponce de Leon Ferreira de Carvalho
Universidade de São Paulo (USP)

Dedico este trabalho aos meus pais José Maria e Rosa Virgínia pelo constante apoio, incentivo e admiração.

AGRADECIMENTOS

A Deus, acima de tudo.

Ao meu orientador, Prof. Guilherme de Alencar Barreto, a quem sou grato pela orientação, paciência e confiança depositada.

Aos meus irmãos, pela ajuda em todas as horas.

Aos colegas de laboratório, por estarem sempre prontos a ajudar, proporcionando excelente ambiente de trabalho.

Aos professores e funcionários do Departamento de Engenharia de Teleinformática que de forma direta ou indireta participaram do desenvolvimento deste trabalho.

Aos colegas professores da Universidade Federal do Piauí, pela confiança e apoio que proporcionaram durante os últimos anos desta jornada.

À FUNCAP (Fundação Cearense de Amparo à Pesquisa) pelo suporte financeiro.

Em especial à Ana Valéria, minha noiva, pelo amor, carinho, incentivo, admiração e apoio incondicional.

“O melhor profeta do futuro é o passado.”

(Lord Byron)

RESUMO

Nesta tese aborda-se o problema de predição recursiva de séries temporais univariadas, também chamado de predição de longo prazo, usando redes neurais recorrentes. Este tipo de problema surge, com frequência, em tarefas de modelagem e predição de sistemas dinâmicos não-lineares, principalmente os que produzem sinais de natureza caótica, em que se observa a presença de dependência temporal (memória) de longa duração. Na predição recursiva, diferentemente da predição de um passo à frente (*one-step-ahead prediction*), as predições são realimentadas para a entrada do modelo neural, característica esta que dificulta a predição de séries com dependência temporal longa devido à propagação do erro de predição. Isto posto, para tratar o problema de predição recursiva de séries temporais, extensões do modelo neural NARX (*Nonlinear AutoRegressive model with eXogenous inputs*) são propostas nesta tese. Estas extensões resultam da tentativa de incorporar à rede NARX diferentes estratégias de modelagem da informação temporal, tanto de curto quanto de longo prazo. Dentre estas estratégias, destacam-se: (i) predição (simultânea) de vários passos à frente, também chamada de predição MIMO (multi-input, multi-output model), (ii) predição via projeções aleatórias dinâmicas, tal como na rede ESN (*echo state network*), (iii) predição via projeções aleatórias estáticas, tal como na rede ELM (*extreme learning machine*), e (iv) predição via modelos recorrentes híbridos baseados nas redes NARX e ELMAN. Além disso, uma metodologia para projeto (i.e. seleção de parâmetros) e comparação dos desempenhos dos modelos propostos é também desenvolvida nesta tese com o objetivo de avaliá-los sob as mesmas condições e servir de referência para estudos futuros. Para este fim, são utilizadas séries temporais sintéticas e reais comumente presentes em *benchmarks* de desempenho. Os resultados obtidos sugerem que os modelos propostos apresentam-se como alternativas eficientes ao estado da arte em modelos de redes neurais recorrentes para predição de séries temporais univariadas, principalmente aqueles baseados em projeções aleatórias devido ao baixo custo computacional.

Palavras-chave: Séries Temporais, Predição Recursiva, Redes Neurais Recorrentes, Rede NARX, Rede de Ecos de Estado, Máquina de Aprendizado Extremo, Rede de Elman.

ABSTRACT

In this thesis, we tackle the problem of recursive prediction of univariate time series, also known as long-term prediction, using recurrent neural networks. This type of problem often emerges from nonlinear dynamical systems modelling and prediction tasks, particularly from those producing signals of chaotic nature, where one can observe the presence of long-term temporal dependencies. In recursive prediction, differently from the one-step-ahead prediction task, predicted values are fed back to the input of the neural model, a feature that makes time series with long-term temporal dependencies more difficult to deal with due to the propagation of prediction errors. That being said, in order to handle the problem of recursive prediction of univariate time series, extensions of the neural NARX (Nonlinear AutoRegressive model with eXogenous inputs) model ar eintroduced in this thesis. These extensions result from attempts to embed into the NARX model different strategies to capture temporal information, either of short-term or long-term nature. Among such strategies, we highlight the following ones: (i) simultaneous prediction of several steps ahead, also known as MIMO (multi-input, multi-output model) prediction, (ii) prediction via dynamical random projections, as in the ESN (echo state network) model, (iii) prediction via static random projections, as in the ELM (extreme learning machine) network, and (iv) prediction via hybrid recurrent models based the NARX and ELMAN networks. Additionally, a novel methodology for the design (i.e. parameter selection) and performance comparison of the proposed models is also introduced in this model with the aim of evaluating them under similar conditions and to serve as reference for further studies. For this purpose, synthetic and real-world benchmarking time series are used. The obtained results suggest that the proposed neural models present themselves as efficient alternatives to the state of the art in recursive prediction of univariate time series using recurrent neural architectures.

Keywords: Time Series, Recursive Prediction, Recurrent Neural Networks, NARX Network, Echo State Network, Extreme Machine Learning, Elman network.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

RNAs	Redes Neurais Artificiais
RNRs	Redes Neurais Recorrentes
NAR	*Nonlinear AutoRegressive*
AR	Auto Regressivos
MA	Média Móvel
ARIMA	Auto Regressivos Integrado de Médias Móveis
FAC	Função de Autocorrelação
UPA	Um-Passo-Adiante
HPA	H-Passos-Adiante
MIMO	*Multiple-Input and Multiple-Output*
MLP	*MultiLayer Perceptron*
FTDNN	*Focused Time Delay Neural Network*
FUNCEME	Fundação Cearense de Metoorologia e Recursos Hídricos
MPE	*Mean Prediction Error*
MSE	*Mean-Squared Error*
NMSE	*Normalized Mean-Squared Error*
NARX	*Nonlinear AutoRegressive model with eXogenous inputs*
NARX-P	NARX com Modo Paralelo
NARX-SP	NARX com Modo Série Paralelo
NARX-MISO	NARX com Múltiplas Entradas e Simples Saída
NARX-MIMO	NARX com Múltiplas Entradas e Múltiplas Saídas
ESN	*Echo State Network*
ELM	*Extreme Machine Learning*
NARX-ELM	Rede NARX utilizando o algoritmo de treinamento da rede ELM
ELMAN-ELM	Rede de Elman utilizando o algoritmo de treinamento da rede ELM
ELMAN/NARX-ELM	Rede Híbrida de Elman com NARX utilizando o algoritmo de treinamento da rede ELM

LISTA DE SÍMBOLOS

f	função matemática
$\mathbb{R}$	conjunto dos números reais
x	variável escalar, i.e., $x \in \mathbb{R}$
t	índice indicativo de tempo contínuo
n	índice indicativo de tempo discreto
$\triangle t$	diferença de tempo
ξ	parâmetro do mapa logístico
τ	atraso de imersão
d_E	dimensão de imersão
$\widehat{x}$	variável escalar predita
d	dimensão do atrator
ρ	distância entre dois vetores
ε^2	erro médio quadrático
σ^2	variância amostral
$\mathbf{x}$	variável vetorial, i.e., $\mathbf{x} \in \mathbb{R}^n$
$\mathbf{w}_i$	vetor de pesos associados ao neurônio i em uma rede neural
$\mathbf{m}_k$	vetor de pesos associados ao neurônio k em uma rede neural
η	taxa de aprendizagem das redes neurais artificiais
$\mathbf{X}$	matriz de dados
$\mathbf{d}$	vetor de saídas desejadas para redes supervisionadas
w_{ij}	peso associado à ligação entre entrada j e neurônio i da camada intermediária de uma rede supervisionada
m_{ki}	peso associado à ligação entre neurônio i da camada intermediária e saída k de uma rede supervisionada
ε_{med}	erro quadrático médio por época de treinamento da rede supervisionadas
$\phi(\cdot)$	função de ativação de um neurônio de rede supervisionada
θ_i	limiar de ativação de um neurônio i de rede supervisionada
u_i	ativação do neurônio i da camada oculta de uma rede supervisionada
y_i	saída do neurônio i na camada de saída de uma rede supervisionada
δ_k	gradiente local do neurônio k em uma rede neural
δ_i	gradiente local do neurônio i em uma rede neural
p	dimensão do vetor de entrada de uma rede neural

m	dimensão do vetor de saída de uma rede neural
q_1	número de neurônios da primeira camada oculta
q_2	número de neurônios da primeira camada oculta
M	número de parâmetros ajustáveis numa rede neural
C	unidade de contexto em uma rede neural recorrente
d_u	ordem de memória de entrada de uma modelo NARX
d_y	ordem de memória de saída de uma modelo NARX

SUMÁRIO

1 INTRODUÇÃO

A antecipação do comportamento futuro dos acontecimentos sempre despertou interesse nas mais diversas áreas do conhecimento humano. Para que seja possível prever os valores futuros, com base em valores passados, é necessário que se disponha de dados históricos do fenômeno de interesse. Todavia, o conjunto de dados, por si só, não permite a predição dos valores futuros, sendo necessário para isso a utilização de algoritmos, técnicas ou métodos de predição de séries temporais, que podem envolver cálculos relativamente simples ou procedimentos de elevada complexidade.

Estudos empíricos, inicialmente em climatologia e hidrologia, como o trabalho pioneiro de Hurst (1951), revelaram a presença de memória longa em dados de séries temporais. Formalmente, dependência de longa duração é uma propriedade observada em certos fenômenos nos quais a correlação entre as observações diminui muito lentamente com a separação (temporal) entre elas. No contexto da análise de séries temporais, esta propriedade é caracterizada pelo lento decaimento da função de autocorrelação. Outra característica deste tipo de série é que sua função de densidade espectral é não limitada na frequência zero, o que equivale dizer que sua função de autocorrelação não é absolutamente somável (MORETTIN; TOLOI, 2004; CORREIA, 1997).

Antes do advento da teoria do caos e da geometria fractal, por volta da década de 1960, o comportamento irregular observado em certos sistemas determinísticos não-lineares era tipicamente modelado como estocástico, isto é, tal comportamento era definido como aleatório e imprevisível (KUGIUMTZIS; LILLEKJENDLIE; CHRISTOPHERSEN, 1994). Em outras palavras, tal comportamento irregular era atribuído a alguma entrada aleatória, externa ao sistema. Segundo uma das premissas da teoria do caos, entradas aleatórias deixaram de ser a única fonte possível de irregularidades em um sistema. Sistemas não-lineares caóticos podem também gerar sinais que se assemelham a sinais estocásticos, mas que foram gerados, contudo, por equações puramente determinísticas. Desta forma, técnicas lineares convencionais têm cedido cada vez mais espaço para técnicas não-lineares que conseguem capturar, com mais eficiência, a dinâmica de sistemas complexos.

Devido ao aprendizado de natureza indutiva, Redes Neurais Artificiais (RNAs) podem ser usadas para inferir relações não-lineares complexas a partir das observações de uma série temporal. Dessa forma, RNAs têm sido utilizadas com sucesso em problemas de predição e modelagem de séries temporais de dinâmica complexa, tais como predição de séries temporais financeiras (DABLEMONT *et al.*, 2003; GURESEN; KAYAKUTLUA; DAIM, 2011), predição

de vazão de rios (ATIYA *et al.*, 1999; TAMPELINI *et al.*, 2011), modelagem de séries temporais biomédicas (COYLE; PRASAD; MCGINNITY, 2005) e predição de tráfego de rede (DOULAMIS; DOULAMIS; KOLLIAS, 2003; ATIYA; ALY; PARLOS, 2005), para mencionar apenas algumas destas aplicações. Geralmente, modelos de RNA têm melhor desempenho que as técnicas lineares tradicionais, tais como os modelos Box-Jenkins (BOX; JENKINS; REINSEL, 1994), quando as séries temporais possuem forte componente não-linear. Neste caso, as habilidades de generalização e aproximação universal de funções de RNA justificam seu melhor desempenho preditivo.

Em predição de séries temporais, normalmente são usados modelos de RNAs *feedforward*, treinados pelo algoritmo *backpropagation*, para serem usadas como preditores de um-passo-adiante (*one-step-ahead prediction*). Nestes casos, a rede é alimentada com valores passados da série até o instante atual e a sua saída estima o próximo valor da série temporal. No próximo instante de tempo, o valor predito não é realimentado para a entrada da rede, sendo usado o valor observado em vez do predito.

Para predição de horizonte mais amplo, lança-se mão da predição múltiplos-passos-adiante (*multistep-ahead prediction*), também conhecida como predição recursiva ou predição iterada, em que a saída do modelo deve ser realimentada para a entrada até atingir o instante futuro desejado. A tarefa de predição múltiplos-passos-adiante é mais difícil de lidar do que a predição de um único passo, devido ao problema da propagação dos erros de predição (SORJAMAA *et al.*, 2007).

Mais recentemente, a aplicação de redes neurais recorrentes (RNRs) ao problema de predição múltiplos-passos-adiante tem sido alvo de vários estudos (GÓMEZ-GIL *et al.*, 2011; ARDALANI-FARSA; ZOLFAGHARI, 2010; GRAVES; PEDRYCZ, 2009; CHTOUROU; CHTOUROU; HAMMAMI, 2008; MENEZES-JÚNIOR; BARRETO, 2008a), que demonstram seu potencial preditivo. Tais estudos mostram que redes recorrentes apresentam melhores desempenhos que redes neurais *feedforward* convencionais. Contudo, o uso de RNRs em predição recursiva é bem menos comum que o uso de RNAs *feedforward*, pois com a presença de laços de realimentação, o problema da propagação dos erros de predição é amplificado tornando o treinamento e o teste daquele tipo de rede bem mais complexo (instável). Além disso, os algoritmos convencionais de treinamento de RNRs, tais como *backpropagation through time* (WERBOS, 1990), real-time recurrent learning (WILLIAMS; ZIPSER, 1989) e *extended Kalman filtering* (PUSKORIUS; FELDKAMP, 1994), têm elevado custo computacional para

redes com muitos neurônios.

Um novo paradigma de projeto de redes neurais vem sendo proposto sob a alcunha de Projeções Aleatórias (*Random Projections*) (MICHE; SCHRAUWEN; LENDASSE, 2010), onde os pesos dos neurônios da camada oculta são iniciados de forma aleatória e mantidos fixos, enquanto apenas os pesos da camada de saída são ajustáveis. Uma vantagem imediata deste tipo de RNA com relação às arquiteturas de redes convencionais está na rapidez do treinamento. Dentro do arcabouço geral das redes neurais baseadas em projeções aleatórias destacam-se as arquiteturas ESN (*Echo State Network*) (JAEGER; HAAS, 2004) e ELM (*Extreme Learning Machine*) (HUANG; WANG; LAN, 2011), sendo a primeira uma arquitetura recorrente e a segunda uma arquitetura *feedforward*.

Embora algumas aplicações das redes ELM e ESN em predição de séries temporais já possam ser encontradas na literatura (SHENG *et al.*, 2012; SOVILJ *et al.*, 2010; SINGH; BALASUNDARAM, 2007), ainda há muito espaço para contribuições, principalmente no que se refere à tarefa de predição recursiva. Esta constatação serve como uma das principais motivações para o desenvolvimento deste trabalho de pesquisa, cujos objetivos serão discutidos a seguir.

1.1 Objetivos Geral e Específicos

O problema-alvo desta tese é o problema de predição recursiva de séries temporais univariadas usando redes neurais recorrentes. Isto posto, o objetivo geral deste trabalho é propor extensões do modelo neural NARX (*Nonlinear AutoRegressive model with eXogenous inputs*) (MENEZES-JÚNIOR; BARRETO, 2008a), tal que várias das limitações das redes recorrentes convencionais (e.g. elevado custo de treinamento), quando aplicadas ao problema de predição recursiva, sejam diminuídas. Estas extensões resultam da tentativa de incorporar à rede NARX diferentes estratégias de modelagem da informação temporal, tanto de curto quanto de longo prazo.

Em face do objetivo geral apresentado no parágrafo anterior, vários objetivos específicos foram sendo colocados ao longo do desenvolvimento da tese, de tal modo a permitir que o objetivo geral fosse cumprido. Tais objetivos específicos são listados a seguir.

- Desenvolver e avaliar uma extensão da rede NARX para predição (simultânea) de vários passos à frente, também chamada de predição MIMO (multi-input, multi-output model).
- Desenvolver e avaliar uma extensão da rede NARX para predição via projeções aleatórias estáticas, tal como na rede ELM (*extreme learning machine*).

- Desenvolver e avaliar uma extensão da rede NARX para predição via projeções aleatórias dinâmicas, tal como na rede ESN (*echo state network*).
- Desenvolver e avaliar uma extensão da rede NARX via modelos recorrentes híbridos baseados nas redes NARX e ELMAN.
- Desenvolver uma metodologia geral para projeto (i.e. seleção de parâmetros) e comparação dos desempenhos dos modelos propostos com o objetivo de avaliá-los sob as mesmas condições e servir de referência para estudos futuros.

1.2 Histórico de Desenvolvimento da Tese

Em 2006, foram gerados os primeiros resultados da avaliação de modelos neurais na tarefa de predição recursiva de séries temporais não-lineares, cujo o principal resultado foi a proposição da rede NARX-MISO (MENEZES-JÚNIOR, 2006). Para a avaliação deste modelo foram utilizadas algumas séries temporais caóticas e séries de tráfego de redes de computadores. Embora tenha conseguido resultados promissores, nesta fase não foi possível avaliar de modo abrangente e sistemático todo o poder computacional da rede NARX-MISO.

Com o intuito de aprofundar as investigações sobre o desempenho da rede NARX-MISO na predição recursiva de séries temporais, a presente pesquisa de doutorado teve seu início em 2007. Os primeiros esforços se deram na tentativa de definir uma metodologia para projeto (i.e. seleção de parâmetros) e de comparação dos desempenhos das arquiteturas neurais de interesse ao problema, entre elas a rede NARX, com o objetivo de avaliá-las sob as mesmas condições e servir de referência para estudos futuros. Esta fase fez-se necessária porque não haviam sido encontrados estudos sistemáticos na literatura especializada.

A segunda dificuldade encontrada ao longo do desenvolvimento desta tese foi a de que as tarefas de predição de séries temporais caóticas geralmente requisitam bastante processamento computacional, principalmente durante o treinamento do modelo NARX-MISO, o que geralmente envolve muitas épocas de treinamento e várias repetições do processo de busca por parâmetros ótimos. Para contornar tal dificuldade buscou-se utilizar linguagens de programação que utilizem melhor o tempo do processador.

Durante o trabalho de mestrado todas os algoritmos tinham sido desenvolvidos em ambientes de programação tais como Matlab© e Octave, principalmente escolhidos pela facilidade de implementação dos códigos. Apesar desta vantagem, estes ambientes de computação científica ainda são lentos quando comparados com linguagens de programação compiladas, tais

como C/C++ e Java. Desta forma, passou-se a migrar as RNAs desenvolvidas em Matlab para a linguagem C++. Para este fim, utilizou-se a biblioteca matemática e de processamento de sinais conhecida como IT++[1].

Embora esta mudança tenha melhorado os desempenhos das RNAs no quesito custo computacional, procurou-se acelerar ainda mais o processo utilizando processamento paralelo distribuído, onde vários computadores são utilizados para realizar uma determinada tarefa. Assim, através do acesso remoto de computadores, as tarefas são distribuídas para vários processadores, obtendo agilidade e ganho de tempo na realização da tarefa de predição de séries temporais. Para apoio deste processo foi utilizada a linguagem *shell script*.

Os resultados da predição com a utilização da metodologia de busca por parâmetros ótimos, utilizando a linguagem C++ e com o processamento paralelo culminaram com a publicação de um artigo no II European Symposium on Time Series Prediction (ETSP'2008) (MENEZES-JÚNIOR; BARRETO, 2008b). Esta proposta é discutida em maior nível de detalhes no Capítulo 4 e exemplificada com a utilização de uma série de precipitação de chuvas da cidade de Fortaleza para um horizonte de predição de 12 meses à frente.

De posse de uma metodologia sistemática de busca de parâmetros ótimos bem definida, partiu-se para a busca por novas arquiteturas neurais para predição recursiva. O primeiro resultado relevante foi obtido com um estudo de comparação de desempenho entre as redes NARX-MISO e ESN, além de diversas outras técnicas de Aprendizado de Máquinas existentes na literatura. Neste estudo foi utilizada a série do Laser Caótico, e o resultado foi publicado no IX Congresso Brasileiro de Redes Neurais (CBRN'2009) (MENEZES-JÚNIOR; BARRETO; FREIRE, 2009). Este estudo comparativo culminou no desenvolvimento de um modelo neural NARX implementado a partir de uma rede ESN. Este modelo será discutido em maior profundidade posteriormente nesta tese. Em seguida, inspirado pelos resultados do modelo híbrido NARX-ESN, buscou-se investigar também uma implementação do modelo NARX a partir de uma rede ELM, culminando numa versão extremamente rápida da rede NARX-MISO. Estas propostas serão discutidas no Capítulo 6.

Como forma de tentar melhorar ainda mais o desempenho da rede NARX-MISO foi proposta uma rede NARX com múltiplas saídas. Esta arquitetura recebeu o nome de NARX-MIMO, sendo que o termo MIMO (*multi-input, multi-output*) refere-se ao fato de a rede ter que prever várias saídas por instante de tempo. Esta idéia é melhor detalhada no Capítulo 5.

[1] Website: *http://sourceforge.net/apps/wordpress/itpp/*

Por fim, suspeitando das possíveis potencialidades da rede recorrente de Elman, foram investigadas algumas estratégias que pudessem tirar maior proveito dos laços de realimentação desta rede. Para este fim, foram propostas variantes da rede de Elman, dentre elas uma que envolve a utilização da derivada da saída da ativação dos neurônios ocultos, e outra que culminou na proposição de um modelo híbrido envolvendo as redes ELMAN, NARX e ELM. Os resultados deste trabalho foram publicados na X Conferência Brasileira de Dinâmica, Controle e Aplicações (DINCON'2011) (MENEZES-JÚNIOR; BARRETO, 2011), e serão discutidos com mais profundidade no Capítulo 7.

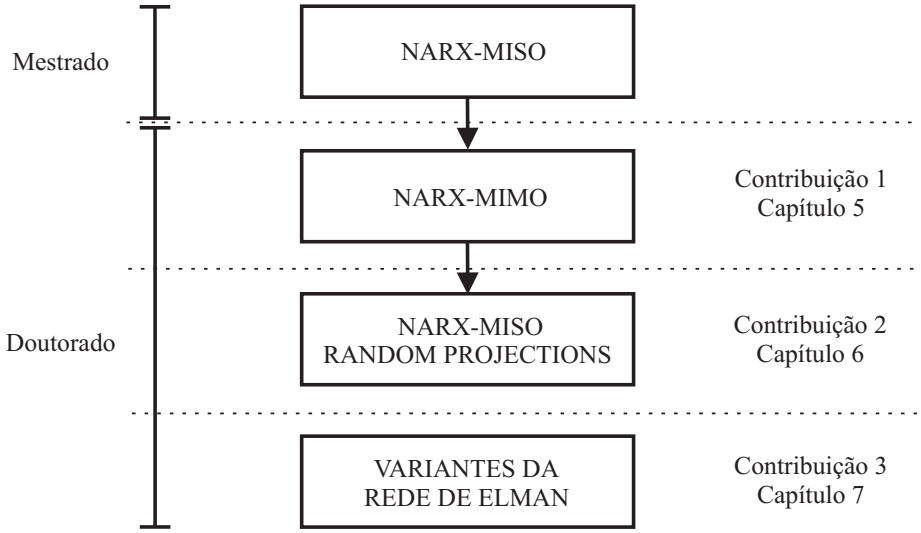


Figura 1 – Ilustração das linhas de investigação desenvolvidas nesta tese.

A Figura 1 traz um resumo das linhas de investigação seguidas nesta tese e as principais contribuições resultantes.

1.3 Produção Científica

Embora algumas publicações resultantes desta tese já tenham sido mencionadas na seção anterior, uma lista completa dos trabalhos publicados ou que se encontram submetidos é fornecida a seguir.

- **MENEZES JÚNIOR, J. M. P.** & BARRETO, G. A. - Long-Term Prediction of Chaotic Time Series using Random Projections Neural Networks: An Empirical Analysis. Submetido ao periódico *Chaos, Solitons and Fractals*.
- **MENEZES JÚNIOR, J. M. P.** & BARRETO, G. A. - Extensões da Rede Recorrente de Elman para Predição Não-linear de Séries Temporais Caóticas: Um Estudo Comparativo,

Anais da X Conferência Brasileira de Dinâmica, Controle e Aplicações (DINCON'2011), Águas de Lindóia-SP, 2011.

- **MENEZES JÚNIOR, J. M. P.**, BARRETO, G. A. & FREIRE, A. L. - Redes Neurais Recorrentes para Predição Recursiva de Séries Temporais Caóticas: Um Estudo Comparativo, *Anais do IX Congresso Brasileiro de Redes Neurais Artificiais* (CBRN'2009), Ouro Preto-MG, 2009.
- **MENEZES JÚNIOR, J. M. P.** & BARRETO, G. A. - Long-Term Time Series Prediction with the NARX Network: An Empirical Evaluation. *Neurocomputing (Amsterdam)*, v. 71, n. 16-18, p. 3335-3343, 2008.
- **MENEZES JÚNIOR, J. M. P.** & BARRETO, G. A. - Multistep-ahead prediction of rainfall precipitation using the NARX network, *Proceedings of the 2nd European Symposium on Time Series Prediction* (TSP'2008), Porvoo, Finlândia, v. 1, p. 87-96, 2008.

1.4 Estrutura da Tese

1.4.1 Metodologia de Organização

Esta tese é subdividida em capítulos com base nas diversas estratégias de predição de séries temporais. Os capítulos são organizados de forma a serem o mais auto-contidos possível em termos de conteúdo. Esta organização permite descrever mais especificamente a origem, as motivações e a fundamentação teórica para cada uma das estratégias de predição. Além disto, a organização proposta visa permitir que as contribuições teóricas desta tese sejam mais adequadamente apresentadas e inseridas no contexto de outras abordagens contidas na literatura.

A forma como a tese está organizada permite que o leitor interessado em uma determinada estratégia se restrinja a um determinado capítulo de interesse. Os detalhes contidos em cada capítulo são descritos de forma resumida na subseção a seguir.

1.4.2 Organização Geral do Restante do Projeto

O restante deste trabalho está organizado em onze capítulos que descrevem os diversos conceitos relacionados à predição recursiva de séries temporais utilizando RNAs. Um breve comentário sobre cada um deles é feito a seguir.

No Capítulo 2 é apresentado um breve resumo dos fundamentos do problema de predição de séries temporais, além da classificação e descrição de alguns métodos de predição de

séries temporais. Também são feitas considerações sobre o problema de reconstrução do espaço de estados de um sistema dinâmico.

O Capítulo 3 tem por objetivo apresentar as arquiteturas convencionais de redes neurais aplicadas aos problemas de predição e modelagem de séries temporais não-lineares. As arquiteturas de redes neurais descritas neste capítulo são todas de aprendizado supervisionado, diferenciando umas das outras apenas pelo modo que processam informação temporal, ou seja, se utilizam ou não laços de realimentação (*feedback loops*). Desta forma, arquiteturas supervisionadas que não contenham tais laços, comumente chamadas de *não-recorrentes* ou *feedforward*, são discutidas em primeiro lugar. Em seguida, arquiteturas contendo laços de realimentação, doravante chamadas de *redes neurais recorrentes*, são apresentadas.

O Capítulo 4 apresenta a metodologia empregada para obtenção dos modelos de predição de séries temporais. São abordados os principais índices de desempenho utilizados para quantificar o quão satisfatório são os resultados das predições dos modelos. Em especial, dar-se-á ênfase na predição de recursiva. São também descritas as séries temporais de precipitação de chuvas, a série caótica de Hénon, a série do Laser Caótico e a série de Mackey-Glass. Logo em seguida é proposto um método para obtenção do modelo mais adequado de uma RNA para um certo conjunto de dados de interesse.

No Capítulo 5 é dada ênfase à descrição do modelo NARX-MISO e sua extensão para o modelo NARX-MIMO, que se diferenciam basicamente pela quantidade de saídas que se pode prever por instante de tempo. O objetivo deste estudo é ressaltar as diferenças no projeto de tais arquiteturas neurais recorrentes, visando oferecer subsídios ao usuário no momento da escolha da arquitetura mais adequada à tarefa de interesse.

No Capítulo 6 são apresentadas as redes baseadas em projeções aleatórias. São introduzidas as ideias por trás dos principais métodos que fazem uso de projeções aleatórias: Computação de Reservatório e Máquina de Aprendizado Extremo, tendo por objetivo apresentar as principais diferenças entre estas duas abordagens, entender suas estruturas e investigar suas potencialidades. Por fim, são propostas variantes da rede NARX que fazem uso do conceito de projeções aleatórias estáticas e dinâmicas.

No Capítulo 7 discute-se propostas de extensões da rede de Elman para melhorar o desempenho desta rede recorrente clássica na tarefa de predição múltiplos-passos-adiante. Dentre as modificações propostas estão o uso de duas camadas ocultas, a realimentação ou da ativação ou da derivada da ativação de uma das camadas ocultas, e a implementação da rede de

Elman usando o algoritmo de treinamento da rede ELM.

Nos Capítulos 8, 9 e 10 são reportados os resultados das simulações computacionais das redes introduzidas nesta tese, em ordem de apresentação.

No Capítulo 11 são feitas as considerações finais, comentários e análise dos resultados obtidos nesta tese. São analisadas as contribuições propostas, bem como são sugeridos trabalhos futuros relacionados com o tema abordado.

2 PREDIÇÃO DE SÉRIES TEMPORAIS

2.1 Introdução

Este capítulo é dedicado à definição de séries temporais univariadas e do problema de predição de valores futuros de tais séries. Também são feitas considerações sobre a reconstrução do espaço de estados.

2.2 Séries Temporais Univariadas

O conceito de séries temporais univariadas está relacionado a um conjunto de observações de uma determinada variável ordenadas sequencialmente ao longo do tempo. Explicado de outra forma, uma série temporal é simplesmente uma sequência de números coletados em intervalos regulares durante um período de tempo. Em séries temporais a ordem dos dados é fundamental, diferentemente do que ocorre em modelos de regressão linear, onde a ordem das observações é irrelevante para a análise. As séries temporais existem nas mais variadas áreas de aplicação, como por exemplo em:

- Economia: preços de ações, taxa de desemprego, taxa de juros.
- Medicina: níveis de eletrocardiograma, eletroencefalograma.
- Epidemiologia: casos de sarampo, AIDS e dengue.
- Meteorologia: temperatura diária, precipitação de chuvas, velocidade dos ventos.
- Energia: consumo de energia elétrica, vazão de reservatórios de usinas hidroelétricas.

Os objetivos de se analisar uma série temporal se concentram na necessidade de descrever propriedades da série como, por exemplo, o padrão de tendência, periodicidade ou a existência de alterações estruturais; no desejo de explicar o comportamento da série através da construção de modelos; no controle de processos, por exemplo, controle estatístico de qualidade; e por fim na predição de valores futuros com base em valores passados, com o intuito de fazer planos a longo, médio e curto prazo e, porventura, na tomada de decisões apropriadas.

Essencialmente as séries temporais podem ser classificadas em três tipos: determinísticas, estocásticas ou caóticas. Em séries temporais determinísticas, os valores futuros podem ser determinados com 100% de certeza e não existe termo aleatório. As séries estocásticas incluem um componente aleatório, fazendo com que o comportamento irregular presente em alguns sistemas seja gerado por um ruído aleatório. Por outro lado, as séries temporais caóticas são

semelhantes às séries temporais estocásticas, mas nas séries caóticas a variabilidade observada não é devido ao ruído, mas sim à influência não-linear entre as variáveis do sistema determinístico subjacente.

Uma série temporal é representada de forma genérica como uma sequência finita de valores de uma certa variável $x \in \mathbb{R}$, $\{x(1), x(2), \ldots, x(N)\}$ ou $\{x(n)\}_{n=1}^{N}$, em que N representa a quantidade de amostras observadas. A predição de séries temporais pode ser definida, de forma sucinta, como a necessidade de encontrar a continuação $\{x(N+1), x(N+2), \ldots\}$ da série.

Na Figura 2 é apresentado um conjunto de observações de uma série temporal coletadas até o instante (N) e, como objetivo, a predição do valor da série até o tempo $(N+4)$. A abordagem padrão de predição envolve a construção de um modelo subjacente que dá origem à sequência observada. Neste sentido, a predição é semelhante à identificação de sistemas. No entanto, na identificação de sistemas, geralmente é assumido que também existam medidas disponíveis de uma entrada exógena de $u(n)$, que aciona o sistema. Além disso, na identificação de sistemas, normalmente se está interessado na construção de modelos que são usados apenas para estimar um único passo de tempo a frente $x(N+1)$. O problema de predição de séries temporais, muitas vezes, implica predições múltiplos passos de tempo a frente.

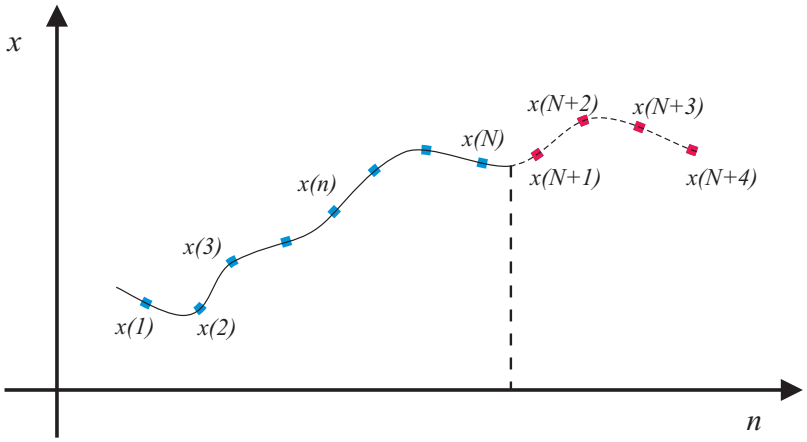


Figura 2 – Observação de uma série temporal e predição.

A determinação das diversas relações de dependência existentes numa série, ou seja, a análise da série, nos leva à escolha de um modelo com o qual obtemos as predições. Os modelos para predição de séries temporais podem ser primeiramente classificados em univariados ou multivariados. Os modelos univariados são baseados numa única série histórica. Já os modelos multivariados são aqueles que envolvem mais de uma série histórica. Os modelos aqui considerados são aqueles que se baseiam na formulação do método de predição somente na informação referente à série temporal em estudo, que são chamados de modelos de predição de

séries temporais univariadas.

2.3 O Problema de Predição de Séries Temporais

Predição de séries temporais pode ser vista como um mapeamento $f : \mathbb{R}^p \rightarrow \mathbb{R}$, a partir do vetor de regressão

$$\mathbf{x}(n) = [x(n) \;\; x(n-1) \;\; \cdots \;\; x(n-p+1)], \tag{2.1}$$

$\mathbf{x}(n) \in \mathbb{R}^p$, para a saída $x(n+1)$. A relação do estado atual $\mathbf{x}(n)$ e o próximo valor da série temporal é dada pela seguinte equação:

$$x(n+1) = f(\mathbf{x}(n)), \tag{2.2}$$

onde $f(\cdot)$ é uma função linear ou não-linear.

Desta forma, o problema de predição se dá em aproximar uma função de mapeamento $f(\cdot)$, fornecendo uma boa aproximação para o próximo valor da série temporal, implementando o seguinte modelo:

$$\widehat{x}(n+1) = \widehat{f}(\mathbf{x}(n)), \tag{2.3}$$

onde $\widehat{x}(n+1)$ é uma estimativa de $x(n+1)$ e $\widehat{f}(\cdot)$ é a aproximação correspondente de $f(\cdot)$. A configuração básica de predição torna-se então

$$x(n+1) = \widehat{f}(\mathbf{x}(n)) + e(n), \tag{2.4}$$

onde o erro de predição, $e(n) = x(n+1) - \widehat{x}(n+1)$, é comumente utilizado para avaliar a qualidade da aproximação.

Morettin e Toloi (2004) descrevem, de uma maneira geral, um ciclo interativo para a construção de um modelo para predição de séries temporais. Tal ciclo é formado por basicamente quatro etapas:

- a primeira etapa do ciclo é a fase de **especificação**, em que uma classe geral de modelos é considerada para análise;
- com base na análise das funções de autocorrelação e autocorrelação parcial define-se a etapa de **identificação** do modelo mais adequado;
- o próximo passo é a etapa de **estimação**, em que os parâmetros do modelo identificado são estimados;

- através de uma série de testes, sendo o principal a análise dos resíduos (erros de predição), o modelo ajustado chega na fase de **validação** ou **diagnóstico**.

Se o modelo não for satisfatório, o ciclo é repetido, voltando-se à fase de identificação. A etapa de identificação é a mais crítica, visto que é possível chegar a uma situação em que vários modelos diferentes se adaptam bem a uma determinada série temporal. O princípio da parcimônia, também conhecido como navalha de Occam (*Occam's razor*), serve como orientação geral nestes casos. Em linhas gerais, este princípio prega que se utilize o modelo mais simples, i.e. com menos parâmetros, caso mais de um modelo explique a série adequadamente.

Existem inúmeros métodos para se realizar predição de séries temporais, desde os métodos mais simples e de fácil entendimento até os mais complexos que envolvem diferentes parâmetros. Desta forma, pode-se definir métodos simples aos métodos que não fazem uso de técnicas de estimação de parâmetros, ou que pouca ou nenhuma análise dos dados é envolvida. O fato de se utilizar métodos estatísticos mais complexos não significa necessariamente uma melhora nos resultados da predição. Métodos simples podem apresentar resultados satisfatórios sobre certas condições, além de permitir uma total compreensão de suas limitações, facilitando a interpretação dos resultados. Deste modo, deve-se primeiro avaliar os benefícios de se utilizar um método simples ou um mais complexo antes de se iniciar a predição em uma determinada aplicação.

Gooijer e Hyndman (2006) revisaram os trabalhos sobre predição de séries temporais publicados entre os anos de 1982 e 2005. Foi elaborado um artigo especial como comemoração aos 25 anos do *International Institute of Forecasters* (IIF). Nesta revisão foram catalogadas técnicas de suavização exponencial, modelos ARIMA, sazonalidade, espaço de estado, modelos não-lineares, modelos de memória longa e modelos ARCH-GARCH. Gooijer e Hyndman (2006) ainda reuniu as vantagens e desvantagens relatadas em cada metodologia e apontou os campos de pesquisa em potencial de crescimento.

2.4 Modelos Matemáticos Simples para Predição

Modelos Matemáticos Simples são construídos para realizar a predição do valor futuro da série temporal pela suavização das observações passadas da série temporal. Assumindo que os valores extremos da série representam flutuações aleatórias, o propósito desses métodos consiste em identificar o padrão básico presente nos dados históricos e, então, usar esse padrão para prever valores futuros. Morettin e Toloi (2004) associa a grande popularidade desses

métodos à simplicidade, à eficiência computacional e à razoável predição obtida. Entre os métodos simples de predição destacam-se o da Média Móvel, o da Suavização Exponencial Simples, o da Suavização Exponencial Linear e o da Suavização Exponencial Sazonal, os quais são apresentados sucintamente na sequência.

2.4.1 Média Móvel

A técnica de Média Móvel consiste em calcular a média aritmética das observações mais recentes, obtida por meio da seguinte operação

$$\widehat{x}(n+1) = \frac{x(n)+x(n-1)+\ldots+x(n-r+1)}{r}, \tag{2.5}$$

onde r é o único parâmetro para este método e representa o número de observações mais recentes da série temporal. Desta forma, este método considera como predição para o período futuro somente a média das observações passadas mais recentes.

O termo média móvel é utilizado porque à medida que a próxima observação se torna disponível, uma nova observação é incluída no conjunto de observações, desprezando-se a observação mais antiga e assim calculando-se uma nova média de observações.

Deve-se observar a relação do método com o valor de r. Se for utilizado um valor grande para este parâmetro, a predição acompanha lentamente as mudanças ocorridas na série, já para um valor pequeno implica numa reação mais rápida da predição em relação aos valores passados. Analisando os valores extremos verifica-se que:

- se $r = 1$, então o valor mais recente da série é utilizado como predição de todos os valores futuros. Este é o tipo de predição mais simples que existe e pode ser denominado de "método ingênuo";
- se $r = N$, então a predição será igual à média aritmética de todos os dados observados, o que é denominado de "média histórica".

2.4.2 Suavização Exponencial Simples

O método de Suavização Exponencial se assemelha ao da Média Móvel por extrair das observações da série temporal o comportamento aleatório pela suavização dos dados recentes. No método de Média Móvel as observações usadas para encontrar a predição do valor futuro contribuem em igual proporção para o cálculo da predição. Enquanto que no método de Suavização Exponencial são atribuídos pesos diferentes a cada observação da série. Desta forma,

a diferença se dá que o método de Suavização Exponencial pode evidenciar informações mais recentes ou mais antigas. O método é definido como

$$\widehat{x}(n+1) = \alpha x(n) + (1-\alpha)\widehat{x}(n), \tag{2.6}$$

onde $\widehat{x}$ é o valor exponencialmente suavizado e α ($0 \leq \alpha \leq 1$) é a constante de suavização. Efetuando a expansão da Equação (2.6), tem-se que

$$x(n+1) = \alpha x(n) + \alpha(1-\alpha)x(n-1) + \alpha(1-\alpha)^2 x(n-2) + \ldots. \tag{2.7}$$

Pode-se verificar que este método é uma média ponderada de valores recentes. Um valor grande para α gera pesos maiores às observações mais recentes. Valores pequenos para α implicam na atribuição de pesos maiores às observações passadas. Desta forma, quanto menor o valor da constante, mais estáveis serão as predições e, consequentemente, qualquer flutuação aleatória no presente contribui com menor importância para a obtenção da predição.

2.5 Modelos Matemáticos para Predição Linear

O método mais antigo e mais estudado de predição de séries temporais remonta ao trabalho de Yule (1927), formalizando o conceito de modelo autoregressivo. O intento de Yule era prever o número anual de manchas solares. A técnica de predição usada por ele consistia em determinar o valor a ser predito através de uma soma ponderada das observações prévias da série. Uma operação puramente linear.

Nos dias atuais, a predição linear de séries temporais está fundamentada nos modelos de Box-Jenkins (BOX; JENKINS; REINSEL, 1994)[1], que integrou o conhecimento existente. Dentre os modelos de Box-Jenkins mais conhecidos destacam-se os modelos autoregressivos (AR), médias móveis (MA) e combinações destes, tais como os modelos ARMA e ARIMA. Todos eles são paramétricos, ou seja, possuem um número finito de parâmetros cujos valores são estimados a partir do sinal ou série temporal.

2.5.1 Modelos Autoregressivos

Um processo autoregressivo de ordem p, AR(p), modela o valor atual de uma variável aleatória $x(n)$ como uma soma ponderada de seus p valores passados, $x(n-1), \ldots, x(n-p)$,

[1] A publicação *Time Series Analysis: Forecasting and Control* por Box, Jenkins e Reinsel (1970) é o texto precursor do livro Box, Jenkins e Reinsel (1994). O livro mais atual inclui um novo material de análise de intervenção, detecção de *outlier* e controle de processos.

adicionado de ruído branco gaussiano $a(n)$. Matematicamente, tem-se

$$\begin{aligned} x(n) &= \phi_0 + \phi_1 x(n-1) + \phi_2 x(n-2) + \ldots + \phi_p x(n-p) + a(n), \\ &= \phi_0 + \sum_{i=1}^{p} \phi_i x(n-i) + a(n), \end{aligned} \tag{2.8}$$

em que ϕ_i, $i = 0, \ldots, p$, são os coeficientes do processo, que juntamente com a ordem da regressão p, constituem os parâmetros do processo. Na Equação (2.8) a sequência $\{a(n), n \geq 0\}$ é uma sequência de ruído branco aditivo gaussiano de média nula e variância constante ($\sigma_a^2 > 0$).

Na forma preditiva, o modelo AR pode ser escrito da seguinte maneira

$$\begin{aligned} x(n+1) &= \phi_0 + \phi_1 x(n) + \phi_2 x(n-1) + \ldots + \phi_p x(n-p+1), \\ &= \phi_0 + \sum_{i=1}^{p} \phi_i x(n-i+1), \end{aligned} \tag{2.9}$$

em que valem todas as definições da Equação (2.8). Independentemente da formulação escolhida, existem várias técnicas para calcular os coeficientes ϕ_i de um modelo AR, sendo a mais comum a dos *Mínimos Quadrados* (MQ) [(AGUIRRE, 2000)], que é equivalente ao método de estimação por máxima verossimilhança (*maximum likelihood*) quando o ruído é gaussiano.

De acordo com a técnica MQ, usando o modelo da Equação (2.9) em uma série temporal com N observações, ou seja, $\{x(n)\}_{n=1}^{N}$, os coeficientes são calculados por meio da seguinte expressão

$$\widehat{\phi} = (\mathbf{Y}^T \mathbf{Y})^{-1} \mathbf{Y}^T \mathbf{p}, \tag{2.10}$$

em que $\widehat{\phi} = [\widehat{\phi}_0 \;\; \widehat{\phi}_1 \;\; \widehat{\phi}_2 \;\; \cdots \;\; \widehat{\phi}_p]^T$ é o vetor de coeficientes, $\mathbf{p}$ é o vetor de predição e $\mathbf{Y}$ é a matriz de regressão. Estes dois vetores e a matriz $\mathbf{Y}$ são dados por

$$\mathbf{p} = \begin{pmatrix} x(p+1) \\ x(p+2) \\ \vdots \\ x(N) \end{pmatrix}, \quad \mathbf{Y} = \begin{pmatrix} 1 & x(p) & \cdots & x(1) \\ 1 & x(p+1) & \cdots & x(2) \\ \vdots & \vdots & \vdots & \vdots \\ 1 & x(N-1) & \cdots & x(N-p-1) \end{pmatrix}. \tag{2.11}$$

Uma vez estimados os coeficientes, estes são utilizados na Equação (2.9) para estimar valores futuros da série temporal. Apesar de sua simplicidade, este método pode apresentar problemas de instabilidade numérica devido à inversão de matrizes, principalmente para valores elevados de p e N pequeno. De qualquer modo, o uso do modelo AR com coeficientes calculados pelo método MQ está amplamente disseminado, não só na Estatística e ciências naturais, como também em Engenharia, Economia e Ciência da Computação, servindo sempre como referência para estudos comparativos.

2.5.2 Modelos de Médias Móveis

Modelos de médias móveis de ordem q, denotados MA(q), são descritos como uma combinação linear finita de q valores passados da sequência de ruído branco

$$x(n) = a(n) + \theta_1 a(n-1) + \theta_2 a(n-2) + \cdots + \theta_q a(n-q), \tag{2.12}$$

em que θ_i são os coeficientes do modelo, que juntamente com sua ordem q constituem os parâmetros do modelo. Estes modelos são mais difíceis de aplicar que modelos AR(p) e o cálculo de seus coeficientes, a partir dos dados observados, é geralmente feito através do método de máxima verossimilhança. Em geral, modelos MA(q) são usados em conjunção com modelos AR(p), a fim de reduzir o número de parâmetros deste.

2.5.3 Modelos Autoregressivos e de Médias Móveis

Para muitas séries encontradas na prática, quando se deseja modelos com um número menor de parâmetros do que os obtidos para um modelo AR(p) ajustado à mesma série, o uso combinado de termos autoregressivos e de médias moveis é a solução adequada (MORETTIN; TOLOI, 2004). Nestes casos, os modelos ARMA(p,q) são a forma mais simples de combinação

$$x(n) = \phi_1 x(n-1) + \phi_2 x(n-2) + \ldots + \phi_p x(n-p) + a(n) + \theta_1 a(n-1) + \theta_2 a(n-2) + \ldots + \theta_q a(n-q), \tag{2.13}$$

em que θ_i e ϕ_i são, respectivamente, os coeficientes autoregressivos e de médias móveis do modelo, que juntamente com as ordens p e q, constituem os parâmetros do mesmo.

2.5.4 Modelos Autoregressivos Integrados de Médias Móveis

Os modelos lineares de Box-Jenkins discutidos até aqui são apropriados somente para descrever séries estacionárias, isto é, estacionária no sentido amplo, em que a série tem média, variância ou autocorrelação que não variam no tempo. Visto que as séries encontradas na prática não são geralmente estacionárias, tais como séries econômicas e financeiras, faz-se necessário discutir um modelo que seja capaz de tratar processos não-estacionários.

Em geral, a estacionariedade de uma série temporal pode ser conseguida através de transformações atuando sobre a série original. Uma forma de tornar séries não-estacionárias em séries estacionárias é através de diferenças entre seus valores consecutivos. Por exemplo,

dada uma série $\{x(n)\}_{n=1}^{N}$ não-estacionária, seja uma nova série $\{w(n)\}_{n=1}^{N-1}$ obtida por meio da seguinte operação

$$w(n) = \Delta x(n) = x(n) - x(n-1). \tag{2.14}$$

Caso esta série ainda não seja estacionária, o mesmo procedimento pode ser novamente aplicado sobre as amostras $w(n)$ até que uma série estocástica seja estacionária o suficiente para permitir que um modelo linear de Box-Jenkins possa ser ajustado a ela.

Uma série temporal $\{x(n)\}_{n=1}^{N}$ tal que tomando-se um número finito de diferenças entre amostras sucessivas torna-se estacionária é chamada *não-estacionária homogênea*. Como o processo é reversível, a série não-estacionária original $\{x(n)\}_{n=1}^{N}$ pode ser obtida a partir da série estacionária omitida pela soma (ou integração) de amostras sucessivas, daí este modelo ser chamado de **Autoregressivo Integrado de Médias Móveis** de ordens p, d e q, ou simplesmente ARIMA(p,d,q).

Do exposto conclui-se que modelos ARIMA são modelos ARMA em que se lança mão de um número d de vezes do expediente de diferenças sucessivas mostrado na Equação (2.14) para produzir uma série estacionária, a partir de uma série não estacionária homogênea. Raramente se verifica valores maiores que $d = 1$ ou $d = 2$. Por isto, o modelo ARIMA é adequado para descrever séries de natureza não-estacionária e que não seja em rajadas (*non-bursty*).

Por fim, pode ser considerado que modelos AR, ARMA e ARIMA são processos com função de autocorrelação que decaem geometricamente com o *lag* (k)[2], ou seja, $\rho(k) \sim r^n$ para algum $0 < r < 1$, à medida que $n \to \infty$. Desta forma, tais modelos são processos indicados para capturar dependência temporal (memória) de curta duração e, portanto, incapazes de capturar os fenômenos observados em séries caóticas.

2.6 Modelos Matemáticos para Predição Não-Linear

A predição foi durante muito tempo dominada por estatísticas lineares. As abordagens tradicionais para predição de séries temporais, tal como o Box-Jenkins ou modelo ARIMA, assumem que a série temporal sob estudo é gerada a partir de processos lineares. Modelos lineares têm vantagens pois podem ser entendidos e analisados mais facilmente e ao mesmo tempo são mais simples computacionalmente. Em alguns casos, aproximações lineares são suficientes para aplicações práticas. No entanto, podem ser totalmente inadequadas se o mecanismo gerador

[2] O parâmetro $k \geq 0$ define a separação temporal (*lag*) entre amostras.

da série temporal for não-linear. Assim, numa série de aplicações, modelos lineares não são satisfatórios, e representações não-lineares devem ser usadas (ZHANG; PATUWO; HU, 1998).

Segundo Aguirre (2000), os sistemas dinâmicos encontrados na prática são, em última análise, não-lineares. A escolha de modelos não-lineares, entretanto, traz consigo um inevitável aumento de complexidade dos algoritmos a serem utilizados. Apesar disto, a razão mais forte para, em uma dada aplicação, optar por modelos não-lineares, deve-se ao fato de modelos não-lineares produzirem certos regimes dinâmicos que modelos lineares não conseguem representar.

Um exemplo de um sistema não-linear bastante simples, com apenas um parâmetro, conhecido como mapa logístico, produz uma série temporal cuja função de autocorrelação se assemelha a de uma sequência de ruído branco (KUGIUMTZIS; LILLEKJENDLIE; CHRISTOPHERSEN, 1994), quando na verdade corresponde a uma série temporal caótica. O mapa logístico ou mapa quadrático é descrito pela seguinte equação

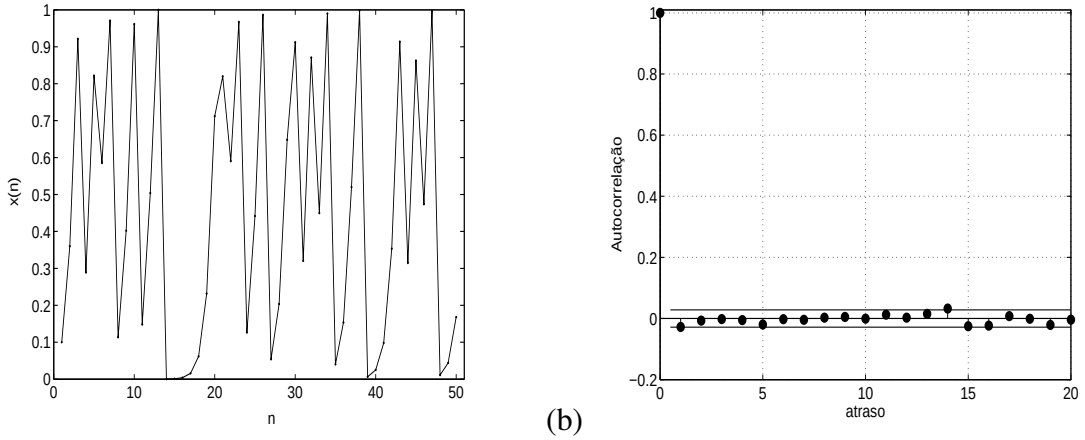
$$x(n+1) = \xi x(n)[1-x(n)], \tag{2.15}$$

em que o parâmetro $\xi > 0$ é uma constante a ser escolhida em função do comportamento desejado. Para $1 \leq \xi \leq 4$, a trajetória da variável de estado x produz valores restritos ao intervalo $[0,1]$ para condições iniciais no mesmo intervalo (KUGIUMTZIS; LILLEKJENDLIE; CHRISTOPHERSEN, 1994).

Para valores de ξ entre 0 e 3, na Equação (2.15), o estado assintótico de $\{x(n)\}$ consiste em apenas um ponto de equilíbrio. Para $3 < \xi < 3{,}57$, a solução assintótica consiste de ciclos-limites de diferentes periodicidades. Para valores de $3{,}57 < \xi \leq 4$, o sistema passa a apresentar comportamento caótico (KAPLAN; GLASS, 1995). Na Figura 3 é mostrada uma realização do mapa logístico para $\xi = 4$, em que pontos sucessivos são ligados por linhas retas para facilitar a visualização.

Devido a esta curiosa, porém falsa, semelhança com processos estocásticos lineares, muitas séries temporais caóticas costumam ser tratadas a partir de modelos lineares convencionais, tal como o modelo autoregressivo com médias móveis (*autoregressive moving average*, ARMA). Contudo, tais modelos têm se mostrado inadequados para a análise e predição de sistemas caóticos, pois não capturam a dinâmica não-linear subjacente à série temporal de interesse. Posto de maneira mais formal, isto se deve ao fato de que modelos lineares conduzem somente a soluções exponencialmente decrescentes ou periodicamente oscilantes, chamadas genericamente de pontos ou soluções de equilíbrio. Sistemas caóticos apresentam outras possíveis soluções

ou comportamentos que só são obtidos quando se usa as ferramentas e modelos não-lineares adequados.



(a) (b)

Figura 3 – (a) Mapa logístico para estado caótico; (b) autocorrelação para o mapa logístico com $\xi = 4$.

A predição de séries temporais pode ser vista como um mapeamento $f : \mathbb{R}^p \to \mathbb{R}$, a partir do vetor de regressão $\mathbf{x}(n) \in \mathbb{R}^p$, $[x(n)\ x(n-1)\ \cdots\ x(n-p+1)]$, para a saída $\widehat{x}(n+1)$. Na predição linear, este mapa corresponde a um hiperplano simples, determinado pelos coeficientes autoregressivos. Na predição não-linear é preciso um domínio muito mais rico de dinâmicas. A configuração básica de predição $x(n+1) = \widehat{x}(n+1) + e(n)$ é mantida a mesma, no entanto, o modelo baseado na autoregressão não-linear segue

$$\begin{aligned} x(n+1) &= f[x(n)\ x(n-1)\ \cdots\ x(n-p+1)] + e(n), \\ &= f(\mathbf{x}(n)) + e(n), \end{aligned} \tag{2.16}$$

onde $f(\cdot)$ é uma função não-linear.

Diferentes métodos de predição não-lineares são frequentemente caracterizados pelo modo como a função $f(\cdot)$ é construída. Pode-se destacar algumas destas representações não-lineares, tais como o modelo bilinear desenvolvido inicialmente por Granger e Andersen (1978); modelos TAR (*Threshold Autoregressive*) realizados a partir de modelos autoregressivos com *threshold* e proposto por Tong e Lim (1980); e a familia de modelos ARCH (*Auto-Regressive Conditional Heterocedasticity*) formulado por Engle (1982), que busca a estimação da variância dos retornos passados.

No entanto, estes modelos não-lineares ainda são limitados, uma vez em que a relação explícita destes modelos pode ser uma hipótese com pouca informação da lei subjacente. Na verdade, a formulação de um modelo não-linear para um determinado conjunto de dados é uma

tarefa muito difícil, pois existem muitos possíveis padrões não-lineares e um modelo não-linear especificado pode não ser suficiente para capturar todas as características do modelo (ZHANG; PATUWO; HU, 1998).

Por volta de 1980, com o aumento do poder computacional dos computadores, dois desenvolvimentos cruciais ocorreram, permitindo que grandes séries temporais pudessem ser analisadas e algoritmos não-lineares mais complexos puderam ser construídos. O primeiro desenvolvimento é a reconstrução do espaço de estados, pelo teorema de imersão de Takens (TAKENS, 1981), baseado no estudo da topologia diferencial e de sistemas dinâmicos.

O segundo é o surgimento da área da aprendizagem de máquina, caracterizada principalmente pelos algoritmos de RNAs. RNAs, que são abordagens não-lineares em sua maioria, são capazes de realizar modelagem não-linear, sem conhecimento *a priori* sobre as relações entre variáveis de entrada e de saída. Com isto, RNAs tornaram-se ferramentas de modelagem mais gerais e flexíveis para a predição não-linear de séries temporais. Aplicações com sucesso de RNAs em um grande número de tarefas de predição e modelagem de séries temporais podem ser encontradas na pesquisa de técnicas e aplicações de inteligência computacional de Palit e Popovic (2005).

Estes dois assuntos são as principais bases teóricas desta tese. A reconstrução do espaço de estados é o próximo tema a ser tratado neste capítulo, ficando as RNAs como tema de discussões nos capítulos seguintes.

2.7 Reconstrução do Espaço de Estados

Nesta seção é apresentada uma breve introdução da teoria de imersão e da reconstrução de espaço de estados. O leitor interessado pode encontrar mais detalhes em (KAPLAN; GLASS, 1995; KANTZ; SCHREIBER, 1997; ABARBANEL; FRISON; TSIMRING, 1998; SCHREIBER, 1999).

A ideia básica da reconstrução do espaço de estado está calcada no fato de que a série temporal de uma certa variável de estado x_i contém informações sobre as outras variáveis de estado não-observáveis, podendo ser usadas para prever o vetor de estado atual $\mathbf{x}(n)$. Ao processo de predição do vetor de estados, a partir de uma única série temporal, dá-se o nome de reconstrução do espaço de estados.

A reconstrução do espaço de estado está baseada no Teorema da Imersão de Takens (*Takens' embedding theorem*) (TAKENS, 1981). Este teorema permite reconstruir um espaço

de estado d_E-dimensional similar ao espaço de estado original, a partir de uma única variável de estado, que é a variável medida. Este espaço reconstruído deve preservar as propriedades invariantes do sistema dinâmico subjacente (SAVI, 2004).

De modo geral, o teorema de Takens é posto da seguinte maneira. Seja uma série temporal de tamanho N (suficientemente grande) e livre de ruído, $\{x(1), x(2), \ldots, x(N)\}$, obtida a partir de uma das variáveis de um sistema dinâmico determinístico. O espaço de estados deste sistema pode ser exatamente reconstruído por um grupo de vetores, chamados coordenadas de atraso, montados a partir de amostras atrasadas daquela série temporal da seguinte forma

$$\mathbf{x}(n) = [x(n) \quad x(n-\tau) \quad x(n-2\tau) \quad \cdots \quad x(n-(d_E-1)\tau)]^T, \tag{2.17}$$

em que $x(n)$ é a amostra da série temporal no tempo n, d_E é chamada de dimensão de imersão (*embedding dimension*) e τ é chamado de atraso de imersão (*embedding delay*). Note que esta equação é uma versão geral do vetor $\mathbf{x}(n)$ de regressão usado em modelos estocásticos (Equação(2.1)), caso em que $\tau = 1$.

Uma ideia semelhante ao teorema de Takens foi proposta originalmente no trabalho de Whitney (1936). Desta forma costuma-se referir também como "Teorema de Whitney", porque ele é o primeiro a provar que uma variedade suave (*smooth manifold*) de dimensão n pode ser imersa em $\mathbb{R}^{2n+1}$.

O teorema de Takens é um importante teorema porque implica na seguinte constatação: se as suposições gerais do teorema são satisfeitas, existe uma função $f(\cdot)$, tal que, $x(n+1) = f(\mathbf{x}(n))$. Em outras palavras, se as coordenadas de atraso $\mathbf{x}(n)$, montadas como na Equação (2.17), reconstroem com exatidão o espaço de estados, então existe uma função $f(\cdot)$ que gera a variável de estado $x(n+1)$ com exatidão. Contudo, como esta função é geralmente desconhecida, o problema de reconstrução do espaço de estados pode intuitivamente ser colocado como um problema de predição de séries temporais, no qual o objetivo é determinar os valores futuros da variável observada, ou seja,

$$\widehat{x}(n+1) = \widehat{f}(\mathbf{x}(n)), \tag{2.18}$$

em que $\widehat{x}(n+1)$ é uma estimativa do valor exato de $x(n+1)$ e $\widehat{f}(\cdot)$ denota uma aproximação da função $f(\cdot)$. Assim, conclui-se que um bom modelo computacional para a aproximação $\widehat{f}(\cdot)$, resulta em uma reconstrução fidedigna do espaço de estados, pois os valores preditos para $\widehat{x}(n+1)$ são próximos dos valores exatos.

Como visto, o teorema Takens demonstra que, na ausência de ruído, um espaço de estados multidimensional pode ser reconstruído a partir de uma série temporal escalar. Este teorema, entretanto, dá pouca orientação sobre considerações de ordem prática para uma boa reconstrução do espaço de estados. Abarbanel *et al.* (1993) discute extensivamente o problema e apresenta metodologias para estimação da dimensão e atraso de imersão de sistemas com ou sem ruído. Casdaglia *et al.* (1991) e Sauer, Yorke e Casdagli (1991) também discutem o problema de reconstrução do espaço de estados, apresentando técnicas de predição e modelagem de séries temporais não-lineares na presença de ruído.

2.7.1 Estimação da Dimensão de Imersão

A dimensão de imersão d_E do espaço de estados reconstruído é um importante parâmetro a ser determinado. Geralmente ela é diferente da dimensão exata (e desconhecida) do espaço de estados, $m = [d] + 1$, em que $[d]$ denota a parte inteira da dimensão fractal do atrator d. Takens (1981) mostrou ser suficiente que $d_E \geq 2[d] + 1$. O teorema garante que o atrator imerso no espaço de estado d_E-dimensional é desdobrado ($unfolded$) sem qualquer auto-interseções. A condição $d_E \geq 2[d] + 1$ é suficiente mas não é necessária, e um atrator pode ser reconstruído também na prática, com uma dimensão de imersão tão baixa quanto $[d] + 1$ (KUGIUMTZIS; LILLEKJENDLIE; CHRISTOPHERSEN, 1994). Nos próximos parágrafos são descritos métodos para estimar a dimensão de imersão d_E, a partir de uma série temporal com ou sem ruído.

Cálculo de invariantes geométricos. Este método baseia-se na tentativa de encontrar um valor assintótico de alguma invariante geométrica (e.g. dimensão de correlação) do sistema dinâmico em função do valor da dimensão de imersão. Assim, quando o invariante geométrico calculado estabilizar em um determinado valor, o valor escolhido para a dimensão de imersão é o menor valor para o qual aquele invariante estabiliza (GRASSBERGER; PROCACCIA, 1983b; GRASSBERGER; PROCACCIA, 1983a).

Decomposição em valores singulares. Broomhead e King (1986) propuseram um método baseado na diagonalização da matriz de covariância dos vetores de reconstrução, identificando os seus autovalores. O número de autovalores não-nulos é um valor estimado da dimensão mínima de imersão.

Método dos falsos vizinhos (*False Neighbors*). Este método é proposto por Kennel, Brown e Abarbanel (1992) e baseia-se no fato de que em um atrator bem reconstruído não

deve haver cruzamento de uma trajetória consigo mesma; ou seja, pontos não devem se repetir, uma vez que a dinâmica é caótica. Assim, avalia-se um vizinho como "verdadeiro" ou "falso" apenas em virtude da projeção do sistema em uma determinada dimensão. Desta forma, um falso vizinho é um ponto do sinal que só corresponde a um vizinho devido a observação das órbitas em um espaço muito pequeno, $D < d_E$. Quando o espaço está imerso em uma dimensão $D > d_E$, todos os pontos vizinhos de todas as órbitas são vizinhos verdadeiros.

Método de Cao (1997). Este método é uma extensão da técnica anterior, sendo voltada para aplicações em séries temporais estocásticas ou determinísticas. Este método também é pouco sensível ao tamanho da série em questão. O procedimento consiste em explorar a estrutura geométrica do atrator à medida que se aumenta o valor de d_E, a partir de 1. Se d_E é muito pequeno, o atrator apresenta auto-intersecções da trajetória do atrator no espaço de estados. Nestes casos, pontos próximos no atrator são, ou vizinhos exatos devido à dinâmica do sistema, ou falsos vizinhos devido às auto-intersecções. Em dimensões maiores, em que as auto-intersecções são desfeitas, os falsos vizinhos são revelados visto que eles vão se distanciando. O objetivo do método de Cao é encontrar um limiar mínimo para d_E, tal que não existam falsos vizinhos no atrator reconstruído a partir desta dimensão de imersão.

Nesta tese, adota-se o método de Cao, pois, o mesmo leva a resultados melhores no processo de predição não-linear associado. Devido a sua importância, o método de Cao está descrito em maiores detalhes no Apêndice A.

2.7.2 Estimação do Atraso de Imersão

Embora Takens (1981) não tenha considerado este parâmetro relevante na sua formulação original, em séries temporais reais, que não estão livres de ruído (muito pelo contrário!), tal parâmetro é da maior importância. Para τ demasiado pequeno, coordenadas de atraso $\mathbf{x}(n)$ consecutivas tornam-se similares, de tal forma que o atrator reconstruído é esticado ao longo de uma diagonal e obscurecido facilmente pelo ruído. Assim, é desejável uma escolha de τ que mantenha coordenadas de atrasos consecutivas mais independentes entre si. Por outro lado, valores demasiado grandes causam perda de informação contida nos dados, tal que dois vetores, temporalmente próximos, tornam-se bastante afastados, dando origem a incertezas na reconstrução (KUGIUMTZIS; LILLEKJENDLIE; CHRISTOPHERSEN, 1994).

Uma das principais ferramentas para a estimação de independência entre termos é a

função de autocorrelação (FAC), cuja expressão, para um sinal de média zero, é dada por

$$R_X(k) = \frac{\sum_{n=1}^{N-k} x(n)x(n+k)}{N-k}, \tag{2.19}$$

em que o parâmetro $k \geq 0$ é separação temporal (*lag*) entre as amostras. A FAC é uma medida quantitativa da dependência temporal entre amostras sucessivas de uma série temporal, propriedade esta associada com a presença de "memória" no sistema. Uma série temporal, em que $R_X(k) \neq 0$ para $k = 0$, e $R_X(k) \approx 0$ para $k > 0$, é típica de sistemas sem memória, de modo que tal sequência é chamada genericamente de ruído branco.

Uma formulação alternativa da FAC, chamada de função coeficiente de autocorrelação (FCAC), divide a Equação (2.19) pela variância amostral $\sigma_X^2 = R_X(0)$ da série, resultando na seguinte expressão

$$\rho_X(k) = \frac{R_X(k)}{R_X(0)} \approx \frac{\sum_{n=1}^{N-k} x(n)x(n+k)}{\sum_{n=1}^{N} x^2(n)}, \tag{2.20}$$

tal que, neste caso, o maior valor de $\rho_X(k)$ é 1, obtido para $k = 0$.

Uma escolha comum para τ é o atraso (*lag*) para o qual a FAC atinge seu primeiro valor nulo. Por este método, as coordenadas de atraso passam a ser linearmente não-correlacionadas. Outra regra semelhante consiste em escolher o atraso de imersão com o *lag* no qual a FAC decai para $1/e = 0{,}37$ (KANTZ; SCHREIBER, 1997). Williams (1997) sugere outro método para a escolha do atraso de imersão mínimo, como sendo o *lag* seguinte ao ponto em que a FAC pára de diminuir; ou seja, no primeiro mínimo da FAC.

Uma objeção aos procedimentos mencionados anteriormente é que a estimação do atraso de imersão através da FAC é baseada em estatísticas lineares, não levando em conta correlações não-lineares (KANTZ; SCHREIBER, 1997). Fraser e Swinney (1986) sugerem uma escolha para τ mais adequada ao problema de modelagem de sistemas dinâmicos, baseado em um critério de medida de independência mais geral, tal como a informação ganha em bits sobre $x(n+\tau)$ dada a medida de $x(n)$. Em suma, esta medida é conhecida como informação mútua e o primeiro mínimo no gráfico desta grandeza, em função de τ, é frequentemente sugerida como uma boa estimativa para τ (KUGIUMTZIS; LILLEKJENDLIE; CHRISTOPHERSEN, 1994). Nesta tese, este é o critério adotado para determinar o atraso de imersão.

A expressão para o cálculo da informação mútua é baseada na entropia de Shannon (1948). Dentro de um intervalo de dados de uma série temporal, é criado um histograma dos dados. Denota-se por p_i a probabilidade que o sinal assuma um valor dentro da ith caixa (*bin*) do

histograma e assume-se que p_{ij} é a probabilidade de que $x(n)$ esteja na caixa i e $x(n+\tau)$ esteja na caixa j. Então, a informação mútua para um atraso no tempo τ é definida como

$$I(\tau) = \sum_{i,j} p_{ij}(\tau)\ ln\ p(ij)(\tau) - 2\sum_{i} p_i\ ln\ p_i, \tag{2.21}$$

em que ln denota o logaritmo natural.

2.8 Tipos de Predição

A capacidade de predizer o comportamento futuro de uma série particular de eventos, com conhecimento apenas do seu presente e do seu passado, é uma das formas de verificar se um modelo matemático definitivamente "entendeu" os dados observados. Neste contexto, "entender" significa remover as possíveis redundâncias nos dados e, consequentemente, descobrir regularidades estatísticas ou dinâmicas na série apresentada. Desta forma, comparar a simulação da série obtida com dados observados é provavelmente a forma mais usual de validar um modelo, isto é, saber se um dado modelo é válido ou não para fazer predições.

Para validar um modelo identificado, ou seja, dizer que ele está apto para ser utilizado, é importante testar a sua resposta (saída) para dados de entrada diferentes daqueles vistos durante a identificação. Estes novos dados podem ser obtidos por meio de novas medições, o que nem sempre é viável. Para contornar este obstáculo, o procedimento mais comum consiste em identificar o modelo apenas com uma parte dos dados, guardando a parte restante para ser usada para testar o desempenho do modelo.

2.8.1 Preditor de Um Passo

A predição de séries temporais é um problema de processamento de sinais em que se tem uma sequência de N amostras de uma determinada variável escalar, $\{x(n),\ x(n-1),\ \ldots,\ x(n-N+1)\}$, uniformemente espaçadas no tempo, e cujo objetivo é obter uma estimativa $\hat{x}(n+1)$, para o próximo elemento da série. Este procedimento é conhecido como predição um-passo-adiante (UPA), em que se estima somente o próximo valor da série temporal, sem realimentação do valor predito para a entrada do regressor, conforme ilustrado na Figura 4. Em outras palavras, o regressor de entrada contém somente observações exatas da série temporal.

Se for lembrado que os algoritmos de estimação de parâmetros normalmente minimizam a soma do quadrado dos resíduos, torna-se evidente que, graças a tais algoritmos, para um determinado conjunto de regressores, os erros de predição de um-passo-adiante serão sempre

Figura 4 – Preditor sem realimentação.

os menores possíveis. Consequentemente, predições UPA não são um bom teste para validar modelos, uma vez que modelos ruins normalmente apresentam predições UPA boas.

Aguirre (2000) sugere outros métodos para validação de modelos com dinâmica caótica como, por exemplo, a análise do maior expoente de Lyapunov, pois, a simples predição UPA não é suficiente. Abarbanel *et al.* (1993) sugere o uso da predição recursiva como validação para sistemas não-lineares caóticos, pois, embora erros de predições UPA sejam frequentemente elevados, estes modelos podem reproduzir melhor o comportamento de um sistema. Desta forma, nem sempre uma boa predição UPA leva a uma boa reprodução da dinâmica do sistema.

2.8.2 Preditor de Múltiplos Passos

Quando se está interessado num horizonte de predição maior, um outro método para construção de preditores é comumente utilizado. Método este conhecido como predição *h*-passos-adiante (HPA) ou "predição com realimentação dos valores preditos", Figura 5. A saída do modelo deve ser realimentada para o regressor de entrada. Neste caso, os componentes do regressor de entrada, previamente compostos apenas de valores exatos da série temporal, são gradativamente trocados por valores preditos. Um esquema ilustrativo para predição HPA, que também pode ser chamada de predição recursiva, é mostrado na Tabela 1.

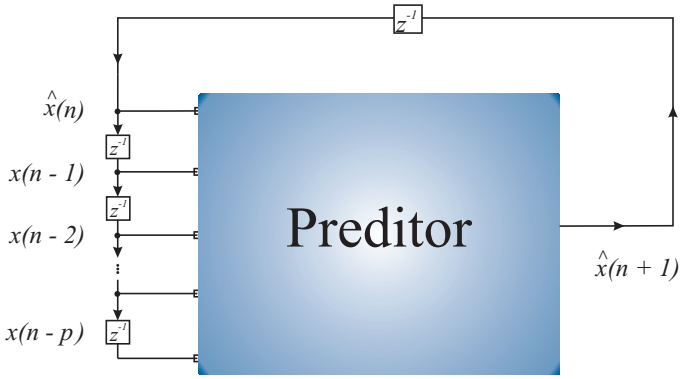


Figura 5 – Preditor recursivo, com realimentação.

Se o horizonte de predição tende ao infinito, em algum momento no tempo, a entrada do regressor começa a ser composta somente de valores previamente estimados da série temporal. Neste caso, a tarefa de predição HPA torna-se uma tarefa de modelagem dinâmica, em que o modelo atua como um sistema autônomo (HAYKIN; PRINCIPE, 1998). A predição HPA, ao contrário da predição UPA, é uma boa maneira de testar se o modelo consegue explicar as observações feitas.

Tabela 1 – Predição h-passos-adiante

Instante	Regressor	Saída
n	$x(n), x(n-1), x(n-2), \ldots, x(n-(d_E-1))$	$\widehat{x}(n+1)$
$n+1$	$\widehat{x}(n+1), x(n), x(n-1), \ldots, x(n-(d_E-2))$	$\widehat{x}(n+2)$
$n+2$	$\widehat{x}(n+2), \widehat{x}(n+1), x(n), \ldots, x(n-(d_E-3))$	$\widehat{x}(n+3)$
$\vdots$	$\vdots$	$\vdots$

Predição HPA e a modelagem dinâmica são mais complexas de se trabalhar do que a predição UPA e acredita-se que estas são tarefas em que as redes neurais desempenham uma importante função, em particular as arquiteturas neurais recorrentes (PRINCIPE; EULIANO; LEFEBVRE, 2000).

2.8.3 Preditor Direto

A predição direta é um método alternativo para a predição múltiplos-passos-adiante. Este método tem como objetivo aproximar uma função de mapeamento $f(\cdot)$ para cada valor a ser predito. Isto é, construir um modelo $f_h(\cdot)$ para os próximos H valores futuros da série temporal

$$x(n+h) = f_h(\mathbf{x}(n)), \qquad h = 1, \ldots, H, \tag{2.22}$$

onde $f_h(\cdot)$ é uma função linear ou não-linear.

Uma característica interessante deste método é o fato de não estar propenso ao acúmulo dos erros de predição, como acontece no preditor recursivo de múltiplos-passos-adiante. Na predição direta a saída do modelo não realimenta o regressor de entrada, assim os componentes do regressor de entrada são compostos apenas de valores exatos da série temporal. A deficiência deste método é que H diferentes modelos devem ser construídos, aumentando assim a complexidade computacional do problema de predição.

Os resultados de Zhang (1994) mostram que a predição direta é melhor que a predição recursiva múltiplos-passos-adiante. Tikka e Hollmén (2008) discutem várias estratégias

de predição, dando preferência para a predição direta. Já Sauer (1994) conclui que a predição recursiva de múltiplos-passos-adiante é melhor que a predição direta.

De toda forma, Taieb, Sorjamaa e Bontempi (2010) observam que os H modelos do método direto são construídos de forma independente, gerando uma independência condicional dos H preditores de $\widehat{x}(n+h)$. Isso evita que o método considere na construção do modelo as complexas dependências entre as variáveis da série temporal, consequentemente, gerando um possível erro de precisão da predição.

2.8.4 Preditor MIMO

Como uma possível forma de corrigir as deficiências de ambos os métodos mostrados anteriormente para a predição múltiplos-passos-adiante, Bontempi (2008) propõe um método de predição com múltiplas entradas e múltiplas saídas. As diversas saídas são destinadas uma para cada horizonte de predição requerido, como indicado na Figura 6. Desta forma, este método transforma o modelo de predição num modelo MIMO (*Multiple-Input and Multiple-Output*).

Figura 6 – Preditor MIMO.

O método de predição MIMO consiste em construir um mapeamento $f : \mathbb{R}^p \rightarrow \mathbb{R}^H$ pela seguinte equação

$$[x(n+H) \ \cdots \ x(n+h) \ \cdots \ x(n+1)] = f[x(n) \ x(n-1) \ \cdots \ x(n-p+1)], \tag{2.23}$$

onde $f(\cdot)$ é uma função linear ou não-linear, H é o horizonte de predição múltiplos-passos-adiante e p é a ordem do vetor de regressão.

Vale notar que neste caso a predição retorna não somente um escalar, mas sim um vetor de predição. Diferente do método direto, que necessita de um modelo para cada horizonte de predição, o método de predição MIMO permite fazer predições de múltiplos-passos-adiante utilizando apenas um modelo de predição.

2.9 Conclusão

Este capítulo teve como principal finalidade mostrar os conceitos básicos de predição linear e não-linear. Foi feito uma revisão do modelo linear ARMA, seguido pela análise da necessidade de se usar modelos não-lineares para predição de séries temporais. Também são feitas determinadas considerações sobre a reconstrução do espaço de estados.

Alguns conceitos abordados neste capítulo são usados nos capítulos seguintes. No próximo capítulo são apresentadas a redes neurais dinâmicas, base para as redes propostas neste trabalho.

3 REDES NEURAIS SUPERVISIONADAS PARA PREDIÇÃO DE SÉRIES TEMPORAIS

3.1 Introdução

Este capítulo tem por objetivo apresentar sucintamente as arquiteturas de redes neurais avaliadas nesta tese, a fim de facilitar a compreensão dos métodos de predição de séries temporais que serão propostos nos capítulos seguintes.

De forma geral, redes neurais artificiais podem ser divididas quanto ao tipo de aprendizado em duas categorias: (*i*) redes com aprendizado supervisionado e (*ii*) redes com aprendizado não-supervisionado. No caso supervisionado, cada entrada apresentada à rede vem acompanhada de uma saída desejada, a fim de permitir uma modificação dos parâmetros ajustáveis em função do erro entre a resposta fornecida pela rede e a saída real desejada. Ao final da etapa de ajuste dos parâmetros, chamada genericamente de treinamento, as respostas da rede para todas as entradas devem ser próximas das saídas desejadas. No caso não-supervisionado, a rede neural detecta padrões e características estatísticas do espaço de entrada, de forma a construir uma representação de dimensionalidade reduzida do mesmo no conjunto de pesos sinápticos de seus neurônios.

As arquiteturas de redes neurais descritas neste capítulo são todas de aprendizado supervisionado, diferenciando umas das outras apenas pelo modo que processam informação temporal, ou seja, utilizam ou não laços de realimentação (*feedback loops*). Desta forma, arquiteturas supervisionadas que não contenham tais laços, comumente chamadas de *não-recorrentes* ou *feedforward*, são discutidas em primeiro lugar. Em seguida, arquiteturas contendo laços de realimentação, doravante chamadas de *redes neurais recorrentes* são apresentadas. As descrições das arquiteturas apresentadas neste capítulo são baseadas, principalmente, nos livros de Principe, Euliano e Lefebvre (2000), Hertz, Krogh e Palmer (1991) e Haykin (1999). Referências adicionais são citadas quando necessárias.

3.2 Predição de Séries Temporais via RNAs

Uma das mais importantes áreas de aplicação de RNAs é a predição de sinais. Várias características das RNAs as tornam atraentes para uma tarefa de predição, quatro delas são discutidas a seguir.

Primeiro, ao contrário dos modelos tradicionais, RNAs são baseadas em métodos que

exigem poucas (ou nenhuma) suposições a priori sobre os dados em análise. RNAs aprendem a partir de exemplos, buscando aproximar relações funcionais entre os dados de entrada e saída, mesmo quando as relações matemáticas subjacentes são desconhecidas ou difíceis de descrever. Assim, RNAs são adequadas para problemas cujas soluções requerem um conhecimento que é difícil de especificar, mas para os quais existem muitos dados ou observações. Neste sentido, RNAs podem ser tratadas como um método estatístico, não-linear, multivariado e não paramétrico. Esta abordagem de modelagem com a capacidade de aprender com a experiência é muito útil para muitos problemas práticos, uma vez que, às vezes, é mais fácil possuir os dados do que ter bons palpites teóricos sobre as leis que regem os sistemas subjacentes.

Segundo, RNAs podem generalizar o conhecimento adquirido para um novo conjunto de dados. Depois de aprender a relação entre os dados de entrada e saída apresentados, as RNAs conseguem frequentemente estimar a parte não apresentada de um novo conjunto de observações, mesmo que as amostras contenham informação ruidosa. Com isto, a predição realizada a partir de exemplos passados é uma área de aplicação ideal para redes neurais em séries temporais, pelo menos em princípio.

Terceiro, RNAs são aproximadores universais de funções (CYBENKO, 1989; HORNIK, 1991; HORNIK; STINCHCOMBE; WHITE, 1989). Tem sido demonstrado que uma rede pode aproximar qualquer função contínua para qualquer precisão desejada, isto é, com grau de precisão arbitrário. Qualquer modelo de predição assume que existe uma relação matemática subjacente (conhecida ou não) entre as entradas (valores passados da série temporal e/ou outras variáveis relevantes) e as saídas (valores futuros). Desta forma, RNAs podem ser um interessante método alternativo para identificar tal função.

Finalmente, RNAs são não-lineares e os sistemas do mundo real são geralmente não-lineares. Apesar disto, a formulação de um modelo não-linear para um determinado conjunto de dados é uma tarefa muito difícil, pois existem muitos possíveis padrões não-lineares e um modelo não-linear pré-especificado não pode ser suficiente para capturar todas as características importantes do sistema. Redes neurais artificiais, que são não-lineares em geral, são capazes de realizar tal modelagem, sem conhecimento a priori sobre as relações entre entrada e saída das variáveis.

Zhang, Patuwo e Hu (1998) catalogou vários trabalhos onde é observado o desempenho das redes neurais na predição de séries temporais, em comparação com métodos estatísticos clássicos. Há muitos relatos inconsistentes na literatura sobre o desempenho das RNAs para

tarefas de predição. Para alguns casos, RNAs possuem desempenho pior do que modelos estatísticos lineares. O motivo para isto pode estar ligado simplesmente ao fato de os dados em questão serem lineares, sem muita complexidade. Com isto, não se poderia esperar que RNAs tivessem desempenho melhor do que modelos lineares para relações lineares. Em outros casos, a dificuldade pode ser explicada no sentido que a estrutura de rede utilizada não seja adequada para modelar o conjunto de dados.

A ideia de utilizar RNAs para a predição de séries temporais não é recente. Um apanhado histórico do uso de redes neurais em tarefas de predição é feito em Zhang, Patuwo e Hu (1998), podendo-se afirmar que a primeira aplicação remonta à década de 60 do século passado. Hu (1964), em sua tese, usa a rede linear adaptativa de Widrow-Hoff [1] para predições climáticas, mas devido à falta de um algoritmo geral de treinamento para redes multicamadas na época, os resultados foram bastante limitados. Werbos (1974) foi o primeiro a formular o algoritmo *backpropagation* e descobrir que RNAs treinadas com este algoritmo superam os métodos estatísticos tradicionais, tais como abordagens de regressão e Box-Jenkins. Mas a grande contribuição no uso de RNAs para a predição veio em 1986, com os trabalhos de Rumelhart, Hinton e Williams (1986) e Werbos (1986), que difundiram o *backpropagation* mostrando a sua utilização para a aprendizagem de máquina e por demonstrarem como isto poderia funcionar.

Uma das primeiras aplicações de sucesso utilizando RNAs na predição de séries temporais é relatado no trabalho de Lapedes e Farber (1987) utilizando duas séries temporais caóticas geradas pelo mapa logístico e pela equação diferencial de Mackey-Glass. Eles projetaram redes neurais *feedforward* com a capacidade de reproduzir tais sistemas dinâmicos não-lineares. Apesar das dificuldades computacionais da época, seus resultados mostram que RNAs podem ser usadas para a modelagem e fazer predição de séries temporais não-lineares com boa acurácia.

Em 1990, um novo estímulo foi dado à pesquisa de predição de séries temporais. Neil Gershenfeld (pesquisador do *MIT Media Laboratory*) e Andrea Weigend (pesquisador da *University of Colorado* e da *Xerox PARC*) naquela ocasião cursavam juntos o programa *Complex Systems Summer School* do *Santa Fé Institute*. As pesquisas de Gershenfeld e Weigend envolviam temas que requeriam a análise de séries temporais. Os dois pesquisadores encontraram grandes dificuldades ao procurar na literatura especializada técnicas que fossem utilizadas nas mais diver-

[1] Algoritmo do mínimo quadrado médio (LMS, *least mean squares filter*), conhecido como algoritmo Widrow-Hoff ou regra delta, serviu de base para às redes neurais artificiais Adaline e Madaline e posteriormente à técnica do *backpropagation*. Bernard Widrow foi co-inventor do algoritmo juntamente com seu então doutorando Marcian Edward Ted Hoff em 1960.

sas áreas do conhecimento. Dentre a escassa literatura encontrada, perceberam ainda a ausência de estudos que buscavam explicar como as técnicas conhecidas até então se relacionavam entre si, e qual a confiabilidade de tais técnicas. Motivados por estas dificuldades, Gershenfeld e Weigend decidiram desafiar a comunidade científica propondo uma competição (CASTRO, 2001).

A ideia tinha o claro intuito de tentar entender e organizar questões pertinentes à análise de séries temporais e difundir novas técnicas para além dos domínios restritos da área do conhecimento específico, na qual trabalhavam. Gershenfeld e Weigend pretendiam também que as séries adotadas na disputa se tornassem *benchmarks* para avaliação de futuros resultados de novas pesquisas. Para surpresa geral, a ideia foi bem recebida pela comunidade científica. A competição foi patrocinada pelo *Santa Fe Institute* e contou com um grupo de consultores envolvidos com análise de séries temporais. Os consultores foram escolhidos entre pesquisadores das áreas de biologia, economia, física pura, física experimental, astrofísica, análise numérica, estatística e sistemas dinâmicos.

Na primavera de 1992 foi realizado um encontro para explorar os resultados da competição, objetivando reunir os participantes do desafio, membros dos grupos que coletaram dados, consultores e demais interessados.

O livro *Times Series Prediction: Forecasting the Future and Understanding the Past* (WEIGEND; GERSHEFELD, 1994), traz uma compilação dos trabalhos apresentados na competição. O maior interesse demonstrado pelos grupos participantes da competição foi na predição em séries temporais e a maior parte da discussão esteve centrada em modelos não-lineares. O número dominante de contribuições e também as melhores predições foram devidas aos métodos conexionistas, conhecidos como redes neurais artificiais.

Nas últimas duas décadas houve um grande esforço da pesquisa de RNAs para a predição de séries temporais. A literatura é vasta e crescente, como pode ser vista na Figura 7, em que apresenta o número de publicações em revistas científicas do grupo editorial Elsevier utilizando RNAs para predição de séries temporais. Esta pesquisa foi feita no final do ano de 2011 utilizando o portal de periódicos da CAPES. Pode-se constatar que o ano de 1994 foi um divisor em número de publicações. Antes desta data existiam poucas publicações, já após, percebe-se que houve um crescimento, podendo-se estimar que no ano de 2012 deverá existir mais de mil publicações com os termos utilizados na pesquisa.

Marquez *et al.* (1992) e Hill *et al.* (1994) fazem uma breve revisão da literatura, comparando RNAs com modelos estatísticos e modelos de regressão para predição de séries

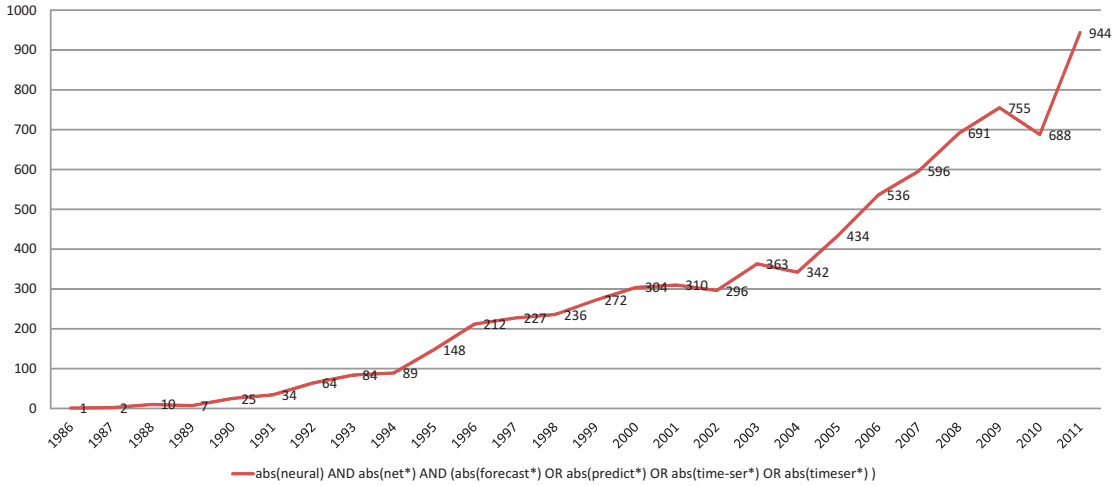


Figura 7 – Número de publicações em revistas cientificas utilizando RNAs para predição de séries temporais.

temporais até o início da década de 1990. Uma abordagem mais global é feita em Zhang, Patuwo e Hu (1998), que apresentou uma investigação de referência nesta área. Apesar de ser uma referência já antiga, este trabalho tem uma vasta bibliografia, que fornece um resumo geral da obra na predição com RNAs até então. Mas o seu principal benefício é que ele serve como um guia inicial do uso de redes neurais para futuras pesquisas. O trabalho de Crone, Hibon e Nikolopoulosc (2011) faz algumas considerações sobre o resultado da competição de predição NN3 (*Forecasting Competition for Neural Networks & Computational Intelligence*). Os autores discutem o desempenho das RNAs frente aos modelos estatísticos, apresentando as principais dificuldades e potencialidades das redes depois do longo caminho que elas trilharam nos últimos 25 anos, desde o estabelecimento do algoritmo *backpropagation* em 1986.

3.3 Predição via RNAs Não-Recorrentes

Redes neurais não-recorrentes são as mais populares e de maior uso em aplicações práticas, devido ao seu comprovado desempenho em tarefas de aproximação de funções e classificação de padrões, fruto da combinação de propriedades computacionais importantes, tais como não-linearidade, capacidade de aprendizado e generalização.

Como toda rede neural supervisionada, redes não-recorrentes necessitam de uma fonte externa que forneça informação sobre o problema em questão. Esta informação é fornecida através de um conjunto de N pares de vetores $\{\mathbf{x}(n), \mathbf{d}(n)\}$, $n = 1, 2, \ldots, N$, em que $\mathbf{x}(n) \in \mathbb{R}^{(p+1)}$ simboliza o vetor de entrada no instante de tempo discreto n e $\mathbf{d}(n) \in \mathbb{R}^m$ denota o vetor de saídas (respostas) desejadas para aquele vetor de entrada.

Cada vetor de entrada é representado como

$$\mathbf{x}(n) = \begin{pmatrix} x_0(n) \\ x_1(n) \\ \vdots \\ x_p(n) \end{pmatrix} = \begin{pmatrix} -1 \\ x_1(n) \\ \vdots \\ x_p(n) \end{pmatrix}, \tag{3.1}$$

em que $p > 0$ denota o número efetivo de variáveis de entrada usadas no problema. O termo "efetivo" é usado aqui porque a componente de entrada $x_0(n) = -1$ não é propriamente uma variável no sentido usual, sendo mantida fixa com o objetivo de permitir uma formulação única para o ajuste dos limiares (*bias*) de ativação dos neurônios nas redes supervisionadas. De modo semelhante, o vetor de saída no instante n é representado da seguinte forma

$$\mathbf{d}(n) = \begin{pmatrix} d_1(n) \\ \vdots \\ d_m(n) \end{pmatrix}, \tag{3.2}$$

em que $m > 0$ indica o número de variáveis de saída da rede neural. Um componente qualquer do vetor de entrada é simbolizado por $x_j(n) \in \mathbb{R}$, enquanto um componente qualquer do vetor de saída é simbolizado como $d_k(n) \in \mathbb{R}$.

Para um dado problema de interesse, considera-se que os vetores, $\mathbf{x}(n)$ e $\mathbf{d}(n)$ estão relacionados segundo alguma relação matemática desconhecida $\mathbf{f}(\cdot)$,

$$\mathbf{d}(n) = \mathbf{f}[\mathbf{x}(n)], \tag{3.3}$$

sendo que o objetivo principal do problema é lançar mão de alguma ferramenta matemática que possa emular o comportamento de $\mathbf{f}(\cdot)$, com base apenas nos pares de vetores $\{\mathbf{x}(n), \mathbf{d}(n)\}$ disponíveis.

Para isto pode-se utilizar uma rede neural supervisionada e não-recorrente para gerar um modelo matemático que atue como uma aproximação do mapeamento $\mathbf{f}(\cdot)$, denotada por $\hat{\mathbf{f}}(\cdot)$

$$\mathbf{y}(n) = \hat{\mathbf{f}}[\mathbf{x}(n)], \tag{3.4}$$

em que se espera que a saída gerada pela rede neural $\mathbf{y}(n)$ seja muito próxima da saída desejada $\mathbf{d}(n)$.

Redes neurais *feedforward* são aproximadores universais de funções (CYBENKO, 1989; HORNIK, 1991; HORNIK; STINCHCOMBE; WHITE, 1989), ou seja, são capazes de

aproximar mapeamentos entrada-saída não-lineares, tais como aqueles genericamente descritos pela Equação (3.4), com grau de precisão arbitrário, sejam tais mapeamentos contínuos ou descontínuos. Esta propriedade é uma das responsáveis pela ampla popularização do uso de redes neurais artificiais em tarefas de reconhecimento de padrões e aproximação de funções. Assim, vale destacar que a formulação geral do problema de aprendizado de uma rede neural se aplica a arquiteturas de redes neurais envolvidas tanto em tarefas de aproximação de funções, quanto em tarefas de classificação de padrões.

As arquiteturas de redes neurais a serem descritas neste capítulo são aplicadas unicamente em problemas de predição e modelagem de séries temporais, problema este caracterizado como uma modalidade de problema de aproximação de função. Na próxima seção é apresentada uma das mais utilizadas arquiteturas de redes neurais não-recorrentes, e que também é usada como base para a construção de grande parte das arquiteturas recorrentes mais conhecidas.

3.3.1 Rede Perceptron Multicamadas

Tipicamente, uma rede Perceptron Multicamada (*Multilayer Perceptron*, MLP) é constituída de uma camada de entrada que recebe os sinais, uma ou mais camadas intermediárias, compostas por neurônios somadores com função de ativação não-linear e uma camada de saída, também composta por neurônios somadores, embora estes possam ter funções de ativação lineares.

As camadas intermediárias são comumente chamadas de camadas escondidas ou ocultas, visto que os neurônios nelas localizados não têm acesso direto aos sinais da entrada nem da saída. A existência de camadas ocultas não-lineares confere à rede MLP o poder computacional de resolver problemas complexos, pois tais camadas têm a função de promover sucessivas alterações na representação dos dados originais até que o problema possa ser resolvido pela última camada de neurônios (camada de saída).

Outra característica da rede MLP é seu alto grau de conectividade, determinado pelas sinapses da rede, interligações entre os neurônios de diferentes camadas, em que cada uma delas está associada a um valor numérico chamado de peso sináptico. A Figura 8 mostra a arquitetura geral de uma rede MLP de uma única camada oculta. Uma vez especificado o número de camadas e a quantidade de neurônios em cada uma delas, o processo de aprendizado da rede MLP é realizada através do ajuste dos pesos sinápticos e limiares de ativação por meio do algoritmo de retropropagação do erro (*Error Backpropagation*). Este algoritmo de treinamento é

detalhado a seguir.

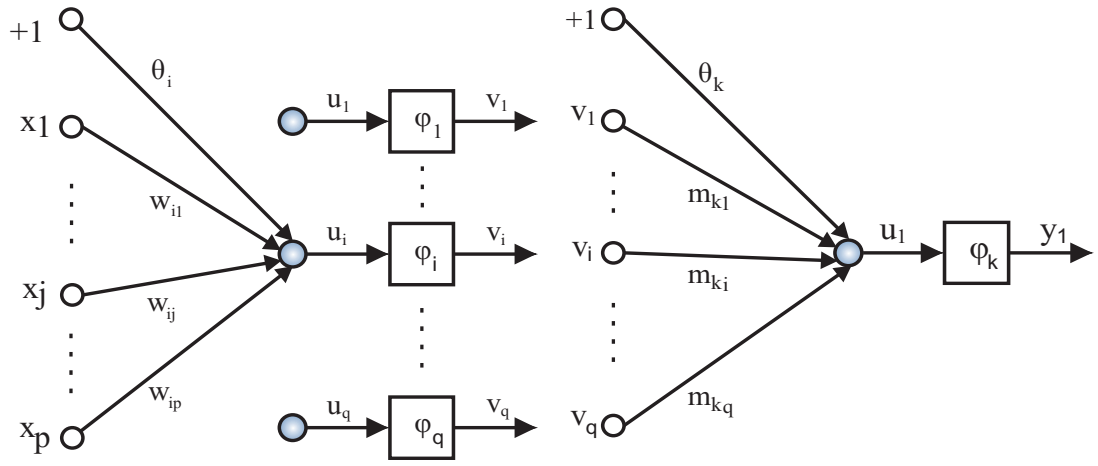


Figura 8 – (a) Neurônios da camada oculta; (b) neurônios de saída.

3.3.1.1 Algoritmo de Retropropagação do Erro

O vetor de pesos associado ao *i*-ésimo neurônio da camada oculta é representado como

$$\mathbf{w}_i(n) = \begin{pmatrix} w_{i0}(n) \\ \vdots \\ w_{ip}(n) \end{pmatrix} = \begin{pmatrix} \theta_i(n) \\ \vdots \\ w_{ip}(n) \end{pmatrix}, \tag{3.5}$$

em que w_{ij} é o peso sináptico conectado a j-ésima entrada ao i-ésimo neurônio da camada oculta e $\theta_i(n)$ é o limiar (*threshold*) associado ao neurônio i.

De modo semelhante, o vetor de pesos associado ao k-ésimo neurônio da camada de saída é representado da seguinte forma

$$\mathbf{m}_k(n) = \begin{pmatrix} m_{k0}(n) \\ \vdots \\ m_{kq}(n) \end{pmatrix} = \begin{pmatrix} \theta_k(n) \\ \vdots \\ m_{kq}(n) \end{pmatrix}, \tag{3.6}$$

na qual m_{ki} é o peso sináptico conectando o i-ésimo neurônio da camada oculta ao k-ésimo neurônio da camada de saída e $\theta_k(n)$ é o limiar associado ao neurônio de saída k. O número de neurônios da camada oculta é denotado por q, $q \geq 2$.

Para cada vetor de entrada apresentado à entrada da rede MLP no instante n, o ajuste dos parâmetros se dá em duas fases: uma direta e outra reversa.

Sentido Direto: esta etapa de funcionamento do algoritmo *backpropagation* envolve o cálculo das ativações e saídas de todos os neurônios da camada oculta e de todos os neurônios da camada de saída. Assim, o fluxo de sinais (informação) se dá dos neurônios de entrada para os neurônios de saída, passando obviamente pelos neurônios da camada oculta. Por isto, diz-se que a informação está se propagando no sentido direto, ou seja,

$$\text{Entrada} \rightarrow \text{Camada Intermediária} \rightarrow \text{Camada de Saída}.$$

Assim, após a apresentação de um vetor de entrada $\mathbf{x}$, na iteração n, o primeiro passo é calcular as ativações dos neurônios da camada oculta

$$u_i(n) = \sum_{j=0}^{p} w_{ij}(n)x_j(n) = \mathbf{w}_i^T(n)\mathbf{x}(n), \quad i = 1,\ldots,q, \tag{3.7}$$

em que T denota a operação de transposição dos vetores e q indica o número de neurônios da camada oculta. Em seguida, as saídas correspondentes são calculadas por meio das seguintes equações

$$v_i(n) = \phi[u_i(n)] = \phi\left[\sum_{j=0}^{p} w_{ij}(n)x_j(n)\right] = \phi\left[\mathbf{w}_i^T(n)\mathbf{x}(n)\right], \tag{3.8}$$

em que para este trabalho $\phi(\cdot)$ é definida pela função *tangente hiperbólica*:

$$\phi[u_i(n)] = \frac{1-\exp[-u_i(n)]}{1+\exp[-u_i(n)]}. \tag{3.9}$$

O segundo passo consiste em repetir as operações das Equações (3.7) e (3.8) para os neurônios da camada de saída, ou seja

$$u_k(n) = \sum_{i=0}^{q} m_{ki}(n)v_i(n), \qquad k = 1,\ldots,m, \tag{3.10}$$

na qual $m \geq 1$ é o número de neurônios de saída. Em seguida, as saídas dos neurônios da última camada são calculadas pela seguinte equação

$$y_k(n) = \phi[u_k(n)] = \phi\left[\sum_{i=0}^{q} m_{ki}(n)v_i(n)\right], \qquad k = 1,\ldots,m, \tag{3.11}$$

tal que a função de ativação $\phi(\cdot)$ assume a forma definida na Equação (3.9).

Sentido Reverso: Esta etapa de funcionamento do algoritmo *backpropagation* envolve o cálculo de gradientes locais e o ajuste dos pesos de todos os neurônios da camada oculta e da camada de saída. Assim, o fluxo de informação se dá dos neurônios de saída para os neurônios da camada oculta. Por isto, diz-se que a informação está se propagando no sentido reverso, ou seja,

Camada de Saída $\rightarrow$ Camada Oculta.

Assim, após os cálculos das ativações e saídas na fase direta, o primeiro passo da fase reversa consiste em calcular os gradientes locais $\delta_k(n)$ dos neurônios da camada de saída

$$\delta_k(n) = e_k(n)\phi'[u_k(n)]. \tag{3.12}$$

em que $e_k(n)$ é o erro entre a saída desejada $d_k(n)$ para o k-ésimo neurônio da camada de saída e a resposta gerada por ele, $y_k(n)$:

$$e_k(n) = d_k(n) - y_k(n). \tag{3.13}$$

A derivada $\phi'[u_k(n)]$ da função tangente hiperbólica, requerida na Equação (3.12), é dada por

$$\phi'[u_k(n)] = \frac{1}{2}\left[1 - y_k^2(n)\right]. \tag{3.14}$$

O segundo passo da fase reversa consiste em calcular os gradientes locais $\delta_i(n)$, dos neurônios da camada oculta

$$\delta_i(n) = \phi'[u_i(n)] \sum_{k=1}^{n} m_{ki}\delta_k(n), \quad i = 1, \ldots, q, \tag{3.15}$$

tal que a derivada $\phi'[u_i(n)]$ é calculada através da Equação (3.14).

O terceiro passo da fase reversa corresponde ao processo de atualização ou ajuste dos parâmetros (pesos sinápticos e limiares) da rede MLP com uma camada oculta. Assim, a regra de atualização dos pesos, w_{ij}, que correspondem aos pesos entre a entrada e a camada de saída, é dada por

$$w_{ij}(n+1) = w_{ij}(n) + \eta\,\delta_i(n)x_j(n), \tag{3.16}$$

em que η é a taxa de aprendizagem. E para os pesos que ligam a camada oculta com a de saída, tem-se que a regra de atualização é dada por

$$m_{ki}(n+1) = m_{ki}(n) + \eta\,\delta_k(n)y_i(n). \tag{3.17}$$

O algoritmo de retropropagação do erro é descrito acima para uma rede MLP com uma única camada de neurônios escondidos, mas o mesmo pode ser generalizado para redes com duas ou mais camadas ocultas sem muito esforço. Redes de uma camada oculta são capazes

de aproximar com precisão arbitrária funções contínuas, enquanto redes MLP de duas ou mais camadas podem aproximar até funções descontínuas.

Da próxima seção em diante começam a ser apresentadas as arquiteturas neurais desenvolvidas a partir da rede MLP especialmente para lidar com problemas de predição não-linear de séries temporais. Estas redes são chamadas genericamente de redes neurais dinâmicas. Primeiramente são apresentadas redes dinâmicas não-recorrentes e, em seguida, é descrito as redes dinâmicas recorrentes.

3.4 RNAs Não-Recorrentes Dinâmicas

O problema de predição não-linear de séries temporais é geralmente colocado na forma de um problema de aproximação de funções. Assim, o interesse em usar a rede MLP para predição está fundamentado justamente na capacidade de aproximação universal desta arquitetura neural. Além disto, para ter bom desempenho em tarefas de predição de séries temporais, a rede MLP deve também ser capaz de representar a dinâmica temporal (relações de causa-e-efeito) do processo não-linear que gerou a série temporal e que está implicitamente representada nesta.

Uma rede neural capaz de modelar a dinâmica de um processo, a partir de uma série temporal, é chamada de rede neural dinâmica, caso contrário, a rede é dita estática. Uma rede neural pode ser concebida como dinâmica ou ser tornada dinâmica a partir de uma rede estática. As arquiteturas de redes neurais dinâmicas são, em sua grande maioria, extensões da rede MLP, que, por sua vez, é originalmente uma rede estática, ou seja, voltada para o processamento de dados estáticos provenientes de sistemas sem memória.

A rede MLP e suas variantes dinâmicas são tornadas sensíveis à estrutura temporal dos sinais portadores de informações através da inclusão de mecanismos de memória de curta duração (*short-term memory*, STM). Estes mecanismos são responsáveis por manter a informação temporal disponível por vários instantes de tempo, a fim de que a rede MLP possa manipulá-la e armazená-la adequadamente em seus pesos sinápticos. Uma forma simples de inserir memória de curta duração dá-se através de atrasadores (*time delays*) ou laços de realimentação (*feedback loops*), que podem ser inseridos tanto interna quanto externamente à rede.

A Figura 9 ilustra a arquitetura geral de uma rede neural estática com memória de curta duração externa. O uso de atrasadores como mecanismos de memória de curta duração da rede MLP dá origem às chamadas redes dinâmicas não-recorrentes, enquanto que o uso de laços de realimentação dá origem às redes dinâmicas recorrentes. Dentre as principais redes

dinâmicas não-recorrentes, pode-se destacar a FTDNN (*Focused Time Delay Neural Network*) e a FIR-MLP (*Finite Impulse Response Multilayer Perceptron*) (WAN, 1990).

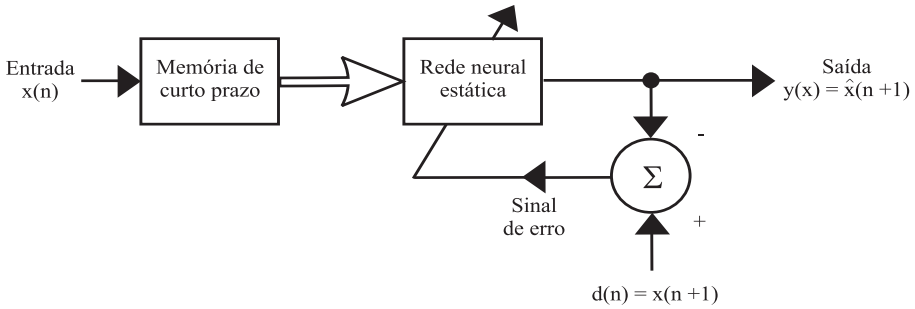


Figura 9 – Arquitetura genérica de uma rede neural dinâmica construída a partir de uma rede neural estática por meio de mecanismos externos de memória de curta duração.

3.4.1 Rede MLP com Atrasadores na Entrada

A Figura 10 ilustra como um número finito de atrasadores pode ser colocado na entrada de uma rede neural a fim de torná-la dinâmica.

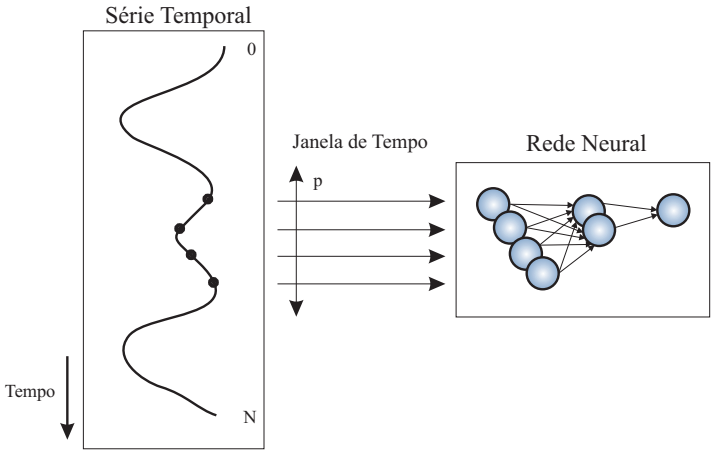


Figura 10 – Exemplo de atrasadores formando uma janela de tempo de comprimento na entrada de uma rede neural.

A janela de tempo formada pelos atrasadores percorre toda a extensão da série temporal, a fim de converter o sinal unidimensional $\{x(n)\}_{t=1}^{N}$ em $N-p-1$ vetores de dimensão $p+1$, sendo estes vetores apresentados na entrada da rede neural. Uma pergunta importante é como selecionar o comprimento p da janela, de modo a capturar adequadamente as propriedades de uma série temporal. Neste trabalho, a janela de tempo é construída de acordo com o teorema de imersão de Takens, discutido na Secção 2.7.

O teorema de imersão fornece a base teórica para predição de séries temporais

não-lineares, onde a relação da predição em relação ao estado atual $\mathbf{x}(n)$ e o próximo valor da série temporal é dada pela seguinte equação:

$$x(n+1) = f[\mathbf{x}(n)] \tag{3.18}$$

Uma vez que a dimensão de imersão d_E e o atraso τ tenham sido escolhidos, a tarefa restante é aproximar uma função de mapeamento $f(\cdot)$. Tem sido demonstrado que uma rede neural *feedforward* com neurônios suficientes é capaz de aproximar qualquer função não-linear para um grau de precisão arbitrária. Assim, pode-se fornecer uma boa aproximação para a função $f(\cdot)$ implementando o seguinte mapeamento:

$$\widehat{x}(n+1) = \widehat{f}[\mathbf{x}(n)] \tag{3.19}$$

onde $\widehat{x}(n+1)$ é uma estimativa de $x(n+1)$ e $\widehat{f}(\cdot)$ é a aproximação correspondente de $f(\cdot)$. O erro de estimação, $e(n+1) = x(n+1) - \widehat{x}(n+1)$, é comumente utilizado para avaliar a qualidade da aproximação.

A janela de tempo, desta forma, é construída de acordo com a Equação (2.17), repetida abaixo para facilitar o entendimento

$$\mathbf{x}(n) = [x(n) \;\; x(n-\tau) \;\; \cdots \;\; x(n-(d_E-1)\tau)]^T, \tag{3.20}$$

em que $\mathbf{x}(n)$ é um vetor que contém d_E elementos da série, contados a partir do elemento atual $x(n)$, espaçados um do outro de τ unidades de tempo. O parâmetro d_E é a dimensão de imersão e o parâmetro τ é o atraso de imersão.

A rede neural dinâmica formada pela introdução de atrasadores na entrada de uma rede MLP estática é conhecida pela sigla FTDNN (*Focused Time Delay Neural Network*) (PRINCIPE; EULIANO; LEFEBVRE, 2000), sendo aqui chamada simplesmente de Rede MLP com Atrasadores na Entrada. Assim, como a rede MLP, que lhe dá origem, a FTDNN é uma rede *feedforward* multicamadas cujos pesos sinápticos e limiares são ajustados de acordo com o algoritmo *backpropagation*.

A arquitetura de uma rede FTDNN com uma camada oculta está mostrada na Figura 11. Nesta figura, o vetor de entrada é definido de acordo com a Equação (3.20), sendo os parâmetros da rede FTDNN modificados a fim de minimizar o erro quadrático médio entre a saída da rede, $y(n) = \widehat{x}(n+1)$ e a resposta desejada $x(n+1)$.

Para o tipo de problema de predição de séries temporais que se está interessado neste trabalho, utiliza-se apenas um neurônio na saída da rede. Matematicamente, isto equivale a fazer

$m = 1$ na Equação (3.11). Assim, uma vez treinada a rede FTDNN, sua saída no instante n é calculada pela seguinte expressão:

$$\begin{aligned} y_1(n) &= \hat{x}(n+1) = \phi\left[\sum_{i=1}^{q} m_{1i} v_i(n)\right], \\ &= \phi\left[\sum_{i=1}^{q} m_{1i}\phi\left(\sum_{j=0}^{d_E-1} w_{ij} x(n-j\tau) - \theta_i\right) - \theta_1\right]. \end{aligned} \tag{3.21}$$

É comum encontrar, na literatura, variantes dinâmicas da rede MLP que possuem não só atrasadores externos, como na rede FTDNN, mas também atrasadores internos, ou seja, inseridos dentro da arquitetura da rede. Tais atrasadores internos são colocados nas saídas dos neurônios das camadas ocultas. Este tipo de rede é denotada genericamente pela sigla TDNN (PRINCIPE; EULIANO; LEFEBVRE, 2000; HAYKIN, 1999), sendo proposta inicialmente em Waibel *et al.* (1989). Deve-se frisar, portanto, que a rede FTDNN é um caso particular da rede TDNN em que não há atrasadores internos, apenas atrasadores externos. Daí a razão do termo *focused* na sigla FTDNN, para indicar que a memória de curta duração está "concentrada" na entrada.

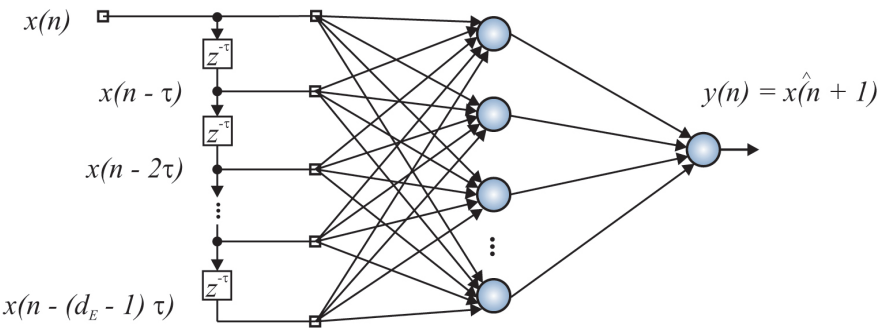


Figura 11 – Arquitetura genérica de uma rede FTDNN de uma camada oculta.

A partir da próxima seção serão apresentadas arquiteturas de redes neurais dinâmicas recorrentes. Tais redes diferem das redes dinâmicas citadas até o presente momento por apresentarem laços de realimentação internos ou externos, também chamados laços locais ou globais, respectivamente.

3.5 RNAs Recorrentes

Uma rede neural dinâmica recorrente, ou simplesmente rede recorrente, é aquela que contém conexões sinápticas realimentadas (ou laços de realimentação), permitindo o fluxo de

sinais de ativação e saídas neurais entre neurônios de camadas distintas, entre neurônios de uma mesma camada, ou ainda de um neurônio para ele mesmo.

Assim como os atrasadores, a recorrência é um tipo de mecanismo de memória de curta duração que permite à rede relembrar informações de um passado recente. A diferença básica entre estes tipos de memória é que, enquanto os atrasadores disponibilizam no instante atual os valores exatos da informação passada, os laços de realimentação realizam algum tipo de processamento (filtragem) sobre a informação passada.

Redes neurais recorrentes constituem uma das mais importantes famílias de arquiteturas das redes neurais e, consequentemente, um grande número de algoritmos foram desenvolvidos com o passar dos anos (NARENDRA; PARTHASARATHY, 1990; HERTZ; KROGH; PALMER, 1991; HAYKIN, 1999; HORNE; GILES, 1995; TSOI; BACK, 1997; PRINCIPE; EULIANO; LEFEBVRE, 2000). Tsoi e Back (1997) discutem e listam como diversas arquiteturas neurais recorrentes podem ser geradas pelas mais variadas combinações de realimentações entre neurônios de uma mesma camada e entre neurônios de diferentes camadas, de tal forma que pode-se facilmente entender a razão da grande diversidade de arquiteturas recorrentes encontradas na literatura.

Pode-se apresentar o modelo de redes neurais dinâmicas recorrentes sob a forma de equações de variáveis de estado. Assim, considerando um caso especial de uma rede neural em que um vetor $\mathbf{x}(\mathrm{n}) \in \mathbb{R}^p$ representa o vetor de entrada e o vetor $\mathbf{v}(\mathrm{n}) \in \mathbb{R}^q$ representa a saída da camada oculta no tempo n, pode-se então descrever o comportamento dinâmico do modelo de redes dinâmicas recorrentes pelo par acoplado:

$$\mathbf{v}(n+1) = \phi(\mathbf{v}(n), \mathbf{x}(n)), \tag{3.22}$$

$$\mathbf{y}(n) = \phi(\mathbf{v}(n)), \tag{3.23}$$

onde $\phi(\cdot)$ é uma função não-linear que caracteriza a camada oculta e a camada de saída (HAYKIN, 1999). Embora essa seja uma representação de redes neurais dinâmicas recorrentes, no decorrer deste trabalho, a representação escolhida será a do mapeamento entrada-saída, isto é, a que já vinha sendo utilizada nas redes dinâmicas.

De um ponto de vista prático, é importante procurar uma resposta à seguinte pergunta: que rede neural dinâmica deve ser usada para tratar com o complexo problema de predição e modelagem não-linear de séries temporais? Recorrente ou não-recorrente? Infelizmente[2], não

[2] Ou felizmente, para quem gosta de diversidade!

há resposta fácil a esta pergunta. A arquitetura apropriada é dependente do problema e já que várias co-existem na literatura, resta ao usuário experimentar e propor um bom número delas, antes de encontrar a(s) arquitetura(s) apropriada(s). Esta postura é adotada nesta tese, em que o desempenho de diversas redes neurais dinâmicas, recorrentes ou não, são testadas no problema supracitado.

3.5.1 Tipos de Conexão de Realimentação

Uma conexão sináptica é definida como uma ligação entre dois neurônios quaisquer. Existem dois tipos maiores de conexões, a saber: conexão de alimentação direta (*feedforward*) e conexão de realimentação (*feedback*). A conexão de alimentação direta ocorre quando um sinal tem orientação da entrada para a saída. Em contraste, a conexão de realimentação tem orientação da saída para a entrada. Desta forma, as conexões realimentam para uma dada camada (ou parte dela) sinais de ativação/saída produzidos por neurônios de outras camadas.

Um modo de classificar redes recorrentes consiste em verificar a extensão espacial das conexões de realimentação existentes, ou seja, se ela envolve apenas os neurônios de uma única camada ou se envolve neurônios de outras camadas. Pode-se enquadrar esta definição em três grupos.

- **Conexão Recorrente Local**: este tipo de conexão envolve apenas um neurônio. Neste caso, o termo local refere-se ao fato de a saída do neurônio ser realimentada para a entrada deste mesmo neurônio. É importante salientar que não é possível ter uma conexão local de alimentação direta. As conexões de alimentação direta devem necessariamente envolver dois neurônios diferentes.
- **Conexão Recorrente Global**: este tipo de conexão acontece entre um neurônio de uma camada para um neurônio de uma camada anterior, ou seja, um sinal de saída de um neurônio é realimentado para a entrada de um outro neurônio localizado em uma camada anterior.
- **Conexão Recorrente Não-Local**: Este é um tipo especial de conexão global, visto que envolve neurônios distintos, porém a conexão é estabelecida entre neurônios de uma mesma camada. Assim, a saída de um neurônio de uma certa camada é realimentada para a entrada de um outro neurônio da mesma camada.

Tendo em vista todas as possíveis conexões que podem ocorrer em redes recorrentes, é fácil perceber a grande variedade de arquiteturas que podem ser formadas pela combinação

de tipos diferentes de conexões e com o número de camada de neurônios. Neste trabalho, o problema de predição e modelagem não-linear de séries temporais será também analisado lançando-se mão de redes dinâmicas recorrentes. A seguir são descritas algumas arquiteturas com recorrências globais e locais estudadas nesta tese.

3.5.2 Redes Recorrentes Simples

Assim como as redes FTDNN e FIR-MLP, uma grande parcela das redes recorrentes de maior utilização são extensões da rede MLP convencional. Não fugindo a esta regra, duas importantes arquiteturas com recorrência, Rede Elman e Rede Jordan, são bastante utilizadas na prática, sendo obtidas facilmente a partir da rede MLP. Vale ressaltar que, como os pesos das conexões de realimentação não são ajustáveis, pode-se usar o algoritmo *backpropagation* padrão para treinar tais redes. A este tipo de rede recorrente dá-se o nome de redes recorrentes simples (PRINCIPE; EULIANO; LEFEBVRE, 2000; HAYKIN, 1999; HERTZ; KROGH; PALMER, 1991).

3.5.2.1 Rede Recorrente de Elman

Esta arquitetura recorrente é proposta por Elman (1990), sendo obtida a partir da rede MLP através da redefinição da camada de entrada da rede, que passa a ser dividida em duas partes. A primeira parte corresponde ao vetor de entrada propriamente dito, conforme definido na Equação (3.20). A segunda parte, chamada de unidades de contexto, consiste na cópia das saídas dos neurônios da camada oculta no instante $n-1$. O termo "cópia", na verdade, é implementado computacionalmente através de um conjunto de conexões de realimentação com pesos fixos e iguais a 1. Desta forma, os valores exatos das ativações dos neurônios da camada oculta no instante $n-1$ são utilizados pelas unidades de contexto para compor o vetor de entrada no instante n. A Figura 12 ilustra uma rede recorrente de Elman com uma camada oculta. A entrada e a saída da rede estão definidas de acordo com o problema de predição de séries temporais.

Todas as conexões da rede de Elman são ajustáveis e do tipo *feedforward*, de tal forma que esta arquitetura pode ser treinada pelo algoritmo *backpropagation*. Do ponto de vista das conexões, a rede de Elman possui apenas recorrências globais, sendo a maior parte de suas conexões sinápticas do tipo *feedforward*.

As ativações dos neurônios da primeira camada oculta da rede de Elman são calcula-

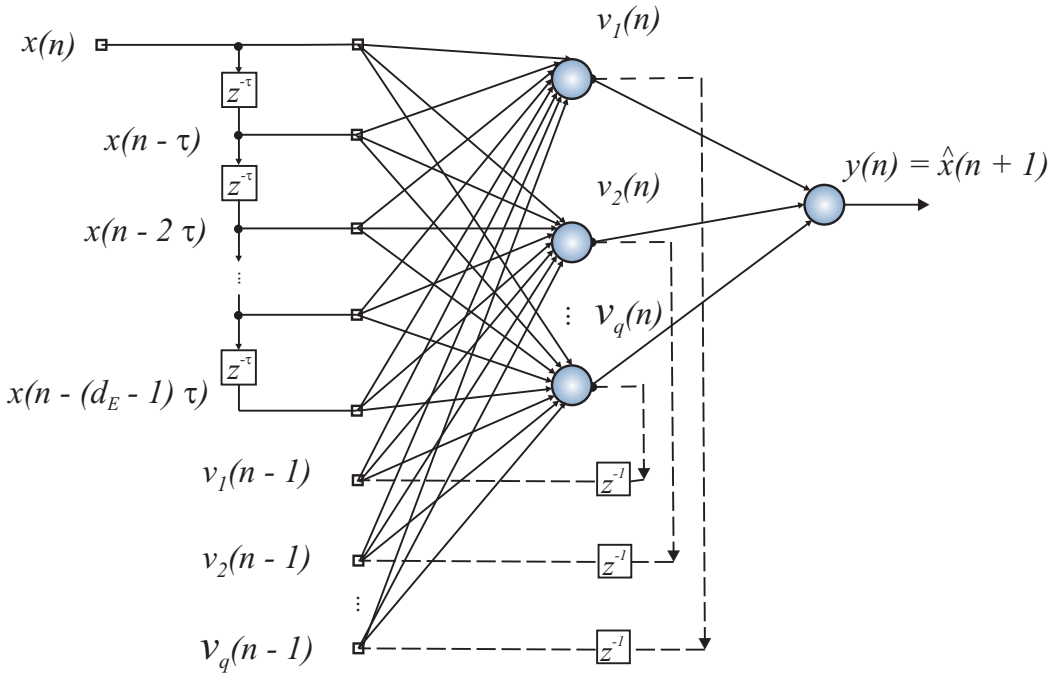


Figura 12 – Arquitetura da rede de Elman aplicada ao problema de predição não-linear de séries temporais.

das por

$$u_i(n) = \sum_{j=0}^{p} w_{ij}(n)x_j(n) + \sum_{l=1}^{q} w_{il}(n)v_l(n-1), \qquad i = 1,\ldots,q, \tag{3.24}$$

tal que a saída dos mesmos é dada por $v_i(n) = \phi[u_i(n)]$. As ativações e as saídas dos neurônios da última camada são calculadas como nas Equações (3.10) e (3.11). Durante o treinamento os pesos w_{ij} e w_{il} são ajustados segundo as regras do algoritmo *backpropagation*.

Para o cálculo do número de parâmetros da rede recorrente de Elman com realimentação das ativações da primeira camada oculta para as unidades de contexto, temos:

$$M = ((p+q+1)\times q) + ((q+1)\times m). \tag{3.25}$$

Como exemplo hipotético, considere as seguintes constantes $p = 4$, $q = 10$ e $m = 1$. Assim, uma rede de Elman, terá $M = 150 + 11 = 161$ parâmetros ajustáveis.

3.5.2.2 Rede Recorrente de Jordan

A rede de Jordan (1986) é outra arquitetura recorrente clássica, sendo inicialmente usada para reconhecimento de sequências temporais. Assim como a rede de Elman, a rede de Jordan também não possui recorrência entre neurônios da mesma camada, sendo por isto enquadrada entre as redes globalmente recorrentes.

Em vez de realimentar as ativações dos neurônios da camada oculta, a rede de Jordan envolve conexões de realimentação dos neurônios da camada de saída para as unidades de contexto. Além disto, este tipo de rede recorrente possui auto-conexões ou auto-realimentações, em que a saída de uma unidade de contexto é realimentada para sua entrada. A Figura 13 ilustra uma rede recorrente de Jordan com uma camada oculta. A entrada e a saída da rede estão definidas de acordo com o problema de predição de séries temporais.

A saída da k-ésima unidade de contexto no instante n, denotada por $C_k(n)$, é dada pela seguinte expressão

$$C_k(n) = \alpha C_k(n-1) + y_k(n-1), \tag{3.26}$$

em que $y_k(n)$ é a resposta do k-ésimo neurônio de saída, calculada como na Equação (3.11), e $0 < \alpha < 1$ é chamado de coeficiente de auto-realimentação. Se a saída $y_k(n)$ for fixada, então C_k decai exponencialmente para $y_k(n)/(1-\alpha)$, esquecendo assim gradualmente valores passados de C_k.

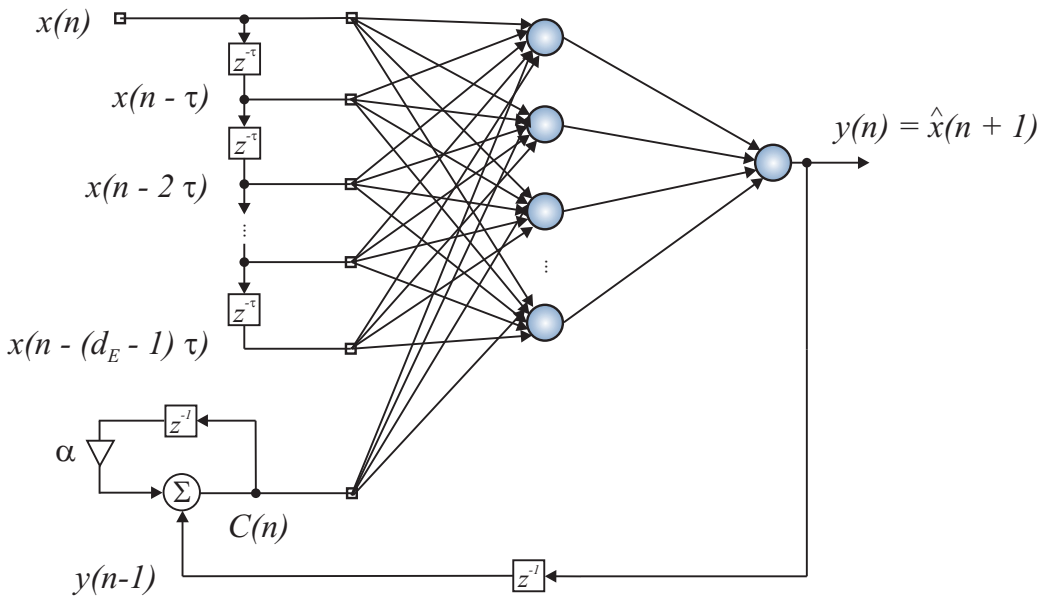


Figura 13 – Arquitetura da rede de Jordan aplicada ao problema de predição não-linear de séries temporais.

Desta forma, a ativação do i-ésimo neurônio da camada oculta é dada por

$$u_i(n) = \sum_{j=0}^{p} w_{ij}(n)x_j(n) + \sum_{k=1}^{m} w_{ik}(n)C_k(n), \qquad i = 1,\ldots,q, \tag{3.27}$$

tal que a saída dos mesmos é dada por $v_i(n) = \phi[u_i(n)]$. As ativações e as saídas dos neurônios da última camada são calculadas como nas Equações (3.10) e (3.11). Durante o treinamento, os pesos w_{ij} e w_{il} são ajustados segundo as regras do algoritmo *backpropagation*.

Para o cálculo do número de parâmetros da rede recorrente de Jordan temos:

$$M = ((p+1+m) \times q) + ((q+1) \times m). \tag{3.28}$$

Como exemplo hipotético, considere as seguintes constantes $p = 4$, $q = 10$ e $m = 1$. Assim, uma rede de Jordan, terá $M = 60 + 11 = 71$ parâmetros ajustáveis.

As redes dinâmicas descritas até agora, recorrentes ou não, são relativamente fáceis de aplicar ao problema de predição e modelagem não-linear de séries temporais. Para este fim, basta definir o vetor de entrada como na Equação (3.20) e definir uma rede com um único neurônio de saída ($m = 1$) cuja saída desejada durante o treinamento é dada pelo próximo valor da série, ou seja, $d(n) = x(n+1)$. Durante o teste, a saída da rede fornece uma estimativa do próximo valor da série, ou seja, $y(n) = \hat{x}(n+1)$. A seguir é descrita uma rede dinâmica, que pode funcionar de modo recorrente ou não, que foi originalmente proposta para lidar não com predição/modelagem não-linear de séries temporais, mas sim com problemas um pouco mais gerais, genericamente chamados de identificação de sistemas não-lineares.

3.5.3 Extinção de Gradientes

Um problema que requer atenção em aplicações práticas de uma rede recorrente é a extinção dos gradientes, relativo ao treinamento das redes para produzir uma resposta desejada no tempo corrente que depende dos dados de entrada no passado distante. A questão é que, por causa da combinação das não-linearidades, uma modificação infinitesinal de uma entrada distante no tempo pode não ter quase efeito no treinamento da rede. O problema pode surgir mesmo se uma grande modificação na entrada distante no tempo tiver algum efeito, mas se este efeito não for mensurável pelo gradiente. O problema da extinção dos gradientes torna a aprendizagem de dependências a longo prazo em algoritmos de treinamento baseados em gradiente difícil, se não virtualmente impossível, em certos casos (BENGIO; SIMARD; FRASCONI, 1994).

Haykin (1999) aponta alguns procedimentos possíveis para aliviar as dificuldades que surgem devido à extinção dos gradientes em redes recorrentes. Dentre eles, pode-se destacar o aumento da abrangência temporal das dependências de entrada-saída apresentando-se à rede durante o treinamento, em primeiro lugar, as sequências mais curtas de símbolos; uso do filtro de Kalman estendido ou sua versão desacoplada para um uso mais eficiente da informação disponível aos algoritmos de aprendizagem baseados em gradiente e por fim o uso de atrasos de tempo nas arquiteturas de redes recorrentes. Este último procedimento faz uso de uma classe de

redes neurais conhecida como modelos NARX e são tema do Capítulo 5 desta tese.

3.6 Conclusão

Este capítulo apresentou sucintamente, porém de forma auto-contida, as principais arquiteturas de redes neurais dinâmicas avaliadas nesta tese. Grosso modo, estas redes podem ser classificadas em redes dinâmicas recorrentes e não-recorrentes de acordo com a presença ou não de conexões de realimentação:

Redes Dinâmicas Não-Recorrentes - Redes FTDNN e FIR-MLP;

Redes Dinâmicas Recorrentes - Redes de Elman e Jordan.

Todas estas arquiteturas, derivadas da rede MLP a partir da introdução de mecanismos de memória de curta duração, cobrem uma parcela razoável das técnicas neurais utilizadas em problemas de modelagem de sistemas dinâmicos não-lineares.

No próximo capítulo, é discutida a metodologia utilizada para a predição de séries temporais. Primeiramente as séries temporais utilizadas são apresentadas, como também os índices de desempenho utilizados para quantificar as predições. Por fim, alguns itens importantes para o bom treinamento das redes neurais são numerados e é proposto um método de obtenção do modelo mais adequado de uma RNA.

4 METODOLOGIAS DE PROJETO E AVALIAÇÃO

4.1 Introdução

Neste capítulo são apresentadas as metodologias utilizadas nesta tese para projeto e avaliação de arquiteturas de RNAs para predição de séries temporais. Ao longo do capítulo serão abordados os principais índices de desempenho utilizados para quantificar o quão satisfatório são os resultados das predições dos modelos. Em especial dar-se-á ênfase na predição h-passos-adiante, isto é, predição recursiva. São descritas cinco séries temporais univariadas: precipitação de chuvas em Fortaleza, série caótica de Hénon, série do Laser Caótico e série de Mackey-Glass. Logo após a especificação dos parâmetros de treinamento de uma rede neural aplicada na tarefa de predição de séries temporais, um método para obtenção da configuração mais adequada das RNAs é descrito.

Todos os algoritmos apresentados neste trabalho foram implementados em linguagem C++. Para este fim foi utilizada a biblioteca matemática e de processamento de sinais conhecida como IT++[1], focado para cálculo com matrizes e processamento de sinais. Na implementação dos algoritmos foram utilizados microcomputadores com processadores de núcleo duplo, com auxílio de processamento paralelo por meio do acesso remoto de outras máquinas similares com o uso da linguagem *shell script*[2].

4.2 Descrição dos Dados

4.2.1 Série Caótica de Hénon

O sistema de Hénon é definido pelo seguinte mapa de tempo discreto (HéNON, 1976)

$$\begin{aligned} s_1(n+1) &= s_2(n) + 1 - a \cdot s_1^2(n), \\ s_2(n+1) &= b \cdot s_1(n). \end{aligned} \tag{4.1}$$

Com o valor de $b = 0$, este mapa se reduz ao sistema caótico conhecido como mapa logístico. Para comportamento caótico, este sistema possui uma pequena faixa de valores para a e b, sendo

[1] Website: *http://sourceforge.net/apps/wordpress/itpp/*

[2] Shell Script é uma ferramenta de automação de instruções. É capaz de executar uma sequência de operações, instruções e testes a partir de um arquivo de texto executável.

que os valores mais usuais para produzir um sistema caótico são a =1,4 e b =0,3 (KANTZ; SCHREIBER, 1997).

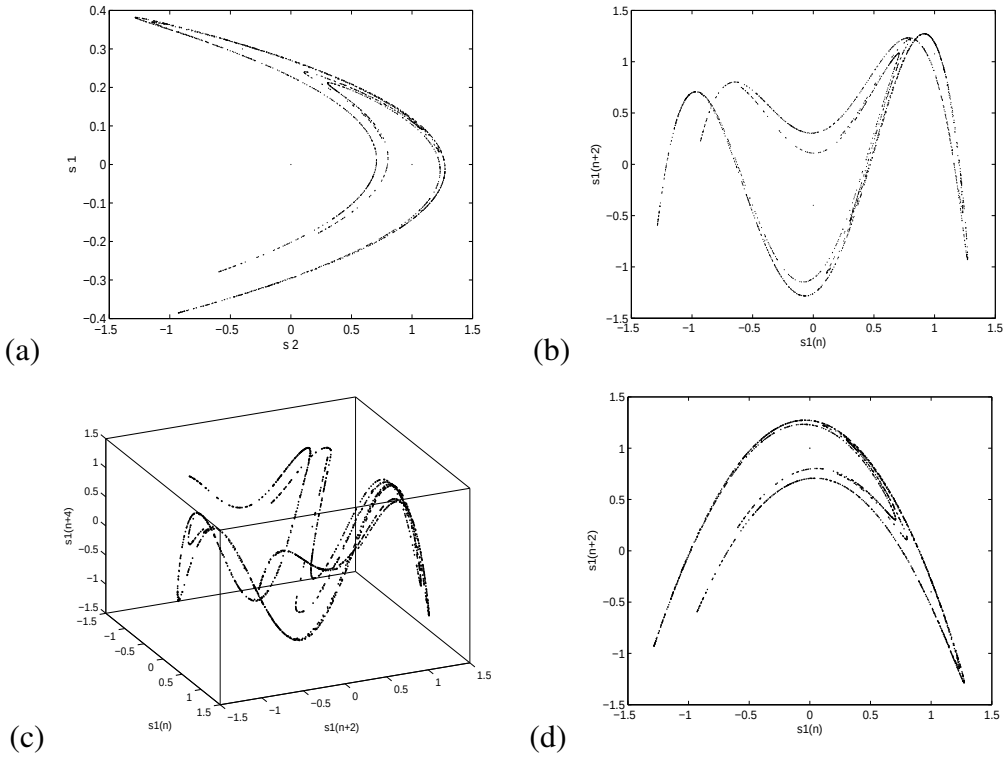


Figura 14 – Reconstrução do atrator de Hénon. (a) atrator original; (b) atrator reconstruído para $\tau = 2$ e $n_d = 2$; (c) $\tau = 2$ e $n_d = 3$; (d) $\tau = 1$ e $n_d = 2$.

Na Figura 14(a) é mostrado o atrator original do mapa com os valores discutidos anteriormente. Na Figura 14(b) é reconstruído o atrator da medida s_1, usando uma dimensão de imersão d_E = 2 e um atraso de imersão τ = 2, resultando em coordenadas de atraso $\mathbf{x}(n) = [x(n) \quad x(n+2)]^T$. É possível notar que existem intercessões na reconstrução do atrator e que estas desaparecem quando se aumenta o valor da dimensão de imersão para 3 (Figura 14(c)). Também é interessante verificar que para uma escolha de coordenadas tal como $\mathbf{x}(n) = [x(n) \quad x(n+1)]^T$, isto é, d_E = 2 e τ = 1, o atrator também é reconstruído sem intercessões, como mostrado na Figura 14(d).

4.2.2 Série de Precipitação de Chuvas

A série temporal consiste na acumulação mensal de precipitação (em milímetros) de chuva na cidade de Fortaleza, capital do estado do Ceará, localizada na região Nordeste do Brasil. Os dados foram fornecidos pela FUNCEME (Fundação Cearense de Meteorologia e Recursos Hídricos) e coletados de janeiro de 1974 até dezembro de 2007, resultando em 408 observações

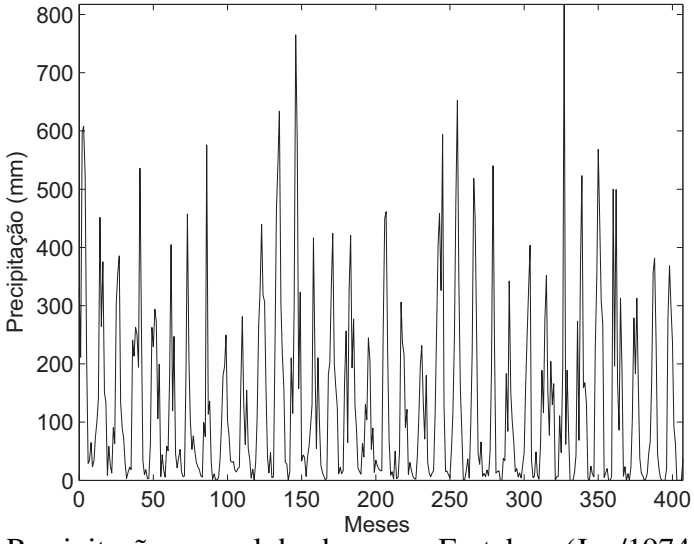


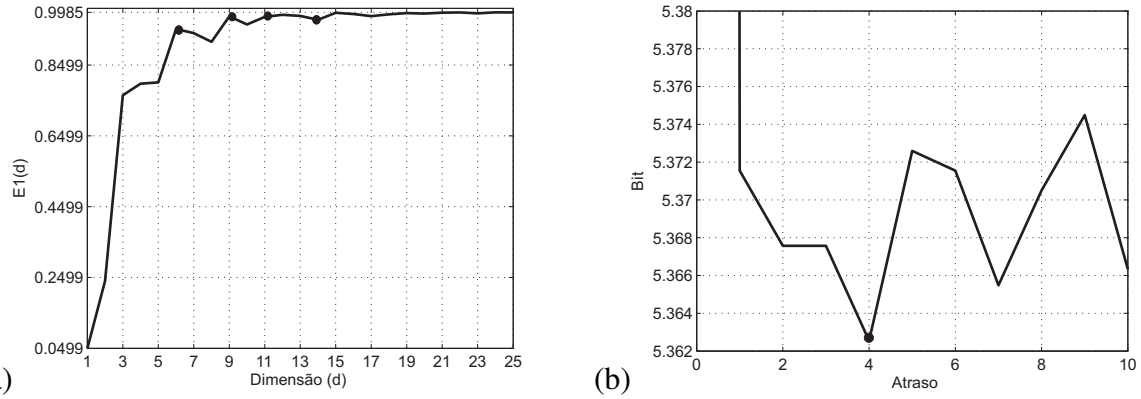
Figura 15 – Precipitação mensal de chuva em Fortaleza (Jan/1974-Dec/2007).

(Figura 15).

Melhorar o índice de acerto das predições do tempo é um dos principais objetivos e desafios da modelagem numérica da atmosfera utilizada nos centros de meteorologia. O desenvolvimento tecnológico verificado nos últimos anos tornou possível aumentar a complexidade dos modelos e a representação dos processos físicos da atmosfera, além de fornecer uma maior quantidade de dados para a geração das condições iniciais dos modelos.

O prognóstico de chuvas, principalmente no período chuvoso no ceará, para os meses de fevereiro, março, abril e maio, é de grande importância para a sociedade, pois é utilizado na prática do processo de tomada de decisão da gerência dos recursos hídricos, na agricultura, turismo, processos erosivos e demais setores da sociedade, sejam eles da esfera privada ou governamental.

A dimensão de imersão (d_E) é estimada pelo método de Cao (CAO, 1997), que é a variante do método de falsos vizinhos (Seção 2.7.1). A curva gerada pelo método de Cao é mostrada na Figura 16(a). O valor recomendado para ser escolhido é o que está próximo do "joelho" da curva (i.e. $d_E = \{6, 9, 11, 14\}$). O atraso de imersão é estimado como $\tau = 4$, obtido pelo método da informação mútua, proposto por Fraser e Swinney (1986). Este método propõe que o primeiro mínimo da curva de informação mútua (ver Figura 16(b)) seja usado como uma boa estimativa para τ.



(a) (b)

Figura 16 – Série de precipitação de chuvas(a) curva do método de Cao para estimação da dimensão de imersão, (b) curva de informação mútua para estimação do atraso de imersão.

4.2.3 Série Caótica de Mackey-Glass

Observações da série temporal de Mackey-Glass são produzidas por uma equação diferencial com atrasos de tempo (Δ) (MACKEY; GLASS, 1977)

$$\frac{dy(t)}{dt} = \beta y(t) + \frac{\alpha y(t-\Delta)}{1+y^{10}(t-\Delta)}, \tag{4.2}$$

em que $y(t)$ é o valor da série temporal no instante t. Para $y(0) \in [0,\Delta]$, o sistema converge para um ponto de equilíbrio estável se $\Delta < 4{,}53$, para um ciclo-limite se $\Delta \in$ [4,53 - 13,3], tornando-se caótico para $\Delta > 16{,}8$, após uma série de duplicações de períodos para $\Delta \in$ [13,3 - 16,8]. Para as simulações, os seguintes valores para os parâmetros são utilizados: α = 0,2, β = - 0,1 e Δ = 17.

A série de Mackey-Glass é gerada a partir da solução numérica da Equação (4.2), usando o método de Euler (MENEZES-JÚNIOR, 2006). Adota-se $\triangle t = 1$ e as primeiras amostras geradas são descartadas para eliminar o efeito transitório devido às condições iniciais. Um trecho contendo 300 amostras da série é apresentado na Figura 17.

A Equação (4.2) modela a dinâmica de produção de células brancas (neutrófilos) no corpo humano. Pelo fato de as taxas de proliferação destas células envolverem um atraso de tempo, dinâmicas periódicas e caos podem ser verificadas. Mackey e Glass sugeriram que flutuações de longo prazo no número de células, observadas em certas formas de leucemia, apresentam uma dinâmica semelhante à observada na Equação (4.2) (KAPLAN; GLASS, 1995).

Na Figura 18(a) observa-se o resultado do cálculo da dimensão de imersão estimada pelo método de Cao. Neste caso, o valor escolhido para uma boa reconstrução do atrator foi 5. Este valor está de acordo com o teorema de imersão ($d_E \geq 2[d]+1$), pois sendo o valor da dimensão intrínseca do atrator de Mackey-Glass encontrado em Farmer (1982) igual a 1,95 para Δ igual a 17, uma condição suficiente para o valor da dimensão de imersão seria $d_E = 5$. Na

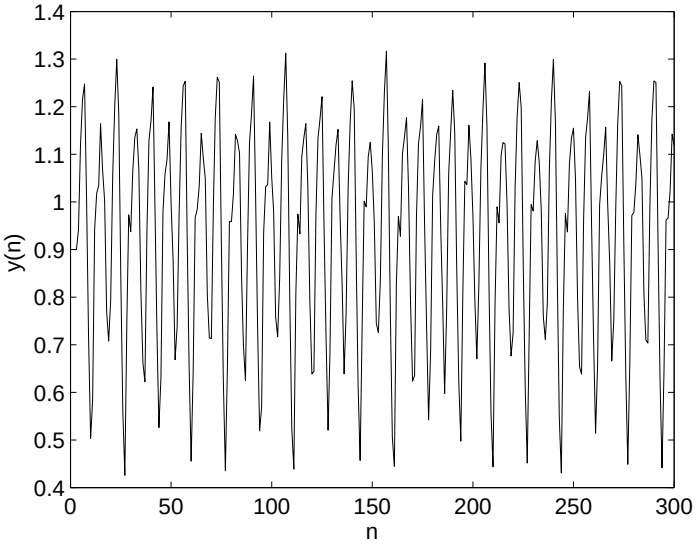
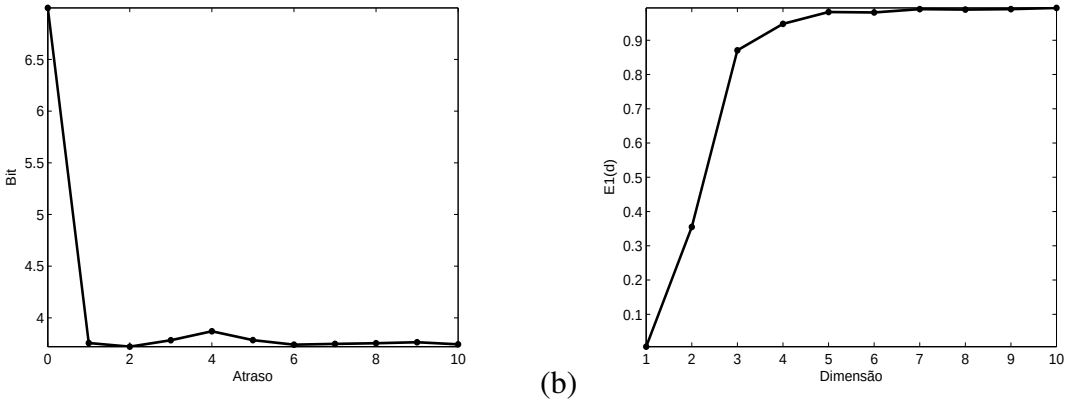


Figura 17 – Série caótica de Mackey-Glass.



(a) (b)

Figura 18 – Série caótica de Mackey-Glass: (a) informação mútua para o cálculo do atraso de imersão; (b) método de Cao para o cálculo da dimensão de imersão.

Figura 18(b) encontra-se o valor igual a 2 para o atraso de imersão calculado pelo método da informação mútua, onde a função da informação mútua atinge o primeiro mínimo.

4.2.4 Série do Laser Caótico

Um outro exemplo de sinal caótico provém de uma sequência de medidas da intensidade de pulsação de um laser de NH_3 infravermelho, obtida de um experimento realizado por Hübner, Abraham e Weiss (1989). Esta série temporal foi disponibilizada inicialmente como parte da competição de predição de séries temporais promovida pelo Instituto Santa Fé, ocorrida nos Estados Unidos em 1992. A Figura 19 contém um trecho contendo 1000 amostras da série caótica do Laser. Na Figura 20(a) determina-se o atraso de imersão igual a 2, utilizando o método da informação mútua, e na Figura 20(b) encontra-se a dimensão de imersão da série real caótica

do Laser como sendo igual a 7.

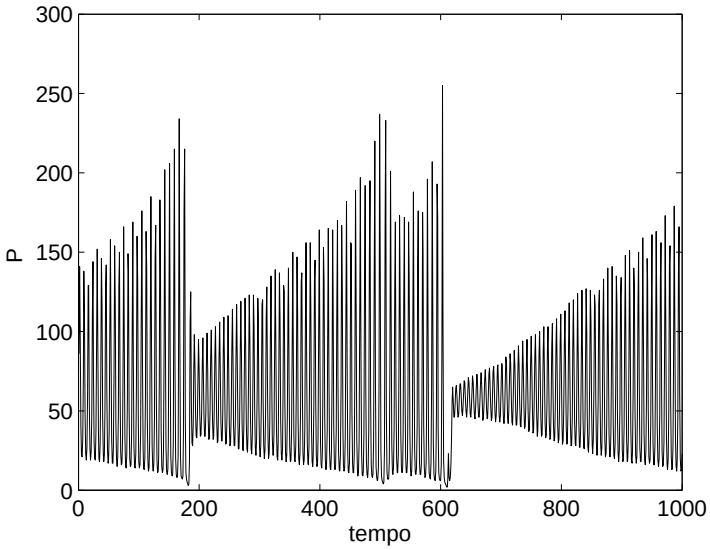
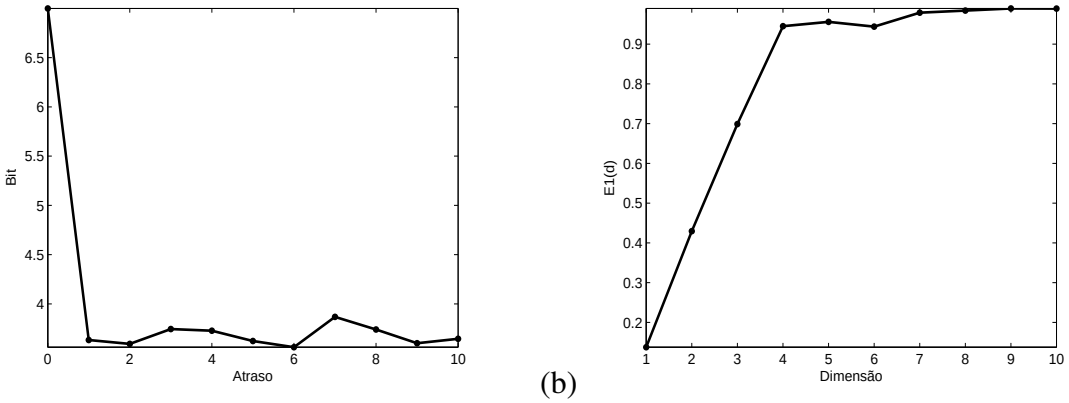


Figura 19 – Série do Laser caótico.

Observando a série do Laser caótico, percebe-se que a potência de saída exibe trechos em que há oscilações regulares de amplitude crescente. Quando um valor crítico é atingido, uma instabilidade ocorre e a oscilação recomeça com uma amplitude de valor mais baixo. Devido a instabilidade, perde-se informação sobre a fase de oscilação e a amplitude se comporta de forma aparentemente imprevisível. A taxa de amostragem dos dados é de oito medidas para cada oscilação. Estas medidas são digitalizadas por um conversor A/D, tal que o erro de discretização tem amplitude igual a $\frac{1}{512}$ da faixa de variação do sinal.



(a) (b)

Figura 20 – Série caótica do Laser: (a) informação mútua para o cálculo do atraso de imersão; (b) método de Cao para o cálculo da dimensão de imersão.

4.3 Índices de Desempenho

Em problemas de predição de séries temporais, uma importante tarefa é a de quantificar a qualidade da predição obtida. Isto permite, por exemplo, comparar diversos algoritmos e diversas estruturas de modelos utilizando índices de desempenho. A seguir são apresentados alguns dos índices de desempenho mais utilizados em problemas de predição de séries temporais.

Primeiramente, como métrica de avaliação do desempenho em tarefa de predição e modelagem de séries temporais, define-se o erro de predição (ou resíduo) como a diferença entre o valor realmente observado para a próxima amostra da série e a estimativa $\widehat{x}(n+1)$ produzida pelo preditor, ou seja,

$$e(n) = x(n) - \widehat{x}(n). \tag{4.3}$$

Uma vez realizado várias predições, o primeiro passo é calcular o erro médio de predição (*Mean Prediction Error*, MPE), que equivale ao conhecido erro quadrático médio (*Mean Square Error*, MSE)

$$\varepsilon^2 = \frac{\sum_{n=1}^{H}(x(n) - \widehat{x}(n))^2}{H}, \tag{4.4}$$

em que H é o comprimento da sequência de exemplos preditos. Valores elevados para ε significam que as predições são ruins e o modelo pode não ser determinístico[3]. Da mesma forma, valores pequenos de ε sugerem que o sistema seja determinístico.

O valor de ε, na Equação (4.4), é um número absoluto, ou seja, por si só não diz se o erro está alto ou baixo. Para decidir o quanto um erro de predição é elevado ou não, deve-se compará-lo com algum valor de referência. Isto é necessário para que modelos distintos possam ser comparados entre si. Para este fim, uma forma bastante utilizada para avaliar a precisão de um modelo é por meio do MSE Normalizado (*Normalized MSE*, NMSE), dado pela seguinte expressão

$$\varepsilon_N^2 = \frac{\varepsilon^2}{\widehat{\sigma}_x^2}, \tag{4.5}$$

em que $\widehat{\sigma}_x^2$ é a variância amostral da série temporal usada para testar o modelo $\widehat{f}(\cdot)$, ou seja,

$$\widehat{\sigma}_x^2 = \frac{\sum_{n=1}^{H}(x(n) - \bar{x})^2}{H}, \tag{4.6}$$

[3] Assume-se aqui que o modelo que gera as predições está adequadamente ajustado!

em que H é o comprimento da série temporal utilizada no teste e $\bar{x}$ é a média amostral desta série. Assim, finalmente, o NMSE pode ser definido pela seguinte expressão:

$$NMSE(H) = \frac{\sum_{n=1}^{H}(x(n) - \widehat{x}(n))^2}{\sum_{n=1}^{H}(x(n) - \bar{x}(n))^2}. \tag{4.7}$$

Comparando as Equações (4.4) e (4.6) percebe-se que a diferença entre elas está somente no segundo termo da diferença. Na expressão do MSE, este termo é $\widehat{x}(n)$, enquanto na expressão da variância é $\bar{x}$. Desta forma, pode-se entender a variância como equivalente ao MSE calculado para o caso em que o modelo gera predições sempre iguais à média amostral. A lógica deste estimador é a seguinte: na dúvida ou na impossibilidade de gerar uma predição mais exata de uma grandeza, adota-se seu valor médio. Esta estratégia é usada, por exemplo, pelas companhias de energia elétrica ou água quando o funcionário que faz a leitura do equipamento medidor não consegue fazê-lo por algum motivo. Na conta de luz/água correspondente àquele mês de leitura não-feita, vem o valor médio dos últimos 12 meses.

Conclui-se então que ao dividir MSE pela variância da série observada, se está na verdade comparando o erro de predição de um dado modelo mais confiável com o erro de predição gerado pelo preditor mais trivial, aquele mostrado na Seção 2.4.1, dado pelo método de média móvel. Quando a predição é considerada boa, tem-se ε_N^2 próximo de zero. Predições ruins geram valores de ε_N^2 próximos de 1, o que significa que o modelo $\widehat{f}(\cdot)$ é tão ruim quanto o modelo que gera predições pelo valor médio.

Uma outra alternativa interessante de avaliação de desempenho é a estatística conhecida como *U de Theil*:

$$U_{theil} = \frac{\sum_{n=1}^{H}(x(n) - \widehat{x}(n))^2}{\sum_{n=1}^{H}(x(n) - x(n-1))^2}. \tag{4.8}$$

Esta estatística é semelhante ao NMSE, com a diferença que, ao invés do preditor trivial utilizar a média temporal do sinal, a estatística *U de Theil* utiliza o valor anterior da série temporal. Deve-se verificar que também se está comparando o erro de predição de um dado modelo com o erro do preditor de média móvel, mas agora utilizando o valor de $r = 1$ na Equação (2.5). Uma comparação entre estes indicadores e vários outros pode ser encontrada em Gooijer e Hyndman (2006).

4.4 Heurística para Encontrar o Melhor Modelo Neural

A determinação do modelo neural é uma tarefa fundamental em qualquer aplicação. Várias métricas são descritas na literatura que oferecem uma indicação das melhores estimativas

dos parâmetros da rede, tais como os encontrados em Charytoniuk e Chen (2000), Hippert, Pedreira e Souza (2001) e Crone e Dhawan (2007) para aplicações com predição de séries temporais e Zhang (2007) para aplicações diversas. No entanto, uma metodologia sistemática para a obtenção dos valores ótimos não é disponível.

Weigend e Gershefeld (1994) afirmam que não é possível construir um algoritmo de predição que seja universal, isto é, que seja capaz de prever qualquer tipo de série temporal. Apesar de óbvia, esta afirmação contribui para que as aspirações de construir modelos de predição devam ser mais modestas. Em vez de buscar o conhecimento completo do futuro, o que se pode ter como objetivo é encontrar o modelo mais adequado para certos tipos de dados, e a definição de "mais adequado" pode ser explicado como o modelo que exige o mínimo de informação para descrever os dados.

A fim de tratar tal problema, nesta tese é proposta uma heurística que realiza uma busca pelos parâmetros de um determinado modelo neural. Esta busca, ao final, irá gerar uma configuração ótima de rede capaz de resolver o problema em questão. A configuração ótima corresponde aos valores apropriados para a dimensão do vetor de entrada, número de épocas de treinamento, número de camadas, número de neurônios em cada camada, bem como o número de saídas para as quais a rede apresenta os melhores resultados. Outras decisões de projeto de rede incluem a seleção de funções de ativação dos neurônios das camadas ocultas e de saída, o algoritmo de treinamento, transformação ou métodos de normalização de dados, conjuntos de treinamento e teste, e medidas de desempenho.

Embora a heurística proposta implemente um algoritmo de busca exaustiva, trata-se de um algoritmo trivial e frequentemente utilizado para tal fim, pois consegue enumerar todos os possíveis melhores parâmetros de um modelo e verificar se eles satisfazem o problema. Entretanto, seu custo computacional é proporcional ao número de candidatos à solução, que, em problemas reais, tende a crescer exponencialmente. Para evitar tal dificuldade na busca dos melhores parâmetros, a heurística reduz o conjunto de candidatos de parâmetros para um espaço de soluções plausíveis.

(*i*) **Métricas Estatísticas**

Uma vez que uma determinada rede tenha sido treinada, ela passa a fornecer estimativas de valores futuros de uma dada série temporal num determinado horizonte de predição h. As predições são executadas de forma recursiva até que o horizonte de predição desejado seja alcançado, ou seja, durante o tempo de h passos os valores previstos são alimentados de volta

para as entradas do modelo.

Por razões de precisão estatística, cada rodada de treino/teste da rede neural é repetida K vezes. Para cada rodada, os pesos são aleatoriamente inicializados na faixa de $[-0,5,+0,5]$. Quantitativamente, para a k-ésima rodada de treino/teste, o modelo é avaliado em termos do NMSE calculado para H passos de tempo de predição:

$$NMSE(H,l) = \frac{1}{H \cdot \sigma_y^2} \sum_{h=1}^{H} \left(x(n+h) - \widehat{x}^{(l)}(n+h) \right)^2, \tag{4.9}$$

onde $x(n+h)$ é o valor observado da série temporal no passo de tempo $n+h$, $\widehat{x}^{(l)}(n+h)$ é o valor predito no passo de tempo $n+h$ para k-th rodada de treino/teste e $\widehat{\sigma}_y^2$ é a variância amostral da série temporal observada.

Todos os resultados utilizam valores da mediana do NMSE de todas as rodadas de treino/teste para um dado horizonte de predição H, ou seja,

$$NMSE_{md} = \text{mediana}[NMSE(H,1)\ \ NMSE(H,2)\ \ \cdots\ \ NMSE(H,K)]. \tag{4.10}$$

O uso da mediana, em vez do valor médio de $NMSE(h,l)$, para um dado h, é preferível, uma vez que a mediana é uma estatística robusta com relação à presença de *outliers* (HUBER, 1981).

(*ii*) **Procedimento Geral**

O fluxograma da Figura 21 apresenta os passos da heurística adotada. A ideia é desenvolver um processo investigativo que implica em variar alguns parâmetros enquanto se monitora a eficiência da predição. Neste caso a otimização é feita utilizando alguns parâmetros, em particular, aqueles que possuem o maior impacto no desempenho da rede MLP treinada com algoritmo *backpropagation*. No presente trabalho, esta heurística também é aplicada a outros tipos de redes neurais ou modelos.

São investigados os valores da dimensão de imersão, atraso de imersão, número de épocas de treinamento, taxa de aprendizagem e número de neurônios em cada camada oculta. A busca se inicia com todos os valores de parâmetros aleatórios e o processo continua variando-se o parâmetro de interesse dentro de uma determinada faixa. O fluxograma da Figura 22 detalha o procedimento que é executado dentro dos blocos 2, 3 e 4 da Figura 21.

(*iii*) **Procedimentos Específicos**

A primeira faixa de valores investigados é do número de épocas de treinamento versus taxa de aprendizagem (bloco 2 da Figura 21). O melhor parâmetro encontrado é aquele dado pelo menor valor da mediana do NMSE, dentro da faixa de busca destes dois parâmetros. Estes

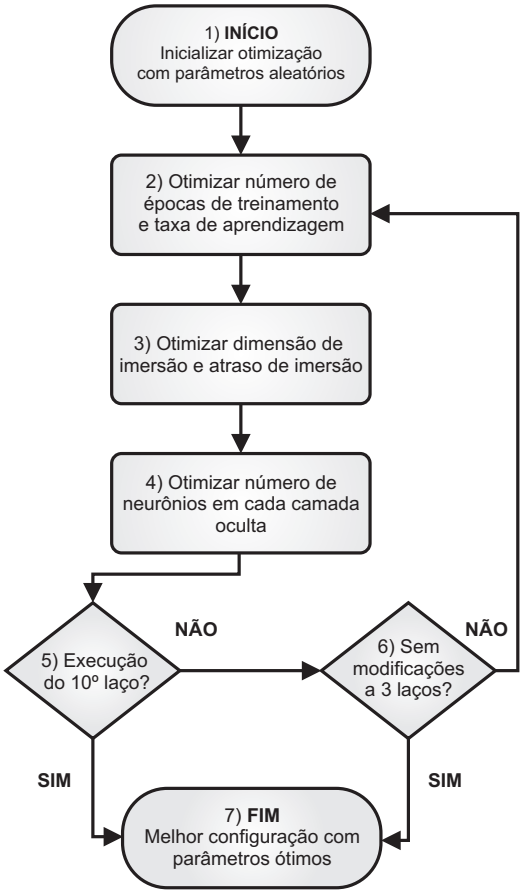


Figura 21 – Procedimento para determinar automaticamente a melhor configuração dos parâmetros da rede neural de interesse.

parâmetros selecionados são guardados e utilizados para a próxima otimização de parâmetros (bloco 3 da Figura 21). A próxima busca tem como meta encontrar os parâmetros do vetor de entrada, isto é, determinar a dimensão e atraso de imersão. Por fim, é investigado o número de neurônios em cada camada oculta dentro de uma certa faixa de valores (bloco 4 da Figura 21), para o caso de uma rede com duas camadas ocultas.

Depois que todos os parâmetros já tenham sido otimizados, o processo é reiniciado com os melhores parâmetros de um ciclo já encontrados. Assim, cada faixa de parâmetro analisada é novamente investigada para a eficiência da predição. Se o NMSE da melhor combinação de uma determinada faixa de parâmetros do novo ciclo não é menor que o do ciclo anterior, usa-se a combinação encontrada anteriormente e os parâmetros não são alterados. Caso contrário, uma nova configuração do modelo é estabelecida e o algoritmo avança para a investigação do próximo conjunto de parâmetros. A heurística continua até que uma condição de parada seja satisfeita.

Vale observar que, para a tarefa de busca, alguns parâmetros foram agrupados em pares: número de épocas de treinamento versus taxa de aprendizagem, dimensão de imersão

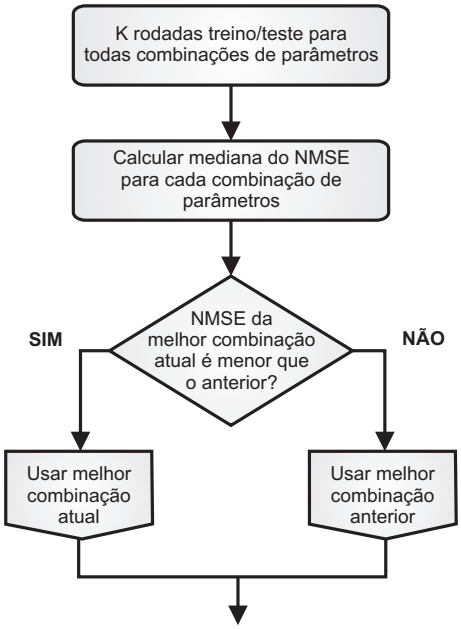


Figura 22 – Bloco funcional de otimização dos parâmetros.

versus atraso de imersão e número de neurônios da primeira camada oculta versus segunda camada oculta, no caso da rede neural com duas camadas ocultas. Embora todos parâmetros possuam correlação entre si, existem correlações mais fortes entre determinados parâmetros. Desta forma, este agrupamento de parâmetros mais correlacionados melhora a tarefa de otimização, pois permite encontrar, de forma direta, a melhor combinação de certos parâmetros. É claro que o ideal seria fazer uma combinação de todos os parâmetros de uma vez, para garantir se encontrar a melhor combinação. Por outro lado, por questões de custos computacionais, seria impraticável se fazer esta busca.

4.5 Exemplo de Uso da Heurística Proposta

Nesta seção, o objetivo é mostrar como a metodologia descrita na seção anterior deve ser utilizada. Para este experimento, é utilizada a série de precipitação mensal de chuva na cidade de Fortaleza, descrita na Seção 4.2.2, para a tarefa de predição recursiva. A rede neural utilizada para verificar o desempenho da predição é a rede FTDNN. Para o treinamento da rede, a série temporal é redimensionada para a faixa [-1,1]. A série redimensionada é dividida em dois conjuntos para a realização da validação holdout, de modo que as primeiras 396 amostras são usadas para o treinamento e as 12 amostras restantes para o teste (predição um ano a frente).

A rede FTDNN tem duas camadas ocultas e um neurônio de saída. Todos os neurônios utilizam função de ativação tangente hiperbólica. O algoritmo *backpropagation* padrão é utilizado para treinar a rede utilizando a predição um-passo-adiante. Cada rodada de

treino/teste da rede FTDNN é repetida 100 vezes (K=100).

A dimensão de imersão (d_E) é estimada pelo método de Cao (Seção 2.7.1) e a curva gerada pelo método é mostrado na Figura 16(a). Os valores a serem escolhidos são os máximos em torno do "joelho" da curva (ou seja $d_E = \{6, 9, 11\}$). O atraso de imersão é estimado como $\tau = 4$, obtido pelo método da informação mútua (Seção 2.7.2). Este método indica que o primeiro mínimo na curva de informação mútua (ver Figura 16(b)) é adotado como uma boa estimativa para τ.

Para otimização do modelo de predição, foram escolhidas faixas de valores dos parâmetros a serem investigados. A seguir é descrita a faixa de parâmetros para execução de cada bloco da Figura 21.

Execução do Bloco 2: para a taxa de aprendizado, escolhemos valores no intervalo de $[0,001...0,2]$ e para o número de épocas de aprendizado, na faixa de $[10...400]$.

Execução do Bloco 3: uma vez que o método de Cao indicou três possíveis valores para d_E, decidimos testar todas as combinações dentro do intervalo $d_E \in [2..18]$ e $\tau \in [2..9]$ e escolher o par (d_E, τ) que retorna o menor valor do NMSE.

Execução do Bloco 4: por fim, o número de neurônios em cada camada oculta é otimizado dentro da faixa de $[2...20]$ neurônios.

A Tabela 2 possui os melhores parâmetros encontrados em cada ciclo de busca do teste com a série de chuva. Cada linha desta tabela representa um ciclo completo do fluxograma da Figura 21. A decisão de mudança ou não dos parâmetros é dada pelo método visto no fluxograma da Figura 22. Neste processo, foram necessários 8 ciclos até que não houvesse mais mudanças nos parâmetros, segundo a heurística adotada como critério de parada. Este procedimento, visto no bloco 6 da Figura 21, se comportou de forma adequada nos testes efetuados, identificando quando o processo de busca, pelos possíveis melhores parâmetros, estabilizou.

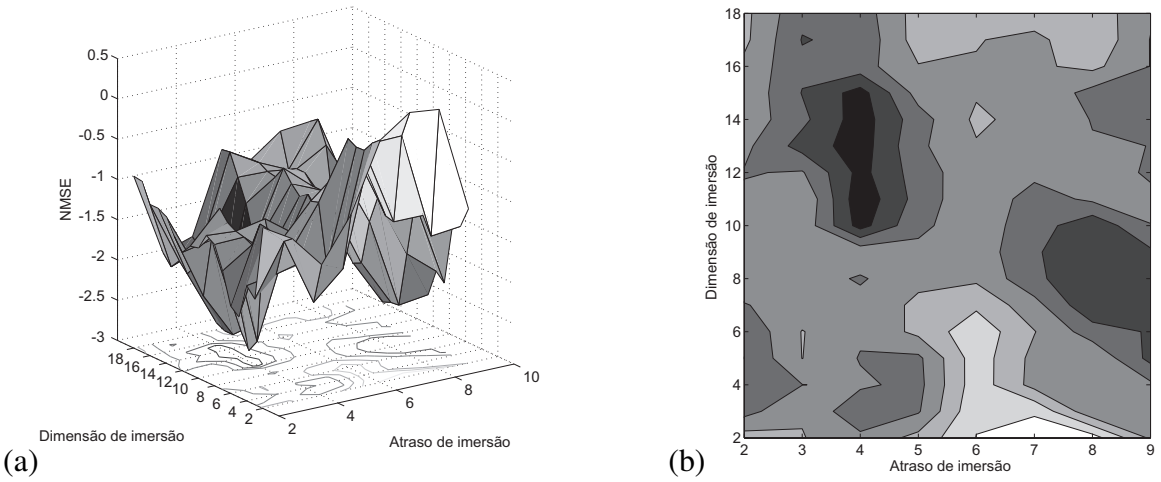
Após vários ciclos da heurística efetuados e satisfeita uma condição de parada, a metodologia retorna os melhores parâmetros dentro dos intervalos que foram testados. Com os resultados do último ciclo de busca, é possível construir um mapeamento do NMSE em função de cada combinação de parâmetros, dentro dos intervalos testados. No problema em questão, são organizadas funções de (d_E, τ), (taxa de aprendizagem, número de épocas) e (número de neurônios na 1ª camada, número de neurônios na 2ª camada).

Os resultados da predição de 12 passos-adiante da rede FTDNN, com duas camadas

Tabela 2 – Ciclos de busca por parâmetros ótimos para a rede FTDNN com 2 camadas ocultas.

Ciclo	Número de épocas	Taxa de aprendizagem	Atraso de imersão	Dimensão de imersão	Nº neurônios 1ª camada oculta	Nº neurônios 2ª camada oculta
1	80	0,05	11	4	4	10
2	130	0,05	11	4	8	8
3	90	0,05	11	4	8	8
4	90	0,05	11	4	8	6
5	90	0,05	11	4	4	6
6	80	0,05	11	4	4	6
7	80	0,05	11	4	4	6
8	80	0,05	11	4	4	6

de neurônios ocultos, são mostrados para o caso da escolha da janela de entrada nas Figuras 23(a) e 23(b). Deve ser lembrando que os resultados dos NMSE nestas figuras estão em escala logarítmica, utilizada para facilitar a vizualização. Esta escala também é utilizada nas outras figuras desta tese. O melhor par encontrado é $(d_E, \tau) = (11, 4)$, confirmando assim os valores sugeridos na Figura 16 para a dimensão e atraso de imersão, estimados pelo método de Cao e informação mútua, respectivamente.

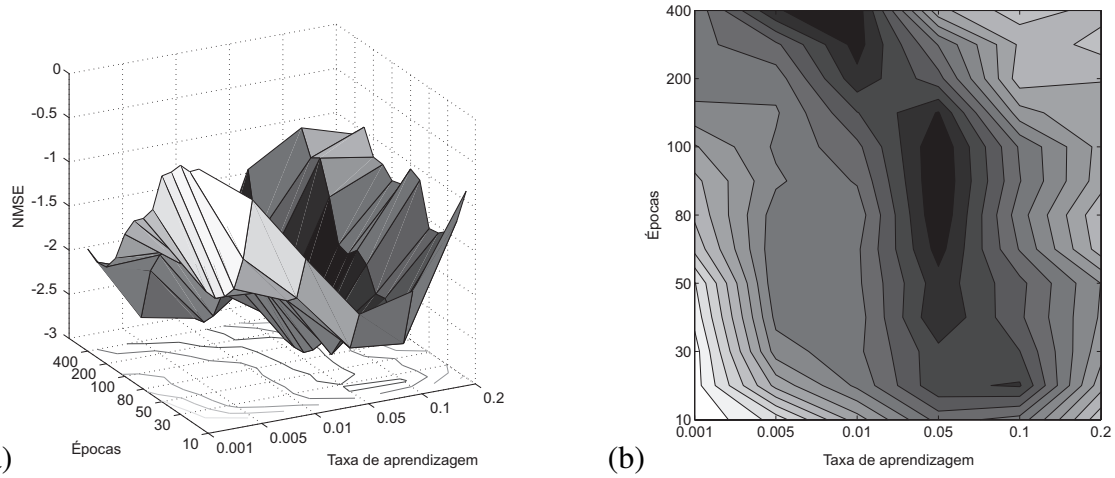


(a) (b)

Figura 23 – Gráfico do NMSE em função da dimensão imersão e do atraso de imersão.

Esta mesma metodologia também é efetuada para encontrar o melhor valor da taxa de aprendizagem e número de épocas de treinamento dentro de uma faixa de análise. Os resultados estão mostrados nas Figuras 24(a) e 24(b). Pode-se verificar um padrão de ocorrência dos mínimos do NMSE, áreas mais escuras, que ocorrem com o aumento da taxa de aprendizagem e com a diminuição do número de épocas. A rede FTDNN com duas camadas obteve o melhor resultado, isto é, aquele que alcança o menor NMSE possui 80 épocas e taxa de aprendizagem

de 0,05.



(a) (b)

Figura 24 – Gráfico do NMSE em função do número de épocas de treinamento e da taxa de aprendizagem.

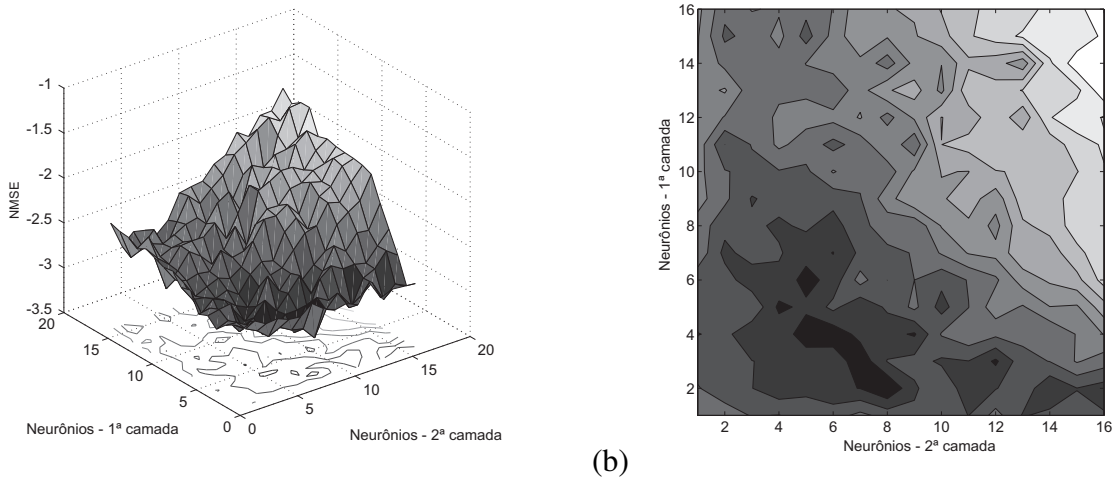
Para o caso da escolha do número de neurônios do modelo FTDNN com duas camadas ocultas são analisados neurônios na faixa de [2, 20], com os resultados gráficos vistos nas Figuras 25(a) e 25(b). Para o modelo FTDNN o erro mínimo é encontrado com 4 e 6 neurônios, na primeira e segunda camada oculta respectivamente. Também é verificada uma região de parâmetros ótimos, região mais escura da Figura 25(b).

É interessante observar na Figura 25(b) que as curvas de nível mantêm a relação proporcional do número de parâmetros ajustáveis. Isto é, com o aumento do número de neurônios na 2ª camada ocultas, o número de neurônios definido na 1ª camada oculta tende a ser menor, com o inverso também acontecendo. Outro fato que deve ser destacado é que nenhuma das heurísticas citadas na Seção B.4 sugeriram valores condizentes com os que são determinados aqui.

4.6 Conclusão

Este capítulo apresentou as metodologias empregadas neste trabalho. Inicialmente foram apresentados os índices de desempenho NMSE e *U de Theil*. Em seguida, as séries temporais utilizadas nesta tese foram descritas. Por fim, foram apresentados os procedimentos para determinar a melhor configuração de uma rede neural, através da proposição de uma heurística para determinar os parâmetros que minimizam o NMSE de um determinado modelo.

Utilizando a série temporal de precipitação de chuvas, a heurística confirma, no caso da escolha dos parâmetros da janela de entrada, os valores sugeridos para a dimensão e



(a) (b)

Figura 25 – Gráfico do NMSE em função do número de neurônios na primeira e da segunda camada oculta.

atraso de imersão, estimados pelo método de Cao e informação mútua, respectivamente. Para o número de neurônios em cada camada oculta, observa-se que nenhuma das heurísticas citadas no Apêndice B (regra do valor médio, raiz quadrada e *Kolmogorov*) sugeriram valores condizentes com os que são determinados aqui. Isto reforça assim a necessidade de melhores heurísticas para a seleção dos parâmetros de um modelo de RNAs para predição de séries temporais.

No próximo capítulo, é apresentada a rede neural baseada no modelo NARX. São descritas as redes NARX-MISO e NARX-MIMO, que são arquiteturas derivadas da rede MLP a partir da introdução de mecanismos de memória de curta duração.

5 REDES NEURAIS NARX-MISO E NARX-MIMO

5.1 Introdução

Este capítulo tem por objetivo dar início à apresentação das arquiteturas de redes neurais propostas nesta tese. Inicialmente, será descrita a rede neural NARX para predição recursiva de um valor da série temporal por vez. Esta arquitetura receberá o nome de rede NARX-MISO, em que o termo MISO refere-se as fato de a rede receber várias entradas e prever apenas uma saída por instante de tempo. Em seguida, será descrita uma extensão da rede neural NARX para predição recursiva de vários valores futuros da série a cada instante de tempo. Esta arquitetura receberá o nome de NARX-MIMO, sendo que o termo MIMO refere-se ao fato de a rede receber várias entradas e prever várias saídas por instante de tempo.

5.2 Rede NARX-MISO

Uma importante e útil classe de sistemas não-lineares de tempo discreto é matematicamente representada pelo modelo NARX (*Nonlinear AutoRegressive model with eXogenous inputs*) (LEONTARITIS; BILLINGS, 1985; LJUNG, 1999; NORGAARD *et al.*, 2000)

$$\begin{aligned} y(n+1) &= f[y(n) \ \cdots \ y(n-d_y+1); \\ &\qquad u(n-k) \ \ u(n-k+1) \ \ \cdots \ \ u(n-d_u-k+1)], \end{aligned} \tag{5.1}$$

em que $u(n) \in \mathbb{R}$ e $y(n) \in \mathbb{R}$ representam, respectivamente, a entrada e a saída do modelo no instante n, enquanto $d_u \geq 1$ e $d_y \geq 1$, $d_u \leq d_y$, são as ordens de memória de entrada e memória de saída, respectivamente. O parâmetro k ($k \geq 0$) é o termo de atraso, conhecido como processo de tempo morto.

Nesta tese, assume-se $k = 0$, obtendo-se assim o seguinte modelo NARX:

$$\begin{aligned} y(n+1) &= f[y(n) \ \cdots \ y(n-d_y+1); \\ &\qquad u(n) \ \ u(n-1) \ \ \cdots \ \ u(n-d_u+1)], \end{aligned} \tag{5.2}$$

que pode ser escrito na seguinte forma de um sistema MISO

$$y(n+1) = f[\mathbf{y}(n); \mathbf{u}(n)], \tag{5.3}$$

onde os $\mathbf{y}(n) \in \mathbb{R}^{d_y}$ e $\mathbf{u}(n) \in \mathbb{R}^{d_u}$ representam os vetores de regressão de saída e entrada, respectivamente.

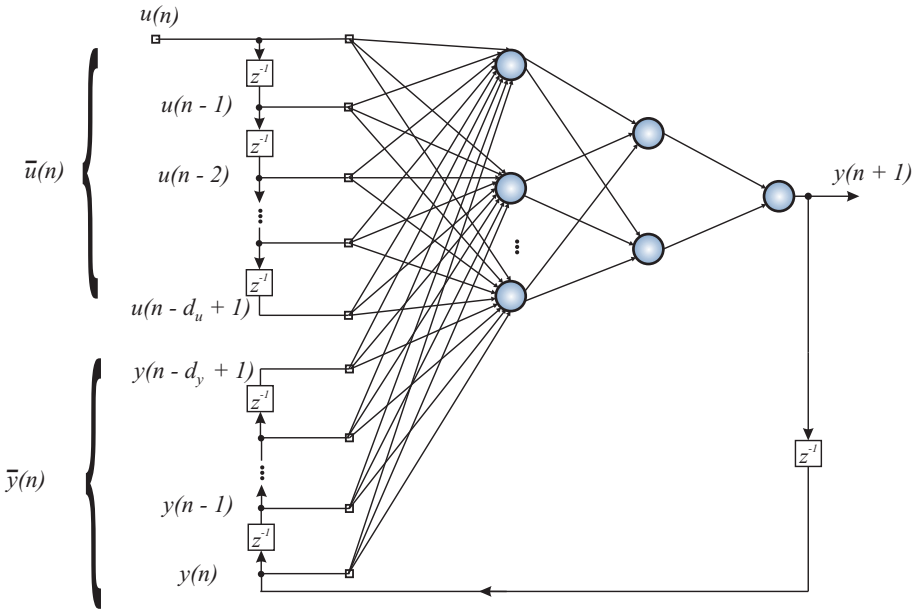


Figura 26 – Rede NARX-MISO com d_u entradas e d_y atrasos da saída e um única saída.

A função não-linear $f(\cdot)$ é geralmente desconhecida e pode ser aproximada, por exemplo, por uma rede Perceptron Multicamadas (MLP) convencional. A arquitetura resultante é chamada de *rede recorrente NARX* (CHEN; BILLINGS; GRANT, 1990; NARENDRA; PARTHASARATHY, 1990), constituindo uma importante classe de arquiteturas neurais dinâmicas computacionalmente equivalentes à máquina de Turing (SIEGELMANN; HORNE; GILES, 1997). A Figura 26 mostra uma rede NARX-MISO com duas camadas ocultas.

No que diz respeito ao treinamento da rede NARX-MISO, ele pode ser realizado de dois modos:

• **Modo de Identificação Paralelo (P)** - Neste caso, também chamado de modo recorrente, a saída estimada é realimentada e incluída na saída do regressor, ou seja:

$$\begin{aligned} \widehat{y}(n+1) &= \widehat{f}[\mathbf{y}_p(n);\mathbf{u}(n)], \\ &= \widehat{f}[\widehat{y}(n)\ \cdots\ \widehat{y}(n-d_y+1);u(n)\ u(n-1)\ \cdots\ u(n-d_u+1)]. \end{aligned} \tag{5.4}$$

• **Modo de Identificação Série-Paralelo (SP)** - Neste caso, também chamado de modo não-recorrente, a saída do regressor é formada somente por valores atuais da saída do sistema, ou seja:

$$\begin{aligned} \widehat{y}(n+1) &= \widehat{f}[\mathbf{y}_{sp}(n);\mathbf{u}(n)], \\ &= \widehat{f}[y(n)\ \cdots\ y(n-d_y+1);u(n)\ u(n-1)\ \cdots\ u(n-d_u+1)]. \end{aligned} \tag{5.5}$$

É interessante notar que o caminho de realimentação mostrado na Figura 26 é apresentado somente no Modo de Identificação Paralelo. Como uma ferramenta para identificação

de sistemas não-lineares, a rede NARX-MISO vem sendo aplicada com sucesso em uma ampla gama de problemas de modelagem entrada-saída, tal como trocadores de calor, estações de tratamento da água, sistemas de transformação catalítica em uma refinaria de petróleo e em predição de séries temporais não-lineares (ver Lin *et al.* (1997) e referências citadas).

Como mencionado na introdução, um tema importante deste trabalho é a questão da predição não-linear de séries temporais com a rede NARX-MISO. Neste tipo de aplicação, a ordem da memória da saída é geralmente definida como $d_y = 0$, reduzindo assim a rede NARX à arquitetura TDNN (LIN *et al.*, 1997), ou seja,

$$\begin{aligned} y(n+1) &= f[\mathbf{u}(n)], \\ &= f[u(n) \;\; u(n-1) \;\; \cdots \;\; u(n-d_u+1)], \end{aligned} \tag{5.6}$$

onde $\mathbf{u}(n) \in \mathbb{R}^{d_u}$ é o regressor do sinal de entrada. Esta formulação simplificada da rede NARX elimina uma parte considerável das suas capacidades de representação como uma rede dinâmica; isto é, toda a informação dinâmica que poderia ser aprendida das memórias passadas da saída é descartada.

Para muitas aplicações práticas, no entanto, tal como modelagem de tráfego de internet auto-similar (GROSSGLAUSER; BOLOT, 1998), a rede neural deve ser capaz de armazenar informação durante um período longo de tempo na presença de ruído. Como abordado anteriormente, tal classe de problema é difícil de lidar com redes neurais recorrentes que utilizam algoritmos de aprendizagem baseados no método do gradiente-descendente, tal como o algoritmo *backpropagation*.

Em algoritmos de treinamento de redes neurais baseados no gradiente, uma fração do gradiente devido às informações n passos de tempo no passado se aproxima de zero quando n se torna grande. Este efeito é conhecido como problema de *vanishing gradient* e tem sido apontado como a principal causa do fraco desempenho dos modelos de RNAs tradicionais, quando se trata de dependências de longo alcance.

A formulação original da rede NARX-MISO não resolve totalmente o problema de *vanishing gradient*, mas tem sido demonstrado que, muitas vezes, tem desempenho muito melhor que as RNAs dinâmicas padrões em uma certa classe de problemas, alcançam a convergência muito mais rápido e possuem melhor desempenho de generalização (LIN *et al.*, 1996). Como apontado em Lin, Horne e Giles (1998), uma explicação intuitiva para essa melhora no desempenho é que as memórias de saída da rede neural NARX- MISO são representadas como conexões

à frente em redes de desdobramento do tempo, que muitas vezes é encontrado em algoritmos de aprendizagem, tais como o *backpropagation* através do tempo (*backpropagation through time*, BPTT). Tais conexões à frente fornecem caminhos mais curtos para a propagação da informação do gradiente, reduzindo a sensibilidade da rede às dependências de longo prazo.

5.2.1 Predição de Séries Temporais com a Rede NARX-MISO

A rede NARX-MISO é usada em problemas de identificação de sistemas, em que estão disponíveis uma série univariada de entrada e uma outra série univariada de saída. Em Menezes-Júnior e Barreto (2008a) uma proposta para utilização da rede NARX-MISO na predição recursiva de séries temporais univariadas é introduzida. Esta formulação é descrita a seguir em mais detalhes.

De acordo com o teorema de Takens (ver Equação (2.17)), uma coleção de valores atrasados no tempo e espaçados num vetor de dimensão d_E deve fornecer informações suficientes para reconstruir os estados do sistema dinâmico observado. Ao fazer isto, está se tentando desdobrar a projeção de volta a um espaço de estado multivariado cujas propriedades topológicas são equivalentes às do espaço de estados que efetivamente gerou a série temporal observada, desde que a dimensão de imersão d_E seja grande o suficiente.

Se for usado $\mathbf{u}(n) = \mathbf{x}_1(n)$ e $y(n+1) = x(n+1)$ na Equação (5.6), então isso leva a uma intuitiva interpretação da reconstrução do espaço de estados não-linear, procedimento equivalente ao problema de predição de séries temporais cujo objetivo é calcular uma estimativa de $x(n+1)$. Assim, a única coisa que se tem que fazer é treinar a rede FTDNN (PRINCIPE; EULIANO; LEFEBVRE, 2000). Uma vez o treinamento tenha sido completado, a rede FTDNN pode ser utilizada para prever o próximo valor da série temporal.

Apesar da possibilidade concreta de se usar a rede FTDNN em predição de séries temporais, é importante lembrar que esta rede pode ser entendida como uma versão simplificada da rede NARX-MISO, obtida pela eliminação da memória da saída (ou seja, $d_y = 0$). Para usar todo o poder computacional da rede NARX como uma arquitetura dinâmica para predição de séries temporais com redundância via recorrência, uma redefinição da entrada e da saída da rede NARX-MISO foi proposta em Menezes-Júnior e Barreto (2008a).

Em primeiro lugar, o regressor do sinal de entrada, denotado por $\mathbf{u}(n)$, é definido pela incorporação das coordenadas de atraso da Equação (2.17):

$$\mathbf{u}(n) = \mathbf{x}_1(n) = [x(n) \;\; x(n-\tau) \;\; \cdots \;\; x(n-(d_E-1)\tau)], \tag{5.7}$$

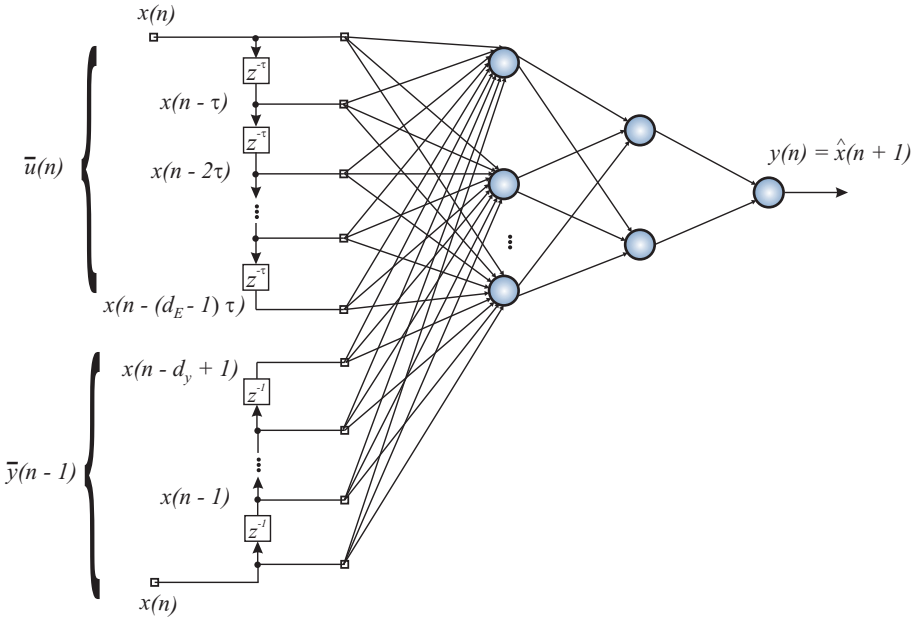


Figura 27 – Arquitetura da rede NARX-MISO durante o treinamento com modo série-paralelo.

onde é substituído $d_u = d_E$. Ou seja, o sinal do regressor de entrada $\mathbf{u}(n)$ é composto de d_E valores observados da série temporal, amostrados a cada τ unidades de tempo.

Em segundo lugar, já que a rede NARX-MISO pode ser treinada em dois modos diferentes, o sinal do regressor de saída $\mathbf{y}(n)$ pode ser escrito de acordo com:

$$\mathbf{y}_{sp}(n) = [x(n) \quad \cdots \quad x(n-d_y+1)], \tag{5.8}$$

ou

$$\mathbf{y}_{p}(n) = [\widehat{x}(n) \quad \cdots \quad \widehat{x}(n-d_y+1)]. \tag{5.9}$$

Nota-se que o regressor de saída para o modo-SP mostrado na Equação (5.8) contém d_y valores passados da série temporal real, enquanto que o regressor de saída para o modo-P, mostrado na Equação (5.9), contém d_y valores passados das estimativas da série temporal. Para uma rede adequadamente treinada, não importa em que modo é feito o treino, na fase de teste, estas saídas são estimativas de valores anteriores de $x(n+1)$.

De agora em diante, as redes NARX-MISO treinadas utilizando os pares de regressão $\{\mathbf{y}_{sp}(n), \mathbf{x}_1(n)\}$ e $\{\mathbf{y}_{p}(n), \mathbf{x}_1(n)\}$ são denotadas por redes NARX-MISO-SP e NARX-MISO-P, respectivamente. Estas redes implementam os seguintes mapeamentos de predição, que podem ser visualizados nas Figuras 27 e 28, respectivamente:

$$\widehat{x}(n+1) = \widehat{f}[\mathbf{y}_{sp}(n), \mathbf{u}(n)] = \widehat{f}[\mathbf{y}_{sp}(n), \mathbf{x}_1(n)], \tag{5.10}$$

$$\widehat{x}(n+1) = \widehat{f}[\mathbf{y}_{p}(n), \mathbf{u}(n)] = \widehat{f}[\mathbf{y}_{p}(n), \mathbf{x}_1(n)], \tag{5.11}$$

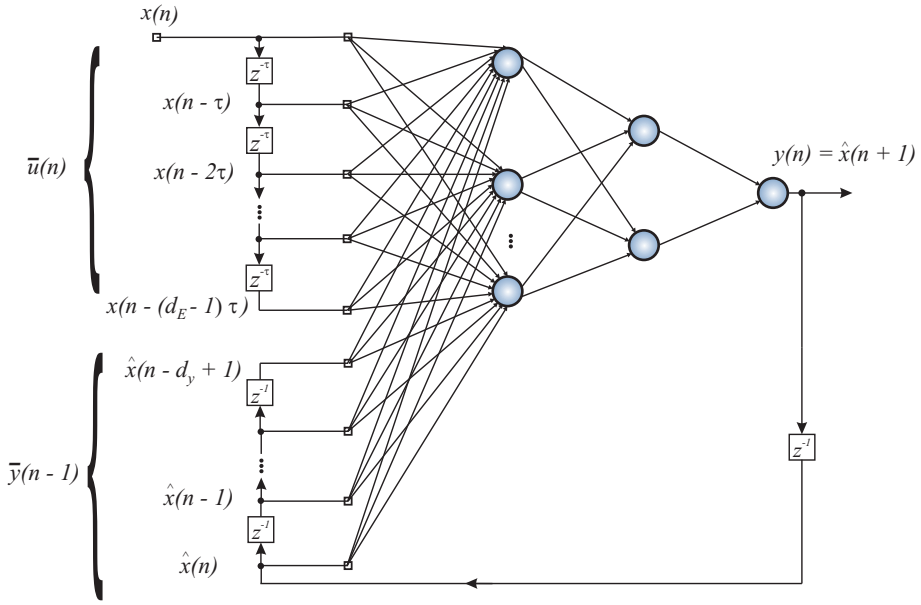


Figura 28 – Arquitetura da rede NARX-MISO durante o treinamento com modo paralelo.

onde a função não-linear $\widehat{f}(\cdot)$ pode ser implementada através de uma MLP treinada com o algoritmo *backpropagation* simples.

É interessante ressaltar que as Figuras 27 e 28 correspondem às diferentes formas que a rede NARX-MISO é treinada, ou seja, no modo SP, ou modo P, respectivamente. Durante a fase de testes, no entanto, uma vez que as predições de longo prazo sejam necessárias, os valores previstos devem ser enviados ao regressor de entrada $\mathbf{u}(n)$ e ao regressor de saída $\mathbf{y}_{sp}(n)$ (ou $\mathbf{y}_p(n)$), simultaneamente. Assim, o modelo preditivo resultante tem dois laços de realimentação, um para o regressor de entrada e outro para o regressor de saída, como ilustrado na Figura 29.

Em relação à redundância, pode-se afirmar que ela está presente em ambos os modos da rede NARX-MISO. No modo série-paralelo, a redundância está presente desde o inicio do treinamento. Já para o modo paralelo, a redundância é via recorrência da saída e está mais presente à medida que o treinamento da rede NARX-MISO avança.

Ao contrário da abordagem baseada na rede FTDNN, referente ao problema de predição de séries temporais não-lineares, a abordagem proposta faz pleno uso da realimentação de saída. As Equações (5.7) e (5.8) são válidas apenas para tarefas de predição um-passo-adiante. Novamente, se o interesse é em tarefas de predição múltiplos-passos-adiante ou recursiva, as estimativas $\widehat{x}$ também devem ser inseridas em ambos os regressores de entrada de forma recursiva.

Pode-se argumentar que, além dos parâmetros d_E e τ, a abordagem proposta introduz mais um a ser determinado, ou seja, d_y. No entanto, este parâmetro pode ser eliminado se for lembrado que, como apontado em Haykin e Principe (1998), o atraso de imersão da Equação (2.17)

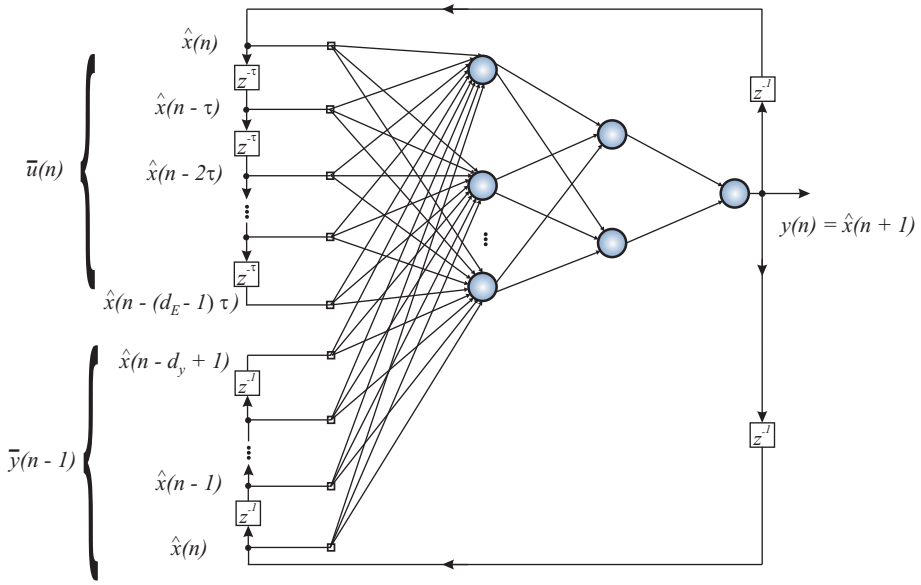


Figura 29 – Arquitetura comum para as redes NARX-MISO-P e NARX-MISO-SP durante a fase de teste (predição recursiva).

tem um forma alternativa dada por:

$$\mathbf{x}_2(n) \triangleq [x(n) \;\; x(n-1) \;\; \cdots \;\; x(n-m+1)] \tag{5.12}$$

onde m é um número inteiro definido como $m \geq \tau \cdot d_E$. Comparando as Equações (5.8) e (5.12), é verificado que uma escolha adequada é dada por $d_y \geq \tau \cdot d_E$, que também satisfaz a condição necessária $d_y > d_u$. No entanto, tem sido encontrado, pela experimentação, que o valor escolhido a partir do intervalo $d_E < d_y \leq \tau \cdot d_E$ é suficiente para alcançar um melhor desempenho de predição com relação àqueles obtidos pelos preditores de séries temporais baseados nas redes neurais convencionais, tais como as arquiteturas TDNN e de Elman.

Finalmente, a abordagem proposta é resumida como segue. A rede NARX-MISO é definida de modo que seu regressor de entrada $\mathbf{u}(n)$ contém amostras da variável medida $x(n)$ separados por $\tau > 0$ intervalos de tempo uns com os outros, enquanto que o regressor de saída $\mathbf{y}(n)$ contém valores reais ou estimados da mesma variável, mas amostrados em intervalos de tempo consecutivos. Com o passar do processo de treinamento, estas estimativas devem se tornar mais semelhantes aos valores reais da série temporal, indicando a convergência do processo de treinamento. Assim, é interessante notar que o regressor de entrada supre informações de médio a longo prazo sobre o comportamento dinâmico da série temporal, uma vez que o atraso τ é geralmente maior do que a unidade, enquanto o regressor de saída, tendo a rede já convergido, supre informações de curto prazo acerca da série temporal.

5.3 Rede NARX-MIMO

A arquitetura NARX-MISO descrita até aqui, recebe várias entradas e prevê apenas uma saída por instante de tempo. Nesta seção é introduzida uma extensão da rede neural NARX para predição recursiva de vários valores futuros da série a cada instante de tempo. Esta arquitetura receberá o nome de NARX-MIMO, sendo que o termo MIMO refere-se ao fato de a rede receber várias entradas e prever várias saídas por instante de tempo. Esta proposta é aplicada na predição múltiplos-passos-adiante de séries temporais univariadas. Esta formulação é descrita a seguir em mais detalhes.

A rede NARX-MIMO utiliza a mesma formulação de entrada da rede NARX-MISO, assim como foi proposto em Menezes-Júnior e Barreto (2008a). Em primeiro lugar, o regressor do sinal de entrada, denotado por $\mathbf{u}(n)$, é definido pela incorporação das coordenadas de atraso da Equação (2.17):

$$\mathbf{u}(n) = \mathbf{x}_1(n) = [x(n) \;\; x(n-\tau) \;\; \cdots \;\; x(n-(d_E-1)\tau)], \tag{5.13}$$

onde é substituído $d_u = d_E$. Ou seja, o sinal do regressor de entrada $\mathbf{u}(n)$ é composto de d_E valores observados da série temporal, amostrados a cada τ unidades de tempo.

Em segundo lugar, no treinamento da rede NARX-MIMO, o sinal do regressor de saída $\mathbf{y}(n)$ é definido no modo-SP, pela seguinte equação

$$\mathbf{y}_{sp}(n) = [x(n) \;\; \cdots \;\; x(n-d_y+1)]. \tag{5.14}$$

Nota-se que o regressor de saída para o modo-SP contém d_y valores passados da série temporal observada, não necessitando de estimativas da série, como acontece no regressor de saída para o modo-P mostrado na Equação (5.9). Desta forma, a rede NARX-MIMO treinada utilizando o par de regressão $\{\mathbf{y}_{sp}(n), \mathbf{x}_1(n)\}$ pode ser denotada por NARX-MIMO-SP.

A novidade da rede NARX-MIMO está na forma como é construído o mapeamento de predição. Nesta rede, diversas saídas ou nós na camada de saída são implementados, um para cada horizonte de predição desejado. Esta ideia é baseada no preditor MIMO, introduzida na Seção 2.8.4 e proposto por Bontempi (2008).

O método de predição MIMO consiste em construir um mapeamento $f : \mathbb{R}^{d_E+d_y} \rightarrow \mathbb{R}^H$ pela seguinte equação

$$[\widehat{x}(n+H) \;\; \cdots \;\; \widehat{x}(n+h) \;\; \cdots \;\; \widehat{x}(n+1)] = \widehat{f}[\mathbf{y}_{sp}(n), \mathbf{u}(n)] = \widehat{f}[\mathbf{y}_{sp}(n), \mathbf{x}_1(n)], \tag{5.15}$$

onde a função não-linear $\widehat{f}(\cdot)$ pode ser implementada através de uma rede MLP treinada com o algoritmo *backpropagation* simples e H é o horizonte de predição múltiplos-passos-adiante.

Em suma, vale notar que a rede NARX-MIMO, conforme formulada para tarefa de predição múltiplos-passos-adiante, retorna não somente um escalar (como na rede NARX-MISO), mas sim um vetor de predição. Daí, a razão do termo MIMO no nome do modelo. Desta forma, dependendo do valor de H, isto é, o tamanho do horizonte de predição, a rede pode possuir muitos neurônios na camada de saída, dificultando o processo de aprendizagem e, consequentemente, piorando a precisão das predições.

Uma solução para o problema é assumir uma abordagem intermediária, em que é utilizado um parâmetro inteiro (s) para dimensão da saída menor que o horizonte de predição (H). Assim a rede NARX-MIMO passa a ser formulada pela seguinte equação

$$[\widehat{x}(n+s) \ \cdots \ \widehat{x}(n+h) \ \cdots \ \widehat{x}(n+1)] = \widehat{f}[\mathbf{y}_{sp}(n), \mathbf{u}(n)] = \widehat{f}[\mathbf{y}_{sp}(n), \mathbf{x}_1(n)], \tag{5.16}$$

em que $1 < s < H$. Vale observar que se $s = 1$ a rede NARX-MIMO se reduz a uma arquitetura com uma única saída, tornado-se uma rede NARX-MISO.

É interessante ressaltar que a rede NARX-MIMO-SP, como vista na Figura 30, corresponde à forma como a rede NARX-MISO é treinada. No entanto, durante a fase de teste, em tarefas de predição múltiplos-passos-adiante ou recursiva, os valores previstos devem ser enviados ao regressor de entrada $\mathbf{u}(n)$ e ao regressor de saída $\mathbf{y}_{sp}(n)$, simultaneamente. Assim, o modelo preditivo resultante tem dois laços de realimentação, um para o regressor de entrada e outro para o regressor de saída.

O passo seguinte é formular como as saídas estimadas serão realimentadas para o regressor de entrada e para o regressor de saída, já que a rede NARX-MIMO não somente possui um escalar de realimentação e sim um vetor de saídas estimadas. Desta forma, é proposto aqui um método que realimenta a estimativa de $\widehat{x}(n+h)$ diminuindo os erros acumulados na predição. A formulação é mostrada na Tabela 3.

Tabela 3 – Realimentação da estimativa da rede NARX-MIMO.

Instante de tempo	Saída da rede com valores estimados da série	Valor a ser realimentado para os regressores
n	$\widehat{x}_0(n+s), \ldots, \widehat{x}_0(n+2), \widehat{x}_0(n+1)$	$\widehat{x}_0(n+1)$
$n+1$	$\widehat{x}_1(n+s+1), \ldots, \widehat{x}_1(n+3), \widehat{x}_1(n+2)$	média $[\widehat{x}_1(n+2), \widehat{x}_0(n+2)]$
$n+2$	$\widehat{x}_2(n+s+2), \ldots, \widehat{x}_2(n+4), \widehat{x}_2(n+3)$	média $[\widehat{x}_2(n+3)\ , \widehat{x}_1(n+3), \widehat{x}_0(n+3)]$
$\vdots$	$\vdots$	$\vdots$

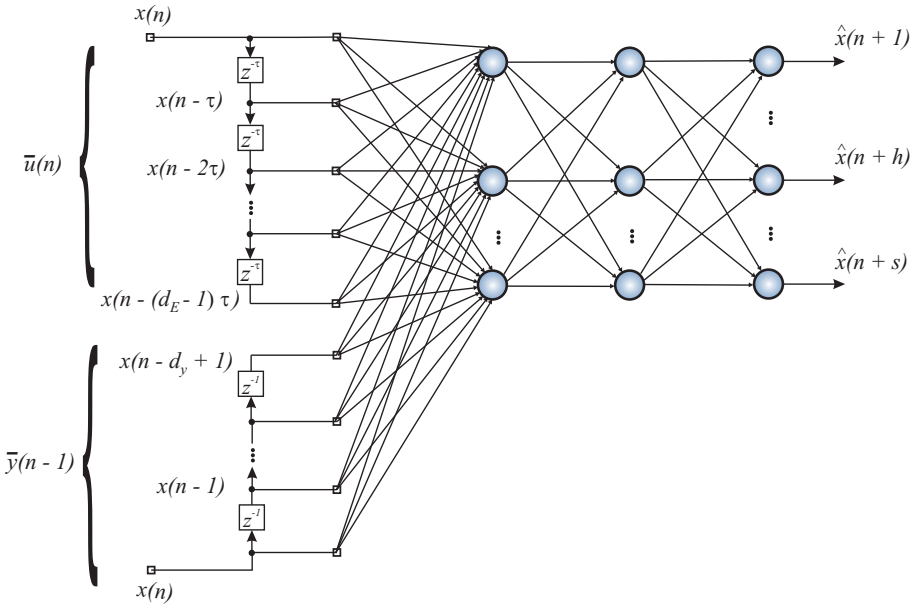


Figura 30 – Arquitetura da rede NARX-MIMO durante o treinamento com modo série-paralelo em que recebe várias entradas e prever várias saídas.

A cada instante de tempo, a estimativa a ser enviada ao regressor de entrada $\mathbf{u}(n)$ e ao regressor de saída $\mathbf{y}_{sp}(n)$ é formada pela média das últimas estimativas já disponíveis pelo preditor MIMO. Percebe-se que, ao passo que o horizonte de predição tende ao infinito, o valor a ser realimentado será composto por um número cada vez maior de estimativas, gerando uma maior confiabilidade dos valores preditos.

Finalmente, a abordagem é resumida como uma forma intermediária entre o preditor recursivo e o preditor MIMO, procurando reduzir os principais problemas de cada método. O primeiro benefício é que este método tem o propósito de remover a suposição de independência condicional entre as predições, que pode ocorrer na abordagem de predição direta, já que é construído um modelo somente para fazer predições múltiplos-passos-adiante. Em segundo lugar, no método proposto o número de saídas é menor que no preditor MIMO tradicional, diminuindo a complexidade da rede reural a ser construída. Por fim, a rede NARX-MIMO reduz o acúmulo dos erros de predição, teoricamente de forma mais eficiente que a rede NARX-MISO, que utiliza o método de predição recursiva.

5.4 Conclusão

Este capítulo apresentou as arquiteturas de redes neurais baseadas do modelo NARX. Estas redes foram classificadas de acordo com o número de saídas. Assim foram descritas as redes NARX-MISO e NARX-MIMO. Estas arquiteturas são derivadas da rede MLP a partir da

introdução de mecanismos de memória de curta-duração.

No próximo capítulo, são discutidos as técnicas baseadas em projeções aleatórias. Primeiramente é apresentada a rede neural recorrente de Ecos de Estado, em seguida a rede neural de Máquina de Aprendizado Extremo.

6 REDES NEURAIS BASEADAS EM PROJEÇÕES ALEATÓRIAS

6.1 Introdução

Este capítulo apresenta os conceitos principais relacionados com dois desenvolvimentos recentes no campo da neurocomputação: a rede de Ecos de Estado (*Echo State Network*, ESN) (JAEGER, 2001) e a rede Máquina de Aprendizado Extremo (*Extreme Machine Learning*, ELM) (HUANG; ZHU; SIEW, 2006). O ponto comum entre as abordagens é que os pesos das unidades de entrada para a camada oculta permanecem fixos em valores definidos a priori, enquanto os pesos da camada de saída são ajustáveis. A rede ESN tem uma estrutura recorrente, enquanto a ELM é uma rede neural *feedforward*.

Isto posto, os objetivos desta exposição sobre as redes ESN e ELM são múltiplos. Primeiramente, busca-se estudar o desempenho das redes NARX-MISO e NARX-MIMO com o estado da arte em algoritmos para predição recursiva (a rede ESN, no caso). Em segundo lugar, busca-se implementar a rede NARX-MISO usando o algoritmo de aprendizado da rede ELM em vez do algoritmo *backpropagation* simples. Por último, mas não menos importante, busca-se fazer um amplo estudo dos parâmetros de treinamento (projeto) da rede ESN, a fim de avaliar e, quiçá, melhorar desempenho de predição recursiva.

6.2 Projeções Aleatórias

Um novo paradigma de projeto de redes neurais vem sendo proposto sob a alcunha de Projeções Aleatórias (*Random Projections*) (MICHE; SCHRAUWEN; LENDASSE, 2010). O interesse neste paradigma por parte da comunidade científica vem aumentando devido à forma singular de projeto de tais arquiteturas, uma vez que esta não requer o treinamento dos neurônios não-lineares da camada oculta, chamada neste contexto de projeções. Apenas os neurônios da camada de saída têm seus pesos ajustados, o que pode ser realizado por métodos recursivos e não-recursivos. Estas arquiteturas também fazem uso geralmente de uma grande quantidade de neurônios ocultos, a fim de compensar a inicialização aleatória dos pesos dos neurônios ocultos.

Os algoritmos de treinamento de redes neurais tradicionais (recorrentes ou *feed-forward*) são caracterizados pelo excesso de parâmetros ajustáveis, pela baixa velocidade de convergência e por problemas causados pela sensibilidade aos pesos iniciais. Já as RNAs baseadas em projeções aleatórias possuem poucos parâmetros a serem ajustados e uma estrutura

fácil de ser construída (HUANG; WANG; LAN, 2011). Mesmo com um projeto simplificado, as RNAs baseadas em projeções aleatórias normalmente possuem desempenho equivalente ao de outras redes neurais convencionais, tais como as redes MLP, RBF e ELMAN, despertando, por isto, o interesse da comunidade científica.

Dentro do arcabouço geral das redes baseadas em projeções aleatórias, destacam-se as redes ESN e ELM, sendo a primeira uma arquitetura recorrente e a segunda uma arquitetura *feedforward*.

A rede ESN insere-se, enquanto arquitetura recorrente, no contexto da Computação de Reservatório (MICHE; SCHRAUWEN; LENDASSE, 2010). Este paradigma de aprendizado dinâmico baseia-se em uma estrutura interna - conhecida como *reservatório* - com conectividade esparsa e aleatória, a fim de extrair informações relevantes da dinâmica do sistema a ser modelado. Os neurônios de tais estruturas são os mesmos comumente utilizados em outras estruturas de redes neurais (unidades lineares e sigmóides). Algumas outras redes que estão sob o paradigma de Computação de Reservatório, são: a rede LSTM (*Long Short-Term Memory*) (GERS, 2001) e a rede LSM (*Liquid State Machines*) (NATSCHLAGER; MAAS; MARKRAM, 2002).

A rede ELM, por outro lado, utiliza uma simples camada de uma rede neural *feedforward*, sem realimentação. Os pesos da camada oculta são iniciados aleatoriamente e não precisam ser atualizados, ficando a cargo apenas do cálculo dos pesos da camada de saída. Desta forma, a principal diferença entre as redes ELM e ESN encontra-se na recorrência da rede neural. A rede ESN faz uso de uma "piscina" de neurônios aleatoriamente interligados uns com os outros, o que pode ser descrito como uma rede neural recorrente (MICHE; SCHRAUWEN; LENDASSE, 2010).

Mais detalhes sobre o funcionamento das redes ESN e ELM são fornecidos a seguir.

6.3 Rede de Ecos de Estado

A rede ESN é uma rede neural recorrente formada, normalmente, por neurônios com função de ativação sigmoidal que são conectadas aleatoriamente, de forma que as únicas partes realmente treinadas são as conexões de saída da rede (JAEGER, 2001). A rede em si, é usada apenas como um reservatório de dinâmicas e as conexões de saída são treinadas usando regressão linear. A Figura 31 ilustra a ideia básica da rede ESN, onde as linhas pontilhadas representam conexões que não são necessariamente utilizadas.

A popularidade da rede ESN vem crescendo entre os vários tipos de redes recorrentes

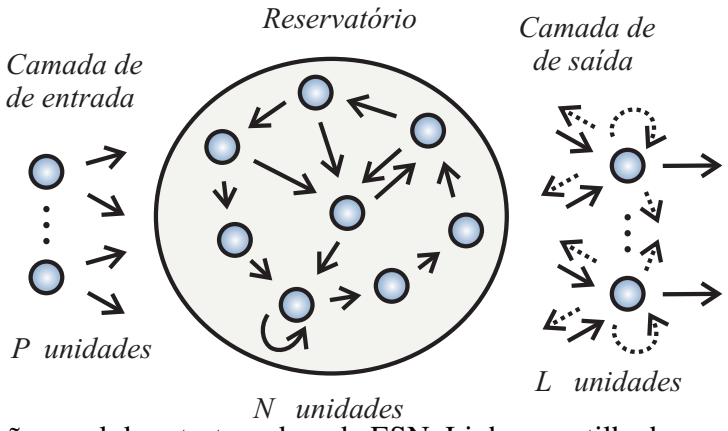


Figura 31 – Ilustração geral da estrutura da rede ESN. Linhas pontilhadas representam conexões que não são necessariamente utilizadas.

devido a forma não-usual de se construir a sua arquitetura, no que se refere à diminuição da quantidade de parâmetros a serem definidos a priori, à quantidade de neurônios que precisam ser treinados e à rapidez com que isto é realizado. Devido a tais características, muitos trabalhos vêm sendo feitos e resultados promissores vêm sendo alcançados, porém este campo ainda está carente de resultados práticos na área de predição de séries temporais.

A arquitetura da rede ESN se destaca das abordagens tradicionais porque consegue eficientemente reter informação temporal em seu reservatório de dinâmicas por um longo período (JAEGER, 2001). Isto faz das redes ESNs, entre outros modelos de sistemas dinâmicos não-lineares, interessantes modelos para séries temporais complexas (e.g. séries de memória longa (HURST, 1951)), com potenciais aplicações em predição recursiva.

(*i*) **Definição da Arquitetura ESN**

A rede ESN é considerada uma rede neural de tempo discreto com P unidades de entrada, N unidades internas e L unidades de saída da rede. O vetor de entrada no instante n é representado por $\mathbf{x}(n) = [x_1(n) \ldots x_P(n)]^T \in \mathbb{R}^p$ e os pesos dessa camada formam uma matriz de dimensões $N \times P$, denotada por

$$\mathbf{W}^{in} = [w_{ij}^{in}]_{N \times P},$$

em que w_{ij}^{in} é o peso da j-ésima unidade de entrada ao i-ésimo neurônio do reservatório.

As ativações dos neurônios do reservatório são denotados por $\mathbf{r}(n) = [r_1(n) \ldots r_N(n)]^T \in \mathbb{R}^N$ e os pesos entre neurônios do reservatório são representados por uma matriz $N \times N$, denotada por

$$\mathbf{W} = [w_{ij}]_{N \times N},$$

em que w_{ij} é o peso que conecta o j-ésimo neurônio do reservatório ao j-ésimo neurônio do reservatório.

Caso haja conexões que se projetam da camada de saída para os neurônios do reservatório, a matriz de pesos correspondente é denotada por

$$\mathbf{W}^{back} = [w_{ij}^{back}]_{N \times L},$$

em que w_{ij}^{back} é o peso que conecta o j-ésimo neurônio de saída ao i-ésimo neurônio do reservatório.

O vetor de ativações dos neurônios do reservatório é atualizado de acordo com a seguinte regra:

$$\mathbf{r}(n+1) = \mathbf{f}(\mathbf{W}^{in}\mathbf{x}(n+1) + \mathbf{W}\mathbf{r}(n) + \mathbf{W}^{back}\mathbf{y}(n)), \tag{6.1}$$

onde $\mathbf{f} = (f_1, ..., f_N)$ são as funções de ativação dos neurônios do reservatório, normalmente funções sigmoidais, tal como a tangente hiperbólica.

As ativações dos neurônios de saída da rede são agrupados em um vetor $\mathbf{y}(n) = [y_1(n) \ ... \ y_L(n)]^T \in \mathbb{R}^L$ sendo calculado da seguinte forma:

$$\mathbf{y}(n+1) = \mathbf{f}^{out}(\mathbf{W}^{out}\mathbf{z}(n+1)), \tag{6.2}$$

em que $\mathbf{f}^{out} = (f_1^{out}, ..., f_N^{out})$ são as funções de ativação das unidades de saída e $\mathbf{z}(n+1) = [\mathbf{x}(n+1) \mid \mathbf{r}(n+1) \mid \mathbf{y}(n)] \in \mathbb{R}^{P+N+L}$ é o vetor resultante da concatenação do vetor de entrada atual ($\mathbf{x}(n+1)$), do vetor de ativações atuais dos neurônios do reservatório ($\mathbf{r}(n+1)$) e do vetor de ativações das unidades de saída no instante anterior ($\mathbf{y}(n)$).

Os pesos da camada correspondente são dispostos numa matriz de dimensões $L \times (P+N+L)$, denotada por

$$\mathbf{W}^{out} = [w_{ij}^{out}]_{L \times (P+N+L)},$$

em que w_{ij}^{out} é o peso que conecta a j-ésima ativação do neurônio de saída ao i-ésimo neurônio de saída.

A arquitetura até aqui descrita pode ser vista de forma detalhada na Figura 32, exibindo a direção de todas as alimentações e realimentações utilizadas. Esta figura também mostra as dimensões de todas as matrizes e vetores da rede ESN formada por P unidades de entrada, N unidades internas e L unidades de saída.

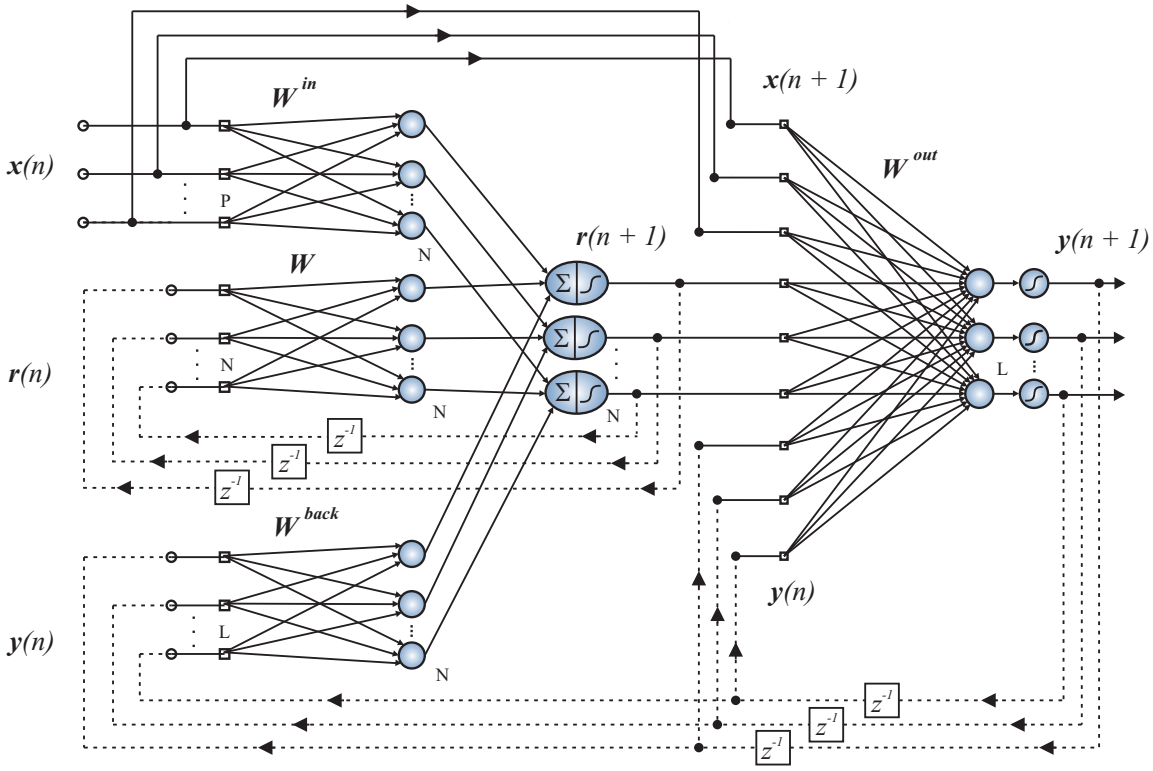


Figura 32 – Arquitetura da rede ESN com todas alimentações e realimentações utilizadas.

(*ii*) **Inicialização dos Pesos**

De acordo com Jaeger (2001), a especificação das matrizes é feita de acordo com as seguintes heurísticas:

1. Gerar aleatoriamente uma matriz esparsa $\mathbf{W}_0$ com conectividade baixa (1% a 10%). Uma matriz esparsa é uma matriz com muitos zeros, ou seja, é uma matriz com poucos elementos não-nulos.
2. Normalizar $\mathbf{W}_0$ para uma matriz $\mathbf{W}_1$, fazendo $\mathbf{W}_1 = (1/|\lambda_{max}|)\mathbf{W}_0$, onde $|\lambda_{max}|$ é o valor absoluto do maior autovalor da matriz $\mathbf{W}_0$.
3. Mudar a escala da matriz $\mathbf{W}_1$ para $\mathbf{W} = \alpha\mathbf{W}_1$, com $\alpha < 1$, segundo o qual $\mathbf{W}$ obtém um raio espectral α.
4. Gerar valores aleatórios uniformemente distribuídos para os pesos das matrizes $\mathbf{W}^{in}$, $\mathbf{W}^{back}$ e $\mathbf{W}^{out}$.

(*iii*) **Período Transitório**

É recomendado executar o treinamento (estimação) da matriz $\mathbf{W}^{out}$ excluindo o período transitório do reservatório, após o qual os efeitos do estado inicial da rede tenham se desvanecido no tempo. Assim, executa-se a rede de $n = 0$ a $n = n_{max}$, onde n_{max} é o tamanho da série de treinamento. Por fim, exclui-se o estado transitório inicial, de $n = 0$ a $n = n_{min}$. Os estados

$\mathbf{r}(n)$, $\mathbf{x}(n)$ e $\mathbf{y}(n)$ de $n = n_{min}+1,...,n_{max}$ são armazenados na matriz $\mathbf{M}_{(n_{max}-n_{min}+1)\times(K+N+L)}$, representada por

$$\mathbf{M} = \begin{pmatrix} \mathbf{x}(n_{min}+1) & | & \mathbf{r}(n_{min}+1) & | & \mathbf{y}(n_{min}+1) \\ \mathbf{x}(n_{min}+2) & | & \mathbf{r}(n_{min}+2) & | & \mathbf{y}(n_{min}+2) \\ \vdots & & \vdots & & \vdots \\ \mathbf{x}(n_{max}) & | & \mathbf{r}(n_{max}) & | & \mathbf{y}(n_{max}) \end{pmatrix}_{(n_{max}-n_{min}+1)\times(K+N+L)} . \quad (6.3)$$

Ao mesmo tempo, a saída desejada $d(n) = y(n+1)$ é armazenada na matriz $\mathbf{D}_{(n_{max}-n_{min}+1)\times L}$. Lembrando que, como o problema de interesse é predição de séries temporais univariadas, usa-se $L = 1$. Logo a matriz $\mathbf{D}$ reduz-se a um vetor, ou seja

$$\mathbf{D} = \begin{pmatrix} d(n_{min}+1) \\ d(n_{min}+2) \\ \vdots \\ d(n_{max}) \end{pmatrix} \in \mathbb{R}^{n_{max}-n_{min}+1}. \quad (6.4)$$

(*iv*) **Cálculo dos Pesos dos Neurônios de Saída**

O último passo de treinamento é a atualização dos pesos ajustáveis da rede, representados pela matriz $\mathbf{W}^{out}$. Desta forma, pode-se utilizar um algoritmo não-recursivo, tal como o método de estimação dos mínimos quadrados, também conhecido como método da pseudoinversa (PRINCIPE; EULIANO; LEFEBVRE, 2000), ou seja:

$$(\mathbf{W}^{out})^T = \left(\mathbf{M}^T\mathbf{M}\right)^{-1}\mathbf{M}^T\mathbf{D}. \quad (6.5)$$

A formulação da Equação (6.5) assume que os neurônios de saída são lineares. Porém, a camada de saída pode possuir neurônios com função de ativação sigmoidal, em geral a tangente hiperbólica. O treinamento continua sendo por meio da Equação (6.5), exceto pelo fato de a matriz $\mathbf{D}$ armazenar agora, em vez da saída desejada, a inversa da função de ativação: $tanh^{-1}(d(n))$.

6.3.1 Extensões da Rede ESN para Aplicação em Predição de Séries Temporais

A descrição anterior da rede ESN foi feita para uma arquitetura geral, com P unidades de entrada, L neurônios de saída. Porém, a arquitetura da rede ESN original, quando aplicada em tarefas de predição de séries temporais univariadas, tem apenas uma unidade de entrada (i.e. $P = 1$) e um único neurônio na camada de saída ($L = 1$).

Mesmo com um número menor de parâmetros quando comparada com redes recorrentes tradicionais, a rede ESN tem algumas particularidades quanto ao uso de realimentações e parâmetros restantes. Jaeger (2001) não dá dicas claras de como projetar a rede ESN, não especificando quais laços de realimentação utilizar e nem justificando a escolha dos valores de certos parâmetros. Desta forma, são descritos a seguir algumas possíveis variações da arquitetura ESN original.

6.3.1.1 Unidade de Entrada

A unidade de entrada é essencial para a rede ESN, pois esta alimenta a rede, recebendo informações dos dados e inserindo conhecimento na rede. Pode-se utilizar a unidade de entrada recebendo apenas o vetor de entrada atual da série temporal em cada instante, passando pelos pesos $\mathbf{W}^{in}$ e alimentando o reservatório, como também por meio de conexões diretas para a camada de saída.

A dimensão de entrada da rede ESN original é unitária (i.e. $P = 1$), mas em aplicações de predição de séries temporais esta dimensão pode ser maior que 1 ($P > 1$). Em tarefas de predição de sinais, a dimensão do regressor de entrada é definida de acordo com o teorema de Takens, ou seja

$$\mathbf{x}(n) = [x(n) \;\; x(n-\tau) \;\; \cdots \;\; x(n-(d_E-1)\tau)]^T, \tag{6.6}$$

em que $\mathbf{x}(n)$ é um vetor que contém $p = d_E$ elementos da série, contados a partir do elemento atual $x(n)$, espaçados um do outro de τ unidades de tempo. Mais adiante nesta tese, a eficácia da utilização da janela de Takens como entrada da rede ESN é avaliada e seu desempenho em tarefas de predição é comparado com a proposta original da rede ESN.

Em Jaeger e Haas (2004), os autores utilizam a rede ESN para predição da série de Mackey-Glass, mas não alimentam a rede com observações atuais dos dados, mantendo como informação para a unidade de entrada apenas um valor constante. Esta configuração também é avaliada, observando o desempenho na tarefa de predição. Deve ser notado que quando isto acontece, o papel de inserir conhecimento na rede fica para a unidade de saída projetada para os neurônios do reservatório.

6.3.1.2 Unidade de Saída Projetada para os Neurônios do Reservatório

A saída da rede é realimentada para a entrada da rede, projetando a saída para os neurônios do reservatório. Os pesos destinados a esta tarefa são os representados pela matriz $\mathbf{W}^{back}$. Jaeger (2001) não deixa claro se é essencial a utilização deste tipo de realimentação. Apesar disso, Jaeger afirma que, durante a fase de treinamento, durante o período transitório das ativações do reservatório, este laço de realimentação é formado pelos próprios valores observados dos dados. Na fase de teste, onde a rede passa a não ser mais alimentada pelos dados observados, esta unidade é verdadeiramente formada pela saída projetada para os neurônios do reservatório.

Como o interesse desta tese reside em tarefas de predição de séries temporais, um único neurônio é utilizado na camada de saída ($L = 1$), e assim também um único valor (escalar) é projetado por vez para as unidades internas. Se for utilizada a mesma ideia de regressor de saída empregado na rede NARX-MISO (Seção 5.2.1), a dimensão da unidade de saída projetada pode ser maior que 1. Desta forma, propõe-se aqui a utilização de um regressor de saída na rede ESN, contendo d_y valores passados da série temporal, ou seja

$$\mathbf{y}(n) = [x(n)\ \ x(n-1)\ \ x(n-2)\ \ \cdots\ \ x(n-d_y+1)]^T \in \mathbb{R}^{d_y}. \tag{6.7}$$

Este regressor, na fase de treino, é formado pelos valores passados da série temporal observada, enquanto que na fase de teste é formado pelos valores passados das estimativas da série temporal.

A arquitetura descrita até aqui pode ser vista de forma detalhada na Figura 33, com as devidas mudanças enunciadas acima. A rede ESN mostrada nesta figura possui o vetor de entrada definido de acordo com o Teorema de Takens, um único neurônio de saída e um regressor de saída contendo d_y valores passados da série temporal, correspondendo à unidade de saída projetada para os neurônios do reservatório.

Diante do exposto, diversas arquiteturas da rede ESN são avaliadas nesta tese, tanto com as variações possíveis da rede ESN original, como também utilizando as variações propostas aqui. Ao todo podem-se construir oito configurações de modelos diferentes com a rede ESN. Estas variantes estão listadas, de forma mais detalhada, na Tabela 4.

6.3.2 Otimização dos Parâmetros da Rede ESN

Devido a ausência de estudos detalhados sobre a especificação de parâmetros da rede ESN, uma das contribuições desta tese é analisar os parâmetros de treinamento da rede ESN. Após intensa experimentação inicial, foram escolhidos alguns parâmetros para serem otimizados,

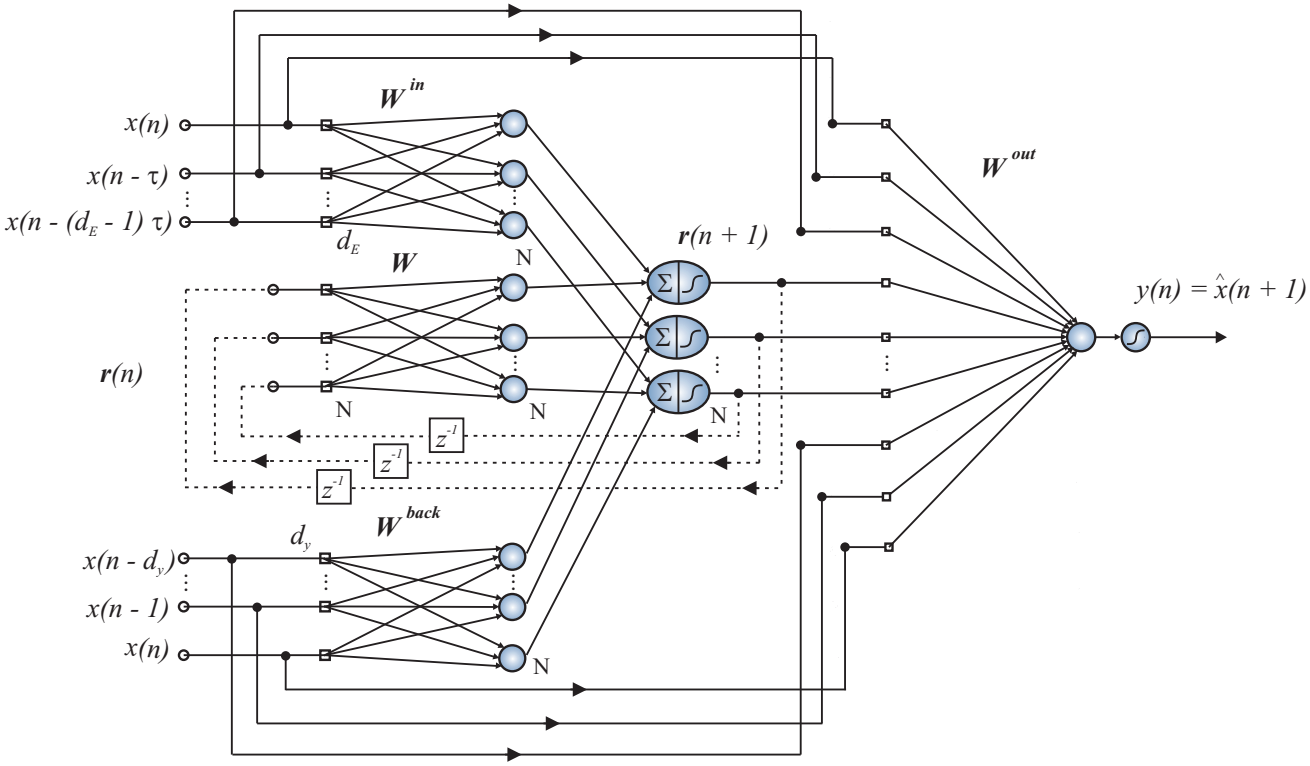


Figura 33 – Rede ESN para tarefas de predição de séries temporais.

Tabela 4 – Configurações dos diferentes modelos testados para a rede ESN.

Configurações das redes ESN(camada de entrada | camada de saída)	Informações da saída e da entrada para neurônios do reservatório	Informação da Unidade de Entrada com Takens (d_E) ou valor fixo	Informação da entrada (d_E) ou valor fixo para camada de saída	Informação da saída com (d_y) atrasadores para camada de saída
ESN(1, d_y | 0, 0)	Sim	Não. Entrada fixa	Não	Não
ESN(d_E, d_y | 0, 0)	Sim	Sim. Takens	Não	Não
ESN(1, d_y | 1, 0)	Sim	Não. Entrada fixa	Sim	Não
ESN(d_E, d_y | d_E, 0)	Sim	Sim. Takens	Sim	Não
ESN(1, d_y | 0, d_y)	Sim	Não. Entrada fixa	Não	Sim
ESN(d_E, d_y | 0, d_y)	Sim	Sim. Takens	Não	Sim
ESN(1, d_y | 1, d_y)	Sim	Não. Entrada fixa	Sim	Sim
ESN(d_E, d_y | d_E, d_y)	Sim	Sim. Takens	Sim	Sim

pois alguns indicavam ter forte influência no desempenho da predição. Para este fim, utiliza-se a metodologia apresentada na Seção 4.4. Os parâmetros da rede ESN a serem analisados são os seguintes:

- **Probabilidade de valores não nulos das unidades do reservatório:** são avaliados valores na faixa de $[0,5\% - 100\%]$. Uma probabilidade extremamente baixa significa que o reservatório é quase todo formado por valores nulos, ou seja, a matriz de conexão **W** é muito esparsa.
- **Amplitude dos pesos do reservatório (W) :** os pesos do reservatório são inicializados

com valores aleatórios com amplitude na faixa de $[0,001-2]$.

- **Amplitude dos pesos da unidade de entrada ($\mathbf{W}^{in}$) e da saída projetada para as unidades internas ($\mathbf{W}^{back}$):** são novamente avaliados os pesos para estas matrizes de valores aleatórios com amplitude na faixa de $[0,001-2]$.
- **Duração do transitório:** os efeitos do estado inicial da rede são geralmente de curta duração, e estão diretamente ligados ao tamanho e complexidade do conjunto de dados utilizados. Desta forma, avalia-se o efeito do transitório na proporção de 0% a 50% do número total de dados utilizados para treinamento.
- **Variação do raio espectral do reservatório:** este parâmetro, o raio espectral α, é utilizado para mudar a escala da matriz $\mathbf{W}$. Deve-se utilizar um valor menor que 1, assim são avaliados valores entre $0,001$ e $0,5$.
- **Número de unidades do reservatório:** em geral utilizam-se valores grandes, maiores do que os utilizados nas redes baseadas na rede MLP. Deve-se apenas ter o cuidado para não escolher valores próximos do número de dados de treinamento. Jaeger (2001) recomenda que se escolham valores 10 vezes menores que o número de dados de treinamento, para evitar *overfitting*.
- **Entrada fixa como unidade de entrada:** este parâmetro é utilizado por Jaeger em algumas aplicações da rede ESN. Geralmente é utilizado um valor baixo, menor que 1, desta forma são avaliados valores na faixa de $[0,001...10]$.

Além dos parâmetros próprios da rede ESN, outros parâmetros que não são exclusivos da rede ESN também são otimizados, como a dimensão de imersão, o atraso de imersão e a ordem do regressor de saída.

6.4 Máquina de Aprendizado Extremo

Para compreender melhor o que será exposto nesta seção, deve-se levar em consideração algumas informações preliminares sobre a rede ELM. Inicialmente, nota-se que esta rede é do tipo *feedforward*, ou seja, sem realimentação, com apenas uma camada de neurônios ocultos e uma camada de neurônios de saída. Esta arquitetura de rede neural é equivalente à da rede MLP, porém apresenta uma fase de aprendizado mais rápida que a da rede MLP, uma vez que também lança mão da ideia de projeções aleatórias para construir a matriz de pesos que conecta as unidade de entrada aos neurônios ocultos.

A Figura 34 ilustra a rede ELM com a entrada $\mathbf{x}(n)$ definida de acordo com o teorema de Takens, com a matriz $\mathbf{W}$ representando os pesos da camada não-linear que permanecem fixos durante o treinamento e o vetor $\mathbf{m}$ representando os pesos do único neurônio de saída. A seguir, é feita uma descrição detalhada da arquitetura e do funcionamento da rede ELM.

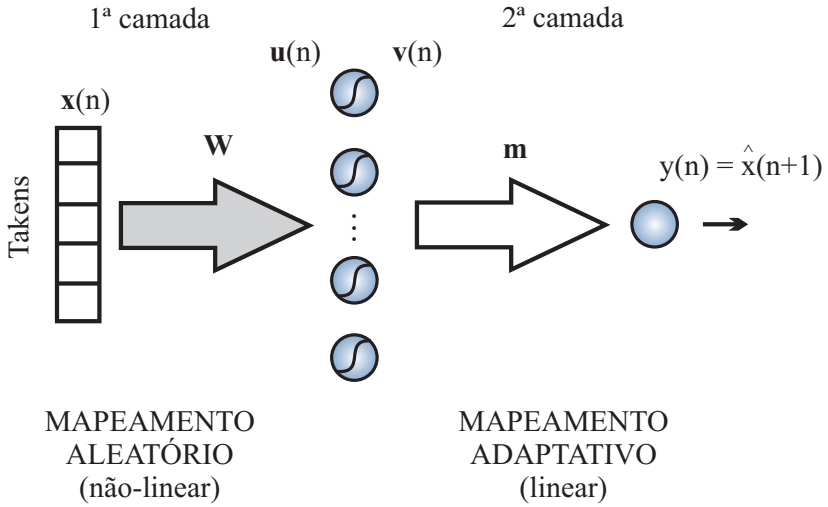


Figura 34 – Ilustração da arquitetura da rede ELM aplicada em predição de séries temporais.

Os neurônios da camada oculta (primeira camada de pesos sinápticos) são representados conforme mostrado na Figura 8(a), enquanto os neurônios da camada de saída (segunda camada de pesos sinápticos) são representados conforme mostrado na Figura 8(b).

O vetor de pesos associado a cada neurônio i da camada oculta, é representado como

$$\mathbf{w}_i = \begin{pmatrix} w_{i0} \\ w_{i1} \\ \vdots \\ w_{id_E} \end{pmatrix} = \begin{pmatrix} \theta_i \\ w_{i1} \\ \vdots \\ w_{id_E} \end{pmatrix}, \tag{6.8}$$

em que θ_i é o limiar associado ao neurônio i. De modo semelhante, o vetor de pesos associado a cada neurônio k da camada de saída é representado como

$$\mathbf{m}_k = \begin{pmatrix} m_{k0} \\ m_{k1} \\ \vdots \\ m_{kq} \end{pmatrix} = \begin{pmatrix} \theta_k \\ m_{k1} \\ \vdots \\ m_{kq} \end{pmatrix}, \tag{6.9}$$

em que θ_k é o limiar associado ao neurônio de saída k. Para aplicação em predição de séries temporais univariadas, usa-se somente um único neurônio na camada de saída, ou seja, faz-se $k = 1$.

Na rede ELM, o aprendizado é executado em três passos distintos, que são comentados a seguir.

(i) **Inicialização Aleatória dos Pesos da Camada Oculta**

O funcionamento da rede ELM envolve o cálculo das ativações e saídas de todos os neurônios da camada oculta e do neurônio da camada de saída, uma vez que os pesos $w_{ij}, i = 1, ..., q$ e $j = 0, ..., d_E$, tenham sido inicializados com valores aleatórios. Formalmente, pode-se escrever:

$$w_{ij} \sim U(a,b) \qquad \text{ou} \qquad w_{ij} \sim N(0,\sigma^2) \tag{6.10}$$

em que $U(a,b)$ é um número aleatório uniformemente distribuído no intervalo (a,b), enquanto $N(0,\sigma^2)$ é um número aleatório normalmente distribuído com média zero e variância σ^2.

Para isto, precisa-se definir uma matriz de pesos $\mathbf{W}$, com q linhas e $d_E + 1$ colunas:

$$\mathbf{W} = \begin{pmatrix} w_{10} & w_{11} & \dots & w_{1d_E} \\ w_{20} & w_{21} & \dots & w_{2d_E} \\ \vdots & \vdots & \vdots & \vdots \\ w_{q0} & w_{q1} & \dots & w_{qd_E} \end{pmatrix}_{q \times (d_E+1)}, \tag{6.11}$$

onde nota-se que a i-ésima linha da matriz $\mathbf{W}$ é composta pelo vetor de pesos do i-ésimo neurônio oculto.

(ii) **Acúmulo das Saídas dos Neurônios Ocultos**

Este passo corresponde a etapa de treinamento da rede, no qual se obtém as ativações dos neurônios ocultos e suas respectivas saídas. Este passo destina-se à obtenção de uma matriz

formada a partir das saídas dos neurônios da camada oculta, calculada conforme a apresentação de cada vetor de entrada à rede neural.

O fluxo de sinais (informação) se dá dos neurônios de entrada para os neurônios de saída, passando obviamente pelos neurônios da camada oculta. Por isto, diz-se que a informação está fluindo no sentido direto (forward), ou seja:

Entrada $\rightarrow$ Camada Oculta $\rightarrow$ Camada de saída

O vetor de entrada propriamente dito é definido como:

$$\mathbf{x}(n) = [x(n) \;\; x(n-\tau) \;\; \cdots \;\; x(n-(d_E-1)\tau)]^T, \tag{6.12}$$

em que $\mathbf{x}(n)$ segue o teorema de Takens (TAKENS, 1981) e é um vetor que contém d_E elementos da série, contados a partir do elemento atual $x(n)$, espaçados um do outro de τ unidades de tempo.

Assim, após a apresentação de um vetor de entrada $\mathbf{x}$, na iteração n, o primeiro passo é calcular as ativações dos neurônios da camada oculta:

$$u_i(n) = \sum_{j=1}^{d_E} w_{ij}x_j(n) - \theta_i(n) = \mathbf{w}_i^T\mathbf{x}(n), \quad i = 1,\ldots,q, \tag{6.13}$$

em que T indica o vetor (ou matriz) transposto, q indica o número de neurônios da camada oculta e $\theta_i(n)$ é o limiar do i-ésimo neurônio da camada oculta.

Em seguida, as saídas correspondentes são calculadas como

$$v_i(n) = \phi\left[u_i(n)\right] = \phi\left[\sum_{j=1}^{d_E} w_{ij}x_j(n) - \theta_i(n)\right], \tag{6.14}$$

em que $\phi(\cdot)$ é uma não-linearidade do tipo sigmoidal.

Vetorialmente, a Equação (6.14) pode ser escrita como

$$\mathbf{v}(n) = \phi(\mathbf{u}(n)) = \phi(\mathbf{W}\mathbf{x}(n)), \tag{6.15}$$

em que a função de ativação $\phi(\cdot)$ é aplicada a cada um dos q componentes do vetor $\mathbf{u}(n)$.

Para cada vetor de entrada $\mathbf{x}(n), n = 1,\ldots,N$, tem-se um vetor $\mathbf{v}(n)$ correspondente, que deve ser organizado (disposto) como uma coluna de uma matriz $\mathbf{V}$. Esta matriz terá q linhas

por N colunas:

$$\begin{aligned} \mathbf{V} &= [\mathbf{v}(1)\,|\mathbf{v}(2)|\,\ldots\,|\mathbf{v}(N)], \\ \mathbf{V} &= \begin{pmatrix} v_1(1) & v_1(2) & \cdots & v_1(N) \\ v_2(1) & v_2(2) & \cdots & v_2(N) \\ \vdots & \vdots & \vdots & \vdots \\ v_q(1) & v_q(2) & \cdots & v_q(N) \end{pmatrix}_{q\times N}. \end{aligned} \tag{6.16}$$

A matriz $\mathbf{V}$ será usada no Passo 3 para calcular os valores dos pesos dos neurônios de saída da rede ELM.

(*iii*) **Cálculo dos Pesos dos Neurônios de Saída**

Sabe-se que para cada vetor de entrada $\mathbf{x}(n), n = 1,\ldots,N$, tem-se um escalar de saída desejado $d(n)$ correspondente. Se estes N escalares forem organizado ao longo das linhas de um vetor $\mathbf{d}$, então este vetor possui dimensão N:

$$\mathbf{d} = \begin{pmatrix} d(1) \\ d(2) \\ \vdots \\ d(N) \end{pmatrix}. \tag{6.17}$$

Pode-se entender o cálculo dos pesos da camada de saída como o cálculo dos parâmetros de um mapeamento linear entre a camada oculta e a camada de saída. O papel do vetor de "entrada" para a camada de saída na iteração n é desempenhado pelo vetor $\mathbf{v}(n)$ enquanto a "saída" é representada pelo escalar $d(n)$. Assim, busca-se determinar o vetor $\mathbf{m}$ que melhor representa a transformação:

$$d(n) = \mathbf{m}^T \mathbf{v}(n). \tag{6.18}$$

Para isto, pode-se usar o método dos mínimos quadrados, também conhecido como método da pseudoinversa. Assim, usando a matriz $\mathbf{V}$ e o vetor $\mathbf{d}$, o vetor de pesos $\mathbf{m}$ é calculado por meio da seguinte expressão:

$$\mathbf{m} = [\mathbf{V}\mathbf{V}^T]^{-1}\mathbf{V}\mathbf{d}, \tag{6.19}$$

em que $\mathbf{m}$ é o vetor de pesos dos neurônios da camada de saída e possui dimensão q.

Singh e Balasundaram (2007) utilizam a rede ELM para predição de séries temporais caóticas. Alguns parâmetros são otimizados em termos do erro médio quadrático, tais como o

número de neurônios escondidos e a ordem da memória de entrada. Estes autores utilizaram as séries de Mackey-Glass, do Laser Caótico e de batimentos cardíacos. Sovilj *et al.* (2010) também utilizaram a rede ELM em tarefas de predições de longo prazo, estudando soluções para processamento da entrada e propondo métodos automáticos para determinação da dimensão da projeção aleatória.

6.4.1 Extensão da Rede ELM para Predição de Séries Temporais

A extensão da rede ELM, proposta a seguir, envolve a ideia já estabelecida para a rede NARX-MISO, discutida no Capítulo 5. A arquitetura sugerida utiliza os mesmos três passos enumerados para o treinamento da rede ELM original. A única diferença é a forma com que a entrada da rede ELM é construída.

6.4.1.1 Rede NARX-ELM

A rede NARX-ELM é baseada na arquitetura analisada no Capítulo 5, sendo que em vez da utilização do algoritmo *backpropagation* ela é obtida a partir da rede ELM. A rede NARX-ELM é construída através da redefinição da camada de entrada, que passa a ser dividida em duas partes. A primeira parte, corresponde ao vetor de entrada, definido conforme a Equação (6.12), que segue o Teorema de Takens. A segunda parte é formada pelo contexto definido pelo regressor de saída da rede NARX. Como discutido no Capítulo 5, esta rede pode ser treinada com o modo-SP, que pode ser escrito de acordo com:

$$\begin{aligned} \mathbf{y}_{sp}(n) &= [x(n)\ x(n-1)\ \cdots\ x(n-d_y+1)]^T \\ &= [x_1^c(n)\ x_2^c(n)\ \cdots\ x_{d_Y}^c(n)]^T, \end{aligned} \tag{6.20}$$

em que $\mathbf{y}_{sp}(n) \in \mathbb{R}^{d_y}$ é chamado de regressor de saída.

Assim, para uma rede NARX-ELM com q neurônios ocultos, as saídas destes neurônios são dadas por

$$v_i(n) = \phi[u_i(n)], \quad i = 1,\ldots,q, \tag{6.21}$$

em que u_i denota a ativação do i-ésimo neurônio oculto, definida como

$$u_i(n) = \sum_{j=1}^{d_E} w_{ij}(n)x_j(n) + \sum_{l=1}^{d_y} w_{il}^c(n)x_l^c(n) - \theta_i(n), \tag{6.22}$$

em que w_{ij} é o peso que conecta a j-ésima unidade de entrada ao i-ésimo neurônio oculto, w_{il}^{c} é o peso que conecta a l-ésima unidade de contexto ao i-ésimo neurônio oculto e $\theta_i(n)$ é o limiar do i-ésimo neurônio oculto.

Vetorialmente, as Equações (6.21) e (6.22) podem ser escritas como

$$\mathbf{v}(n) = \phi(\mathbf{u}(n)) = \phi(\mathbf{W}^1\mathbf{x}(n) + \mathbf{W}^c\mathbf{y}_{sp}(n)), \tag{6.23}$$

com $\mathbf{v}(n)$ e $\mathbf{u}(n)$ denotando, respectivamente, os vetores contendo as saídas e as ativações dos neurônios ocultos. A matriz de pesos $\mathbf{W^1}$ conecta as unidades de entrada aos neurônios ocultos, enquanto a matriz de pesos $\mathbf{W}^c$ conecta as unidades de contexto aos neurônios ocultos.

Note que o regressor de saída para o modo-SP mostrado na Equação (6.20) contém d_y valores passados da série temporal observada. Desta forma, a rede ELM com esta estrutura não possui recorrência na fase de treinamento, pois as duas partes que formam a unidade de entrada da rede são oriundas da própria série observada.

Por fim, o acúmulo das saídas dos neurônios ocultos e o cálculo dos pesos dos neurônios de saída seguem as mesmas considerações matemáticas definidas para a rede ELM original (Seção 6.4), descritos nas Equações (6.16) e (6.19), respectivamente.

A Figura 35 ilustra uma rede NARX-ELM com a entrada $\mathbf{x}(n)$ definida de acordo com o teorema de Takens, um contexto definido como regressores de saída conforme utilizado na rede NARX, as matrizes $\mathbf{W}^1$ e $\mathbf{W}^c$ representando os pesos da camada não-linear que são definidos aleatoriamente durante o treinamento e o vetor $\mathbf{m}$ representando os pesos ajustáveis do único neurônio de saída.

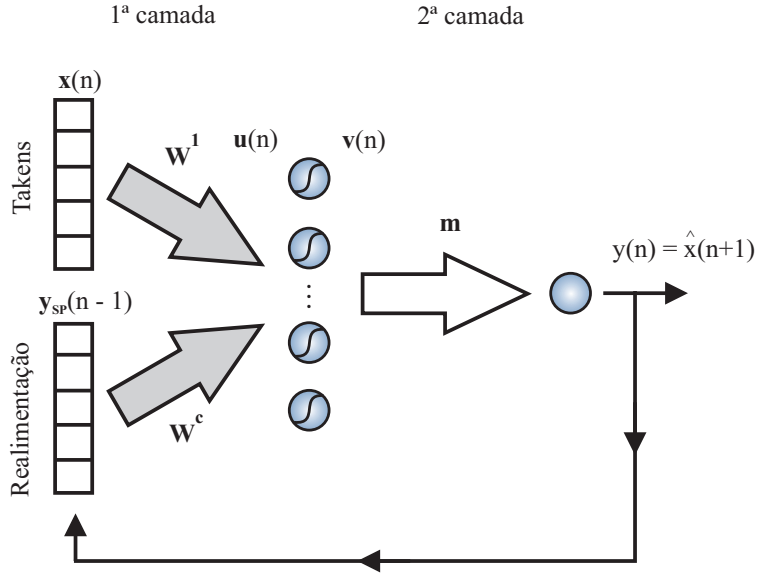


Figura 35 – Rede NARX-ELM aplicada em predição de séries temporais.

6.4.1.2 Trabalho Correlato

Em poucas palavras, pode-se dizer que a rede NARX-ELM é uma implementação da rede NARX usando o algoritmo de treinamento da rede ELM, em vez do algoritmo *backpropagation* simples, tal como proposto em Menezes-Júnior e Barreto (2008a).

Em Bouchachia (2009) uma ideia semelhante é proposta em que a rede NARX é implementada por meio da rede RBF, dando origem ao modelo NARX-RRBFN. O autor aplica a arquitetura NARX-RRBFN em problemas de predição múltiplos-passos-adiante utilizando duas séries temporais caóticas.

6.5 Conclusão

Este capítulo apresentou as arquiteturas de redes neurais baseadas em projeções aleatórias. A primeira arquitetura introduzida foi a rede ESN, que tem uma estrutura recorrente. A segunda arquitetura foi a rede ELM, que é apenas uma rede neural *feedforward*. Nestas redes os pesos das unidades de entrada para a camada oculta permanecem fixos em valores definidos a priori, enquanto os pesos da camada de saída são ajustáveis.

São propostas também variantes tanto para a rede ESN como para a rede ELM. Na rede ESN são feitos estudos dos laços de alimentação e realimentação da arquitetura, como também nos parâmetros de treinamento da rede. Já para a rede ELM é proposta uma modificação baseada na rede NARX-MISO. Em suma, este capítulo visa enumerar variantes das arquiteturas baseadas em projeções aleatórias.

O próximo capítulo traz as exposições das extensões da rede de Elman para tarefa de predição de múltiplos-passos-adiante de séries temporais univariadas.

7 EXTENSÕES DA REDE DE ELMAN

7.1 Introdução

A rede de Elman tem sido utilizada recentemente com resultados promissores, por exemplo, em Ardalani-Farsa e Zolfaghari (2010), onde os autores utilizam a rede de Elman para fazer predições de séries caóticas. Com o fim do treinamento, a rede de Elman é novamente utilizada para fazer predições dos resíduos, caso estes possuam comportamento caótico. Por fim, a rede NARX é utilizada para fazer predição utilizando os valores observados da série temporal e dos resíduos preditos pela rede de Elman.

Tampelini *et al.* (2011) utilizam a rede de Elman para fazer modelagem e predição de séries temporais de precipitação de chuva e assim fazer predição da vazão de uma bacia hidrográfica com base somente nos dados de precipitação. Wang e Gao (2011) aplicam a rede de Elman para analisar o nível de fósforo, nitrogênio e oxigênio e assim fazer predições da qualidade da água de um lago.

Neste capítulo são propostas extensões da rede de Elman a fim de melhorar o desempenho da rede de Elman clássica na tarefa de predição múltiplos-passos-adiante. Dentre as modificações propostas estão o uso de duas camadas ocultas, a realimentação ou da ativação ou da derivada da ativação de uma das camadas ocultas, e a implementação da rede de Elman usando o algoritmo de treinamento da rede ELM.

7.2 Rede Recorrente de ELMAN

Esta arquitetura recorrente foi introduzida por Elman (1990), sendo obtida a partir da rede MLP através da redefinição da camada de entrada da rede, que passa a ser dividida em duas partes. A primeira parte corresponde ao vetor de entrada propriamente dito, definido aqui, para fins de predição de séries temporais, como

$$\mathbf{x}(n) = [x(n) \;\; x(n-\tau) \;\; \cdots \;\; x(n-(d_E-1)\tau)]^T, \tag{7.1}$$

em que $\mathbf{x}(n) \in \mathbb{R}^{d_E}$ é um vetor que contém d_E elementos da série, contados a partir do valor atual $x(n)$, espaçados um do outro de τ unidades de tempo. No contexto da área de modelagem e predição de séries temporais, o parâmetro d_E é chamado de dimensão de imersão e o parâmetro τ é chamado de atraso de imersão. A definição do vetor $\mathbf{x}(n)$ na Equação (7.1) segue o teorema de

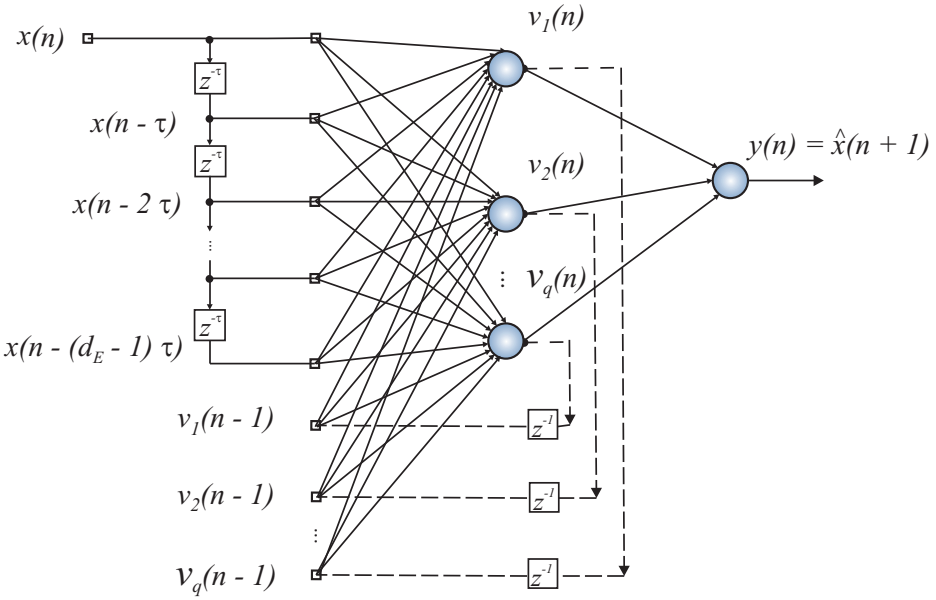


Figura 36 – Rede de Elman com uma camada oculta aplicada ao problema de predição de séries temporais univariadas.

Takens (TAKENS, 1981), para fim de reconstrução de atratores e na predição de séries temporais caóticas.

A segunda parte da entrada da rede de Elman contém as *unidades de contexto*, cujos valores são obtidos a partir da realimentação das saídas dos neurônios ocultos no instante $n-1$, ou seja

$$\begin{aligned} \mathbf{x}^c(n) &= [x_1^c(n)\ x_2^c(n)\ \cdots\ x_q^c(n)]^T \in \mathbb{R}^q \\ &= [v_1(n-1)\ v_2(n-1)\ \cdots\ v_q(n-1)]^T, \end{aligned} \tag{7.2}$$

em que $\mathbf{x}^c(n) \in \mathbb{R}^q$ é chamada de *vetor de contexto*.

A Figura 36 ilustra uma rede recorrente de Elman aplicada ao problema de predição de séries temporais. Vale ressaltar que a rede de Elman foi proposta originalmente como tendo apenas uma camada oculta. A formulação matemática que segue é, portanto, para este tipo de arquitetura. Assim, para uma rede de Elman com q neurônios ocultos, as saídas destes neurônios são dadas por

$$v_i(n) = \phi[u_i(n)] = \frac{1-\exp\{-u_i(n)\}}{1+\exp\{-u_i(n)\}}, \quad i = 1,\ldots,q \tag{7.3}$$

em que u_i denota a ativação do i-ésimo neurônio oculto, definida como

$$u_i(n) = \sum_{j=1}^{d_E} w_{ij}(n)x_j(n) + \sum_{l=1}^{q} w_{il}^c(n)x_l^c(n) - \theta_i(n), \tag{7.4}$$

em que w_{ij} é o peso que conecta a j-ésima unidade de entrada ao i-ésimo neurônio oculto, w_{il}^{c} é o peso que conecta a l-ésima unidade de contexto ao i-ésimo neurônio oculto e $\theta_i(n)$ é o limiar do i-ésimo neurônio oculto.

Por fim, a saída do neurônio de saída (apenas um neste caso) é dada por

$$y(n) = \phi[a(n)] = \frac{1-\exp\{-a(n)\}}{1+\exp\{-a(n)\}}, \tag{7.5}$$

em que $a(n)$ denota a ativação da unidade de saída, definida como

$$a(n) = \sum_{i=1}^{q} m_i(n)v_i(n) - \theta(n), \tag{7.6}$$

em que m_i é o peso que conecta a saída do i-ésimo neurônio oculto ao neurônio de saída.

Vetorialmente, as Equações (7.3) e (7.4) podem ser reescritas como

$$\mathbf{v}(n) = \phi[\mathbf{u}(n)], \tag{7.7}$$

onde

$$\mathbf{u}(n) = \mathbf{W}^1(n)\mathbf{x}(n) + \mathbf{W}^c(n)\mathbf{x}^c(n), \tag{7.8}$$

com $\mathbf{v}(n)$ e $\mathbf{u}(n)$ denotando, respectivamente, os vetores contendo as saídas e as ativações dos neurônios ocultos. A matriz de pesos $\mathbf{W^1} = [w_{ij}]_{q\times d_E}$ conecta as unidades de entrada aos neurônios ocultos, enquanto a matriz de pesos $\mathbf{W}^c = [w_{il}^c]_{q\times q}$ conecta as unidades de contexto aos neurônios ocultos.

Já as Equações (7.5) e (7.6), que descrevem a saída da rede, podem ser reescritas vetorialmente como:

$$y(n) = \phi[a(n)], \tag{7.9}$$

onde

$$a(n) = \mathbf{m}(n)\mathbf{v}(n). \tag{7.10}$$

com $y(n)$ e $a(n)$ denotando, respectivamente, a saída e a ativação da unidade de saída. O vetor de pesos $\mathbf{m} = [m_i]_{q\times 1}$ conecta os neurônios ocultos à única unidade de saída.

Por simplicidade, denota-se a rede recorrente de Elman como sendo Elman($d_E + q$, q,1), a fim de destacar as seguintes grandezas: a dimensão do vetor de entrada (d_E), a dimensão do vetor de contexto (q), o número de neurônios ocultos (q) e o número de neurônios de saída

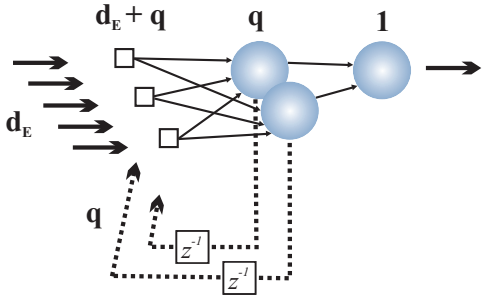


Figura 37 – Rede de Elman com realimentação das ativações da camada oculta.

(apenas um, neste caso). Uma outra forma de destacar estas grandezas é através da Figura 37, onde estão destacados, de modo mais compacto, todas as grandezas da rede de Elman com uma camada oculta.

Mais uma vez, vale mencionar que a rede de Elman original possui apenas uma camada oculta. Nesta tese, são propostas extensões da rede de Elman tanto para arquiteturas de uma camada, quanto para arquiteturas com duas camadas ocultas. As extensões para duas camadas ocultas têm, naturalmente, o custo computacional aumentado, em função do aumento do número de parâmetros ajustáveis. O treinamento também passa a ser mais longo. Resta saber se a melhoria no desempenho compensa o aumento dos custo.

Um dos objetivos desta tese é justamente comparar os desempenhos das arquiteturas propostas e avaliar se há ganhos na capacidade preditiva das redes recorrentes resultantes que compensem o custo computacional adicional. Todas as extensões da rede de Elman avaliadas neste capítulo são descritas a seguir.

7.3 Variantes da Rede de Elman

As primeiras extensões da rede de Elman a serem propostas envolvem a arquitetura original com uma camada oculta. Em seguida, são descritas as extensões da rede de Elman para uma arquitetura com duas camadas ocultas.

7.3.1 Redes com Uma Camada Oculta

A rede de Elman original realimenta, para as unidades de contexto, as saídas dos neurônios ocultos no instante anterior ($n-1$). A extensão aqui proposta realimenta as derivadas das saídas (funções de ativação) dos neurônios da camada oculta no instante $n-1$. A principal razão para realimentar as derivadas é que a derivada é uma fonte de informação da dinâmica do sistema, que pode reter informação temporal no laço de realimentação ao longo do tempo.

Matematicamente, o vetor de contexto passa a ser definido como

$$\begin{aligned} \mathbf{x}^c(n) &= [x_1^c(n)\ x_2^c(n)\ \cdots\ x_q^c(n)]^T \in \mathbb{R}^q \\ &= [v_1'(n-1)\ v_2'(n-1)\ \cdots\ v_q'(n-1)]^T, \end{aligned} \tag{7.11}$$

em que a derivada da função de ativação tangente hiperbólica do i-ésimo neurônio oculto é calculado por, no instante n,

$$v_i'(n) = \frac{1}{2}\left[1 - v_i^2(n)\right], \tag{7.12}$$

sendo a saída $v_i(n)$ calculada conforme a Equação (7.3). Esta extensão da rede de Elman será denotada doravante de

$$D\text{-Elman}(d_E + q,\, q,\, 1),$$

onde o prefixo D é para lembrar que as derivadas das funções de ativação dos neurônios ocultos são realimentadas para as unidades de contexto. A Figura 38 exibe, simplificadamente, esta variante da rede de Elman, onde o bloco ϕ' representa as derivadas das funções de ativação dos neurônios ocultos.

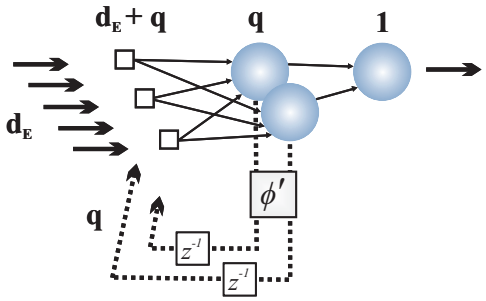


Figura 38 – D-Elman$(d_E + q, q, 1)$, rede de Elman com realimentação das derivadas das ativações da camada oculta.

7.3.2 Redes com Duas Camadas Ocultas

Com duas camadas ocultas, pode-se escolher a partir de qual camada oculta realimentar as saídas (ou suas derivadas) dos neurônios. Isto posto, quatro variantes são propostas:

- (*i*) realimentar as ativações da primeira camada oculta para as unidades de contexto,
- (*ii*) realimentar as derivadas das ativações da primeira camada oculta para as unidades de contexto,
- (*iii*) realimentar as ativações da segunda camada oculta para as unidades de contexto,

- (*iv*) realimentar as derivadas das ativações da segunda camada oculta para as unidades de contexto.

(*i*) **Realimentando as Ativações da 1^a Camada Oculta**: Esta extensão da rede de Elman será denotada por

$$\text{Elman}(d_E + q_1,\, q_1, q_2,\, 1),$$

onde q_1 (q_2) simboliza o número de neurônios da primeira (segunda) camada oculta. O vetor de contexto passa a ser definido como

$$\begin{aligned} \mathbf{x}^c(n) &= [x_1^c(n)\;\; x_2^c(n)\;\; \cdots\;\; x_{q_1}^c(n)]^T \in \mathbb{R}^{q_1} \\ &= [v_1(n-1)\;\; v_2(n-1)\;\; \cdots\;\; v_l(n-1)\;\; \cdots\;\; v_{q_1}(n-1)]^T, \end{aligned} \tag{7.13}$$

em que $v_l(n-1)$ é a saída do l-ésimo neurônio da primeira camada oculta, no instante $n-1$. A Figura 39(a) traz um esquema ilustrativo da arquitetura da rede Elman($d_E + q_1$, q_1,q_2, 1).

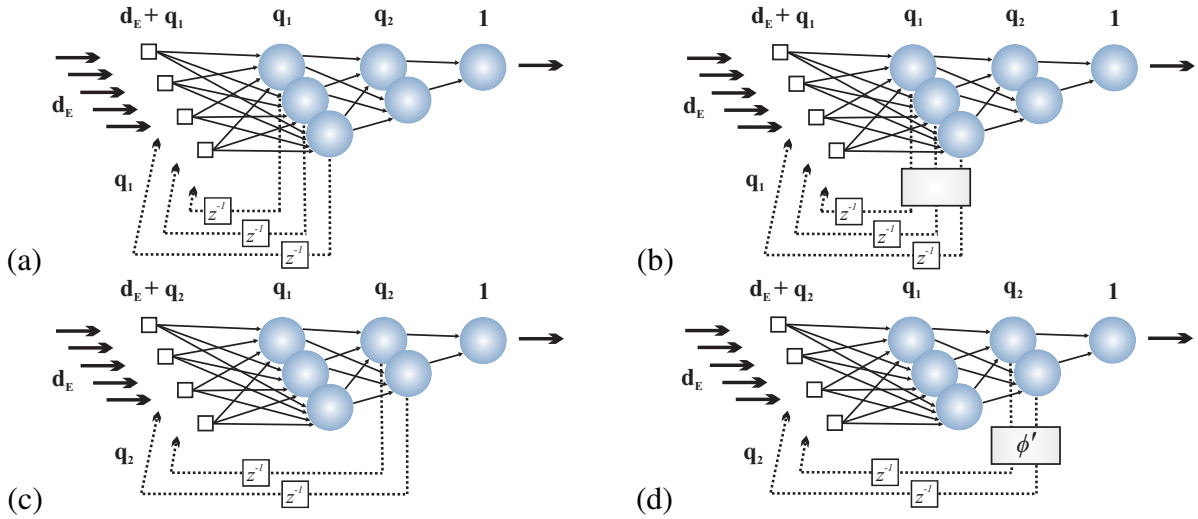


Figura 39 – Variantes da rede de Elman. (a) Elman($d_E + q_1$, q_1,q_2, 1), (b) D_1-Elman($d_E + q_1$, q_1,q_2, 1), (c) Elman($d_E + q_2$, q_1,q_2, 1), (d) D_2-Elman($d_E + q_2$, q_1,q_2, 1).

A ativação do i-ésimo neurônio da primeira camada oculta da rede Elman($d_E + q_1$, q_1,q_2, 1) é definida como

$$u_i^{(1)}(n) = \sum_{j=1}^{d_E} w_{ij}^{(1)}(n) x_j(n) + \sum_{l=1}^{q_1} w_{il}^c(n) x_l^c(n) - \theta_i^{(1)}(n), \quad i = 1, \ldots, q1, \tag{7.14}$$

em que $w_{ij}^{(1)}$ é o peso que conecta a j-ésima unidade de entrada ao i-ésimo neurônio da primeira camada oculta, w_{il}^c é o peso que conecta a l-ésima unidade de contexto ao i-ésimo neurônio da primeira camada oculta e $\theta_i(n)$ é o limiar do i-ésimo neurônio da primeira camada oculta.

Assim, para a rede de Elman com q_1 neurônios na primeira camada oculta, as saídas destes neurônios são dadas por

$$v_i(n) = \phi[u_i^{(1)}(n)] = \frac{1-\exp\{-u_i^{(1)}(n)\}}{1+\exp\{-u_i^{(1)}(n)\}}, \quad i = 1,\ldots,q1, \tag{7.15}$$

em que $u_i^{(1)}$ denota a ativação do i-ésimo neurônio da primeira camada oculta.

A ativação do k-ésimo neurônio da segunda camada oculta é definida como

$$u_k^{(2)}(n) = \sum_{i=1}^{q_1} w_{ki}^{(2)}(n)v_i(n) - \theta_k^{(2)}(n), \quad k = 1,\ldots,q2, \tag{7.16}$$

em que $w_{ki}^{(2)}$ é o peso que conecta o i-ésimo neurônio da primeira camada oculta ao k-ésimo neurônio da segunda camada oculta e $\theta_k^{(2)}(n)$ é o limiar do k-ésimo neurônio da segunda camada oculta.

A saída do k-ésimo neurônio da segunda camada oculta é dado por

$$z_k(n) = \phi[u_k^{(2)}(n)] = \frac{1-\exp\{-u_k^{(2)}(n)\}}{1+\exp\{-u_k^{(2)}(n)\}}, \quad k = 1,\ldots,q2, \tag{7.17}$$

em que $u_k^{(2)}$ denota a ativação do k-ésimo neurônio da segunda camada oculta.

A ativação do único neurônio de saída é calculado da seguinte forma

$$a(n) = \sum_{k=1}^{q_2} m_k(n)z_k(n) - \theta^{(3)}(n), \tag{7.18}$$

em que m_k é o peso que conecta a saída do k-ésimo neurônio da segunda camada oculta ao neurônio de saída.

Por fim, a saída é definida como

$$y(n) = \phi[a(n)]. \tag{7.19}$$

(*ii*) **Realimentando as Derivadas das Ativações da 1^a Camada Oculta**: Esta extensão da rede de Elman será denotada por

$$D_1\text{-Elman}(d_E + q_1,\, q_1, q_2,\, 1),$$

em que o prefixo D_1 indica que as derivadas das funções de ativação dos neurônios da primeira camada oculta é que devem ser realimentadas para as unidades de contexto. O vetor de contexto passa a ser definido como

$$\begin{aligned} \mathbf{x}^c(n) &= [x_1^c(n)\ x_2^c(n)\ \cdots\ x_{q_1}^c(n)]^T \in \mathbb{R}^{q_1} \\ &= [v_1'(n-1)\ v_2'(n-1)\ \cdots\ v_l'(n-1)\ \cdots\ v_{q_1}'(n-1)]^T, \end{aligned} \tag{7.20}$$

em que $v_l'(n-1)$ é a derivada da função de ativação do l-ésimo neurônio ($l = 1, \cdots, q_1$) da primeira camada oculta, no instante $n-1$. Na Figura 39(b) mostra-se uma ilustração da arquitetura D_1-Elman($d_E + q_1$, q_1,q_2, 1).

Para a rede D_1-Elman($d_E + q_1$, q_1,q_2, 1), a formulação matemática é a mesma da rede Elman($d_E + q_1$, q_1,q_2, 1), exceto pela redefinição do vetor de contexto como está na Equação (7.20).

(*iii*) **Realimentando as Ativações da 2^a Camada Oculta** - Esta extensão da rede de Elman será denotada por

$$\text{Elman}(d_E + q_2, q_1, q_2, 1),$$

onde q_1 (q_2) simboliza o número de neurônios da primeira (segunda) camada oculta. O vetor de contexto passa a ser definido como

$$\begin{aligned} \mathbf{x}^c(n) &= [x_1^c(n)\ x_2^c(n)\ \cdots\ x_{q_2}^c(n)]^T \in \mathbb{R}^{q_2} \\ &= [z_1(n-1)\ z_2(n-1)\ \cdots\ z_l(n-1)\ \cdots\ z_{q_2}(n-1)]^T, \end{aligned} \tag{7.21}$$

em que $z_l(n-1)$ é a saída do l-ésimo neurônio ($l = 1, \cdots, q_2$) da segunda camada oculta, no instante $n-1$. A Figura 39(c) exibe a arquitetura da rede Elman($d_E + q_2$, q_1, q_2, 1).

A ativação do i-ésimo neurônio da primeira camada oculta da rede Elman($d_E + q_1$, q_1,q_2, 1) é definida como

$$u_i^{(1)}(n) = \sum_{j=1}^{d_E} w_{ij}^{(1)}(n) x_j(n) + \sum_{l=1}^{q_2} w_{il}^c(n) x_l^c(n) - \theta_i^{(1)}(n), \quad i = 1, \ldots, q1, \tag{7.22}$$

em que $w_{ij}^{(1)}$ é o peso que conecta a j-ésima unidade de entrada ao i-ésimo neurônio da primeira camada oculta, w_{il}^c é o peso que conecta a l-ésima unidade de contexto ao i-ésimo neurônio da primeira camada oculta e $\theta_i(n)$ é o limiar do i-ésimo neurônio da primeira camada oculta. É importante ressaltar que a única diferença da Equação (7.22) para a Equação (7.14) da rede Elman($d_E + q_1$, q_1,q_2, 1) é o termo superior do segundo somatório.

As saídas dos neurônios da primeira camada oculta são dadas por

$$v_i(n) = \phi[u_i^{(1)}(n)] = \frac{1 - \exp\{-u_i^{(1)}(n)\}}{1 + \exp\{-u_i^{(1)}(n)\}}, \quad i = 1, \ldots, q1, \tag{7.23}$$

em que $u_i^{(1)}$ denota a ativação do i-ésimo neurônio da primeira camada oculta.

As ativações e saídas dos neurônios da segunda camada oculta, bem como a ativação e saída do neurônio de saída são os mesmos da rede Elman($d_E + q_1$, q_1,q_2, 1), dados pelas Equações (7.16), (7.17), (7.18) e (7.19), respectivamente.

(*iv*) **Realimentando as Derivadas das Ativações da 2^a Camada Oculta**: Por fim, esta extensão da rede de Elman será denotada por

$$D_2\text{-Elman}(d_E + q_2,\, q_1,\, q_2,\, 1),$$

em que o prefixo D_2 serve para lembrar que as derivadas das funções de ativação dos neurônios da segunda camada oculta é que devem ser realimentadas. O vetor de contexto passa a ser definido como

$$\begin{aligned} \mathbf{x}^c(n) &= [x_1^c(n)\ x_2^c(n)\ \cdots\ x_{q_2}^c(n)]^T \in \mathbb{R}^{q_2} \\ &= [z_1'(n-1)\ z_2'(n-1)\ \cdots\ z_l'(n-1)\ \cdots\ z_{q_2}'(n-1)]^T, \end{aligned} \tag{7.24}$$

em que $z_l'(n-1)$ é a derivada da função de ativação do l-ésimo neurônio ($l = 1, \cdots, q_2$) da segunda camada oculta, no instante $n-1$. Na Figura 39(d) mostra-se uma ilustração da arquitetura D_2-Elman($d_E + q_2$, q_1, q_2, 1).

Para a rede D_1-Elman($d_E + q_2$, q_1,q_2, 1) a formulação matemática é a mesma da rede Elman($d_E + q_2$, q_1,q_2, 1), exceto pela redefinição do vetor de contexto como está na Equação (7.24).

7.4 Extensões da Rede de Elman Usando o Modelo da rede ELM

As extensões da rede Elman propostas a seguir utilizam o modelo da rede ELM introduzido no Capítulo 6. As redes envolvem as ideias já estabelecidas para a rede NARX-MISO e para a rede de Elman. Todas as arquiteturas propostas utilizam os mesmos três passos enumerados para o treinamento da rede ELM original. A única diferença se encontra nas realimentações e na forma com que as unidades de entrada são construídas.

Naturalmente as extensões da rede de Elman usando o modelo da rede ELM propostas tem o custo computacional aumentado, uma vez que a dimensão da unidade de entrada é maior do que a rede sem realimentações. No entanto esta dificuldade pode ser desprezível pela velocidade com que a rede ELM é treinada. Desta forma, um dos objetivos desta tese é justamente comparar os desempenhos das arquiteturas propostas e avaliar se há ganhos na capacidade preditiva das

redes recorrentes resultantes que compensem a complexidade adicional. As extensões da rede ELM avaliadas nesta tese são descritas a seguir.

7.4.1 Rede ELMAN-ELM

Esta arquitetura recorrente é baseada na arquitetura analisada na Seção 7.2, sendo que ao invés da utilização do algoritmo *backpropagtion*, é obtida a partir da rede ELM básica. A rede ELMAN-ELM é construída através da redefinição da camada de entrada da rede, que passa a ser dividida em duas partes. A primeira parte corresponde ao vetor de entrada definido conforme a Equação (7.1), que segue o Teorema de Takens. A segunda parte da entrada da rede de Elman contém as unidades de contexto, cujos valores são obtidos a partir da realimentação das saídas dos neurônios ocultos no instante $n-1$, ou seja

$$\begin{aligned} \mathbf{x}^c(n) &= [x_1^c(n)\ x_2^c(n)\ \cdots\ x_q^c(n)]^T \in \mathbb{R}^q \\ &= [v_1(n-1)\ v_2(n-1)\ \cdots\ v_q(n-1)]^T, \end{aligned} \tag{7.25}$$

em que $\mathbf{x}^c(n) \in \mathbb{R}^q$ é chamada de vetor de contexto.

Assim, para uma rede ELMAN-ELM com q neurônios ocultos, as saídas destes neurônios são dadas por

$$v_i(n) = \phi[u_i(n)], \quad i = 1,\ldots,q \tag{7.26}$$

em que u_i denota a ativação do i-ésimo neurônio oculto, definida como

$$u_i(n) = \sum_{j=1}^{d_E} w_{ij}(n)x_j(n) + \sum_{l=1}^{q} w_{il}^c(n)x_l^c(n) - \theta_i(n), \tag{7.27}$$

em que w_{ij} é o peso que conecta a j-ésima unidade de entrada ao i-ésimo neurônio oculto, w_{il}^c é o peso que conecta a l-ésima unidade de contexto ao i-ésimo neurônio oculto e $\theta_i(n)$ é o limiar do i-ésimo neurônio oculto.

Vetorialmente, as Equações (7.26) e (7.27) podem ser escritas como

$$\mathbf{v}(n) = \phi(\mathbf{u}(n)) = \phi(\mathbf{W}^1\mathbf{x}(n) + \mathbf{W}^c\mathbf{x}^c(n)), \tag{7.28}$$

com $\mathbf{v}(n)$ e $\mathbf{u}(n)$ denotando, respectivamente, os vetores contendo as saídas e as ativações dos neurônios ocultos.

Por fim, o acúmulo das saídas dos neurônios ocultos e o cálculo dos pesos dos neurônios de saída seguem as mesmas considerações matemáticas definidas para a rede ELM original (Seção 6.4), descritos nas Equações (6.16) e (6.19), respectivamente.

A Figura 40 ilustra uma rede ELMAN-ELM com a entrada $\mathbf{x}(n)$ definida de acordo com o Teorema de Takens, um contexto de realimentação formado pela saída dos neurônios ocultos ($\mathbf{x}^c(n) = \mathbf{v}(n-1)$), as matrizes $\mathbf{W}^1$ e $\mathbf{W}^c$ representando os pesos da camada não-linear que são definidos aleatoriamente durante o treinamento e o vetor $\mathbf{m}$ representando os pesos do único neurônio de saída.

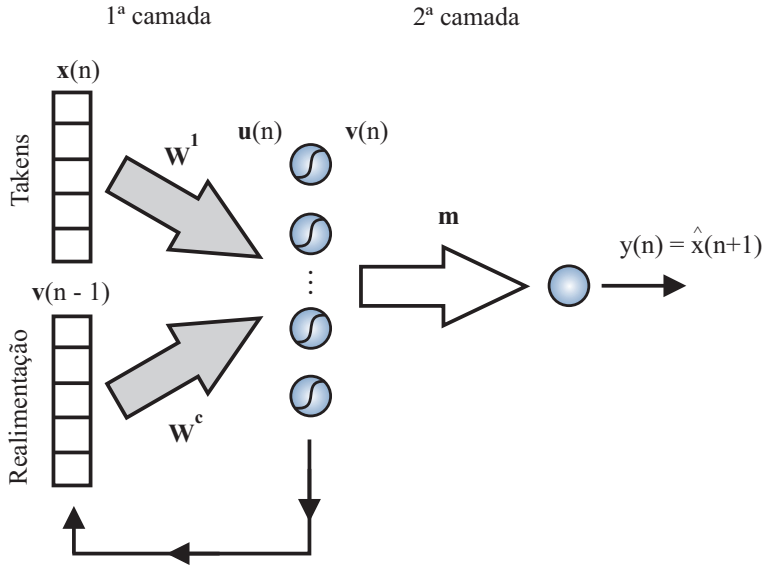


Figura 40 – Rede ELMAN-ELM aplicada em predição de séries temporais.

7.4.2 Rede Híbrida ELMAN/NARX-ELM

Nesta seção, é proposto um modelo híbrido, que consiste de uma recorrência da saída dos neurônios da camada oculta e uma recorrência da camada de saída da rede. Esta arquitetura recorrente é a combinação das propostas da rede NARX-ELM e da rede ELMAN-ELM, sendo assim baseada na rede NARX-MISO e na rede de Elman ao mesmo tempo. Esta rede é obtida a partir da rede ELM básica através da redefinição da camada de entrada, que passa a ser dividida em três partes:

1. A primeira parte, vetor $\mathbf{x}(n) \in \mathbb{R}^{d_E}$, corresponde ao vetor de entrada definido conforme a Equação 7.1, que segue o Teorema de Takens .
2. A segunda parte da entrada da rede ELMAN/NARX-ELM corresponde as unidades do primeiro contexto, cujos valores são obtidos a partir da realimentação das saídas dos neurônios ocultos no instante $n-1$.

$$\begin{aligned} \mathbf{x}^{c1}(n) = \mathbf{v}(n-1) &= [v_1(n-1)\ \ v_2(n-1)\ \ \cdots\ \ v_q(n-1)]^T \in \mathbb{R}^q \\ &= [x_1^{c1}(n)\ \ x_2^{c1}(n)\ \ \cdots\ \ x_q^{c1}(n)]^T, \end{aligned} \tag{7.29}$$

3. A terceira parte corresponde ao segundo contexto, definido pelo regressor de saída da rede NARX .

$$\begin{aligned}\mathbf{x}^{c2}(n) = \mathbf{y}_{sp}(n) &= [x(n)\ x(n-1)\ \cdots\ x(n-d_y+1)]^T \in \mathbb{R}^{d_y} \\ &= [x_1^{c2}(n)\ x_2^{c2}(n)\ \cdots\ x_{d_Y}^{c2}(n)]^T\end{aligned} \tag{7.30}$$

Assim, para uma rede ELMAN/NARX-ELM com q neurônios ocultos, as saídas destes neurônios são dadas por

$$v_i(n) = \phi[u_i(n)], \quad i = 1,\ldots,q \tag{7.31}$$

em que u_i denota a ativação do i-ésimo neurônio oculto, definida como

$$u_i(n) = \sum_{j=1}^{d_E} w_{ij}(n)x_j(n) + \sum_{l=1}^{q} w_{il}^{c1}(n)x_l^{c1}(n) + \sum_{m=1}^{d_y} w_{im}^{c2}(n)x_m^{c2}(n) - \theta_i(n), \tag{7.32}$$

em que w_{ij} é o peso que conecta a j-ésima unidade de entrada ao i-ésimo neurônio oculto, w_{il}^{c1} é o peso que conecta a l-ésima unidade do primeiro contexto ao i-ésimo neurônio oculto, w_{im}^{c2} é o peso que conecta a l-ésima unidade do segundo contexto ao i-ésimo neurônio oculto e $\theta_i(n)$ é o limiar do i-ésimo neurônio oculto.

Vetorialmente, as Equações (7.31) e (7.32) podem ser escritas como

$$\mathbf{v}(n) = \phi(\mathbf{u}(n)) = \phi(\mathbf{W}^1\mathbf{x}(n) + \mathbf{W}^{c1}\mathbf{x}^{c1}(n) + \mathbf{W}^{c2}\mathbf{y}_{sp}(n)), \tag{7.33}$$

com $\mathbf{v}(n)$ e $\mathbf{u}(n)$ denotando, respectivamente, os vetores contendo as saídas e as ativações dos neurônios ocultos.

Esta nova abordagem tem por objetivo aliar a capacidade de modelar dependências temporais de curto prazo da rede de Elman, com a capacidade de modelar dependências de longo prazo da rede NARX. Com isto, tenta-se verificar se esta combinação de informações pode trazer algum ganho na tarefa de predição múltiplos-passos-adiante.

7.5 Conclusão

Neste capítulo foram propostas algumas extensões da rede recorrente de Elman treinadas com o *backpropagation* voltadas para predição de séries temporais univariadas. Dentre as modificações propostas estão o uso de duas camadas ocultas, a realimentação ou da ativação ou da derivada da ativação de uma das camadas ocultas. Também são propostas redes baseadas na implementação da rede de Elman usando o algoritmo de treinamento da rede ELM.

O próximo capítulo apresenta os resultados das simulações computacionais das RNAs utilizando o modelo NARX-MISO e sua extensão para o modelo NARX-MIMO. É avaliada a capacidade preditiva destas redes comparando com outras arquiteturas neurais, tais como, FTDNN e Elman, na tarefa de predição múltiplos-passos-adiante.

8 RESULTADO PARA AS REDES NARX-MISO E NARX-MIMO

8.1 Introdução

Este capítulo tem o propósito de avaliar o desempenho das redes NARX-MISO e NARX-MIMO em relação aos de outras arquiteturas neurais, tais como FTDNN e Elman, na tarefa de predição recursiva, usando a série do laser caótico e a série de precipitação mensal de chuva na cidade de Fortaleza. O objetivo deste estudo é ressaltar as diferenças no projeto de tais arquiteturas neurais recorrentes, a fim de oferecer subsídios ao usuário no momento da escolha da arquitetura mais adequada à tarefa de interesse.

8.2 Resultados para a Série de Precipitação de Chuva

Nesta seção, o objetivo é avaliar a habilidade da rede NARX-MISO na predição de precipitação mensal de chuva na cidade de Fortaleza. Assim, uma comparação do desempenho da predição de longo prazo das redes TDNN e Elman, e também uma comparação com o modelo AR (Box-Jenkins) é realizado. Também é apresentada, nesta seção, a continuação dos experimentos iniciados no Capitulo 4. Desta forma a heurística formalizada no capítulo anterior é expandida para as redes de Elman e NARX-MISO, com uma ou duas camadas ocultas.

Para o treinamento das redes, a série temporal é redimensionada para a faixa $[-1,1]$. Não é removida da série nenhuma tendência nem sazonalidade. A série redimensionada é dividida em dois conjuntos para a realização da validação *holdout*, de modo que as primeiras 396 amostras são usadas para o treinamento e as 12 amostras restantes para o teste (predição um ano a frente).

Para o modelo AR, a série temporal é, primeiramente, transformada pelo método Box-Cox para reduzir a variação de dados e torná-lo mais distribuidamente Gaussiana:

$$Z_t^+ = \begin{cases} (Z_t^\lambda - 1)/\lambda, & \text{if } \lambda \neq 0 \\ \log Z_t, & \text{if } \lambda = 0 \end{cases} \tag{8.1}$$

onde Z_t é a observação original, Z_t^+ é a observação transformada e λ é o expoente de transformação. Usando o método de Hinkley (HINKLEY, 1977) é estimado o valor ótimo de $\lambda = 0.25$. Em seguida, uma transformação para retirar a sazonalidade é aplicada na série temporal $\{Z_t^+\}$:

$$Z_t^{++} = \frac{Z_{t(r,m)}^+ - \mu_m}{\sigma_m}, \tag{8.2}$$

onde m é o indice do mês ($m = 1, \ldots, 12$), r é o índice do ano ($r = 1, \ldots, 34$), μ_m e σ_m são, respectivamente, a média e desvio padrão da precipitação de chuva do m-th mês para todos os anos, e $Z^{+}_{t(r,m)}$ é a observação correspondente para o m-th mês do ano r. Finalmente, um modelo AR(1) é ajustado para a série temporal $\{Z_t^{++}\}$, cuja ordem foi encontrada via método AIC.

Todas as redes neurais possuem uma (com N_1 neurônios) ou duas camadas ocultas (com N_1 e N_2 neurônios, respectivamente) e um neurônio de saída. Todos os neurônios utilizam função de ativação tangente hiperbólica. O número de neurônios em cada camada oculta é determinado segundo a metodologia empregada na Seção 4.4. O algoritmo *backpropagation* padrão é utilizado para treinar as redes utilizando a predição um-passo-adiante. Para a rede de Elman, apenas as saídas dos neurônios na primeira camada oculta são alimentadas de volta para a camada de entrada. A taxa de aprendizado e o número de épocas também é ajustado pela metodologia empregada na Seção 4.4.

Depois que uma determinada rede tenha sido treinada, ela passa a fornecer predições de valores futuros de uma série temporal num determinado horizonte de predição h. As predições são executadas no modo recursivo até que um horizonte desejado de predição seja atingido, ou seja, durante h passos os valores previstos são realimentados de volta para a entrada do modelo.

Por razões de precisão estatística, cada rodada de treino/teste de cada modelo é repetido K=100 vezes. Quantitativamente, para cada k-th rodada de treino/teste, os modelos são avaliados em termos do NMSE. Assim, para encontrar a melhor configuração do modelo, é utilizada a metodologia adotada na Seção 4.4 e os resultados podem ser visualizados na Tabela 5 para o caso de duas camadas ocultas e na Tabela 6 para o caso de uma camada oculta.

Tendo o método de Cao e da informação mútua indicado os possíveis valores para d_E e τ, é decidido testar todas as combinações na faixa de $d_E \in [2...18]$ e $\tau \in [2...9]$. Já o número de neurônios nas camadas ocultas são escolhidos na faixa de N_1 e $N_2 \in [2...20]$. Por fim, para a dimensão do contexto da rede NARX-MISO é utilizada a faixa de $d_y \in [1...20]$. Para todas as investigações, escolhe-se o par os parâmetros que levem ao menor valor do NMSE.

Novamente, assim como na Seção 4.5, os pares encontrados para dimensão e atraso de imersão confirmam os valores sugeridos na Figura 16, sendo que agora os pares (d_E, τ) são determinados para outras redes neurais utilizadas. A dimensão (d_E determinada está entre 14 e 10, já para o atraso de imersão o valor determinado é de $\tau = 4$ para quase todas as redes, com exceção da rede NARX-MISO que teve $\tau = 6$. A respeito da rede NARX-MISO, o valor $d_y = 9$ foi o mais adequado para os modelos utilizados.

Tabela 5 – Busca por variáveis ótimas para as redes FTDNN, Elman e NARX-MISO com 2 camadas ocultas.

laço	max. épocas	tx. aprendizagem	dimensão de imersão	atraso de imersão	neurônios 1 camada	neurônios 2 camada	dimensão (d_y) NARX-MISO
FTDNN							
1	90	0,05	11	4	6	10	-
2	100	0,05	11	4	8	8	-
3	90	0,05	11	4	8	8	-
4	90	0,05	11	4	8	6	-
5	80	0,05	11	4	4	6	-
6	80	0,05	11	4	4	6	-
7	80	0,05	11	4	4	6	-
Elman							
1	20	0,05	12	4	8	6	-
2	20	0,05	12	4	14	10	-
3	20	0,05	14	4	14	10	-
4	20	0,05	14	4	14	10	-
5	20	0,05	14	4	12	12	-
6	20	0,05	14	4	12	12	-
7	20	0,05	14	4	12	12	-
NARX-MISO							
1	130	0,01	7	6	10	10	9
2	70	0,01	10	6	14	12	9
3	40	0,05	9	6	12	12	9
4	40	0,05	9	6	14	8	9
5	50	0,05	10	6	10	8	9
6	50	0,05	10	6	10	8	9
7	50	0,05	10	6	10	8	9

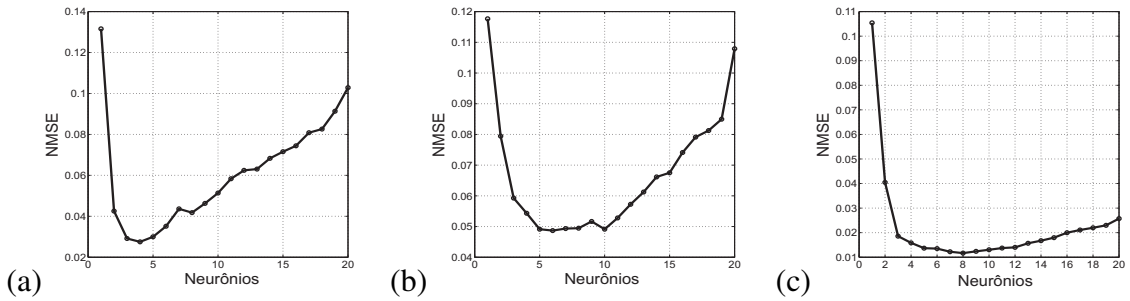
Na Figura 41 avalia-se a influência do número de neurônios na camada oculta no desempenho para os modelos com apenas uma camada oculta. Pode-se verificar uma maior robustez da escolha do número de neurônios da camada oculta para o modelo NARX-MISO em relação as outras duas redes testadas (FTDNN e Elman). O pior caso ficou para a relação NMSE versus o número de neurônios da rede FTDNN, onde a curva possui um concavidade bem acentuada, com valor mínimo em 4 neurônios e valores fora dessa região aumentam rapidamente os erros de predição.

Tendo sido selecionados e testados todos os modelos, o próximo objetivo é avaliar o desempenho da predição em termos do valor do NMSE de todas as redes neurais avaliadas. O Boxplot[1] para os valores de $NMSE(h,k)$, obtidos para $h = 12$ e $l = 1,\ldots,100$, é mostrado na Figura 42. O modelo autoregressivo (AR) não apresenta variação de resultados pois seus parâmetros são computados pelo método dos mínimos quadrados.

[1] Boxplot (ou diagrama de caixas) é uma forma gráfica de representar dados numéricos através de cinco quantidades: a menor observação, quartil inferior, mediana, quartil superior e maior observação.

Tabela 6 – Busca por variáveis ótimas para as redes FTDNN, Elman e NARX-MISO com 1 camada oculta.

laço	max. épocas	tx. aprendizagem	dimensão de imersão	atraso de imersão	neurônios 1 camada	dimensão (d_y) NARX-MISO
FTDNN						
1	80	0,05	13	4	6	-
2	70	0,05	14	4	4	-
3	90	0,05	14	4	4	-
4	90	0,05	14	4	4	-
5	90	0,05	14	4	4	-
Elman						
1	20	0,05	14	4	6	-
2	20	0,05	14	4	8	-
3	20	0,05	14	4	8	-
4	20	0,05	14	4	8	-
NARX-MISO						
1	20	0,05	10	6	10	9
2	40	0,05	10	6	8	9
3	40	0,05	10	6	8	9
4	40	0,05	10	6	8	9



(a) (b) (c)

Figura 41 – Números de neurônios na camada oculta. (a) FTDNN, (b) Elman, (c) NARX-MISO.

Pode-se observar primeiramente com estes resultados a superioridade das redes neurais em relação ao modelo linear nesta tarefa de predição. Outro ponto a se destacar é o bom desempenho da rede FTDNN com apenas uma camada oculta, por possuir resultado superior, tanto em relação a FTDNN com duas camadas ocultas, como sobre a rede recorrente de Elman. Por sua vez, a rede NARX-MISO possui resultado melhor tanto em média como em variância sobre os outros modelos testados, demonstrando ser computacionalmente mais poderosa no mecanismo de aprendizagem da dinâmica não-linear.

Finalmente, a Figura 43 apresenta o resultado da predição de 12 meses à frente para a rede NARX-MISO, correspondendo ao ano de 2007. A linha sólida indica os valores efetivamente observados de precipitação de chuvas, os retângulos verticais indicam os valores das médias históricas dos meses correspondentes de 1974 à 2007, e a linha pontilhada mostra a média dos valores previstos em 100 execuções. Os resultados são considerados muito bons

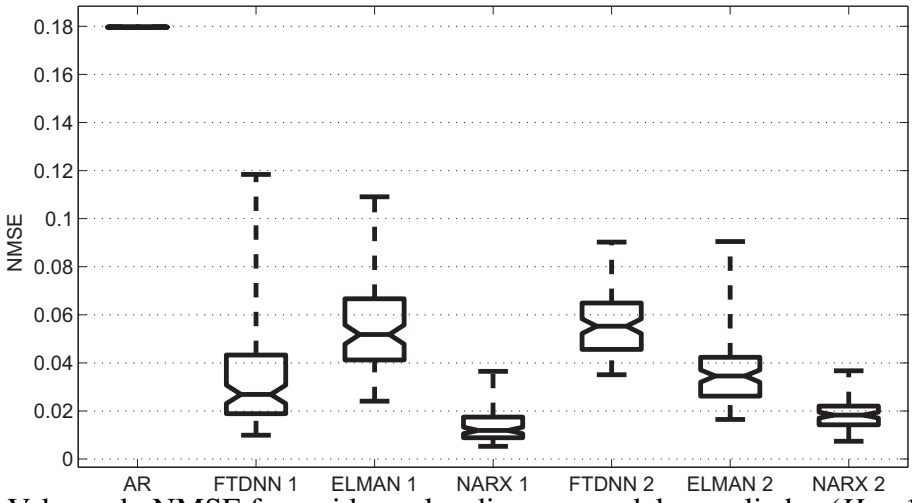


Figura 42 – Valores do NMSE fornecidos pelos diversos modelos avaliados ($H = 12$): modelo AR e redes com 1 e 2 camadas ocultas.

pelos meteorologistas da FUNCEME, sendo comparável com os resultados fornecidos pelos complexos modelos fenomenológicos (numéricos) de dinâmica do clima que realmente são usados para determinar as predições de precipitação para a sociedade. A principal vantagem do uso das redes neurais artificiais é a sua velocidade em fornecer as estimativas (poucos minutos, incluindo o treino e teste), enquanto que o modelo numérico leva várias horas.

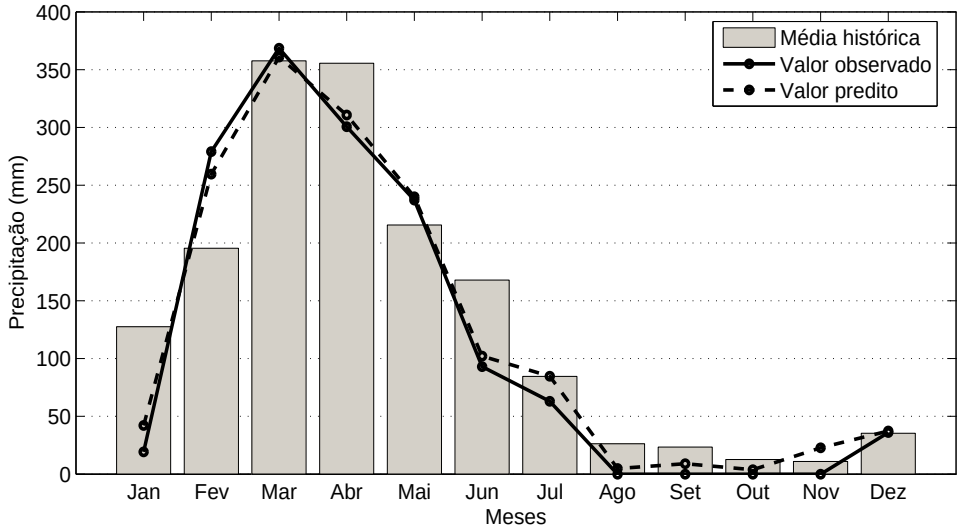


Figura 43 – Valores preditos pela rede NARX-MISO ($H = 12$), média de 100 repetições.

8.3 Resultados para a Série do Laser Caótico

No próximo teste, avaliam-se as redes neurais recorrentes com a série do laser caótico, largamente usada em estudos de *benchmark*. Da série temporal original, foram extraídas

1100 amostras, sendo que as 1000 primeiras são destinadas para o treino e as 100 últimas para o teste. As observações da série são normalizadas para o intervalo $[-1,+1]$. Os resultados da predição são calculados para a mediana de 10 rodadas de treino/teste com reinicializações aleatórias dos pesos da rede.

Tendo vários ciclos da heurística de busca pelos melhores parâmetros sido efetuada e satisfeita uma condição de parada, a metodologia retornará os melhores parâmetros dentro dos intervalos que foram testados. Com os resultados do último ciclo de busca, é possível construir um mapeamento do NMSE em função de cada combinação de parâmetros, dentro dos intervalos testados. No problema em questão, são organizadas funções de (d_E, τ), (taxa de aprendizagem, número de épocas), (número de neurônios na 1ª camada, número de neurônios na 2ª camada) e ordem do regresso de saída (d_y).

Os resultados da predição de 100 passos-adiante da rede NARX-MISO com duas camadas de neurônios ocultos são mostrados, para o caso da escolha da janela de entrada, nas Figuras 44(a) e 44(b). Na Figura 20 encontram-se os valores sugeridos pelo método de Cao e da informação mútua, onde o método de Cao indica valores para dimensão de imersão por volta de $d_E = 7$ e e para o atraso de imersão, por volta de $\tau = 2$. O melhor par encontrado na Figura 44 é $(d_E, \tau) = (8,4)$, próximos dos valores estimados pelos métodos baseados no teorema de imersão de Takens.

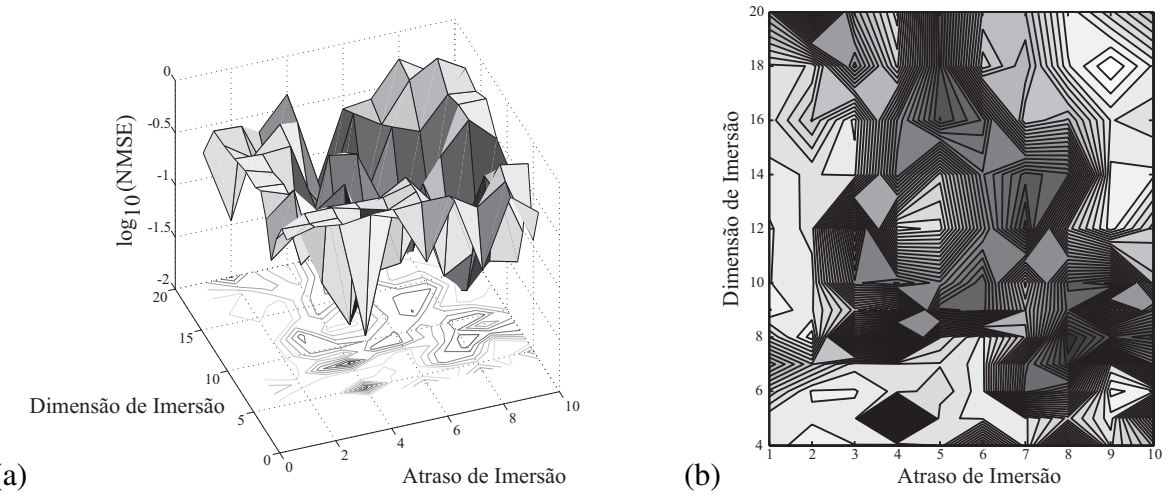


Figura 44 – Série do laser caótico. NARX-MISO com duas camadas ocultas. (a) e (b) Dimensão e atraso de imersão.

Esta mesma metodologia também é realizada para encontrar o melhor par da taxa de aprendizagem e número de épocas de treinamento dentro de uma faixa de análise. O resultado é

mostrado nas Figuras 45(a) e 45(b). Pode-se verificar um padrão de ocorrência dos mínimos do NMSE, áreas mais escuras, que ocorrem em 2500 épocas e taxa de aprendizagem de 0,05.

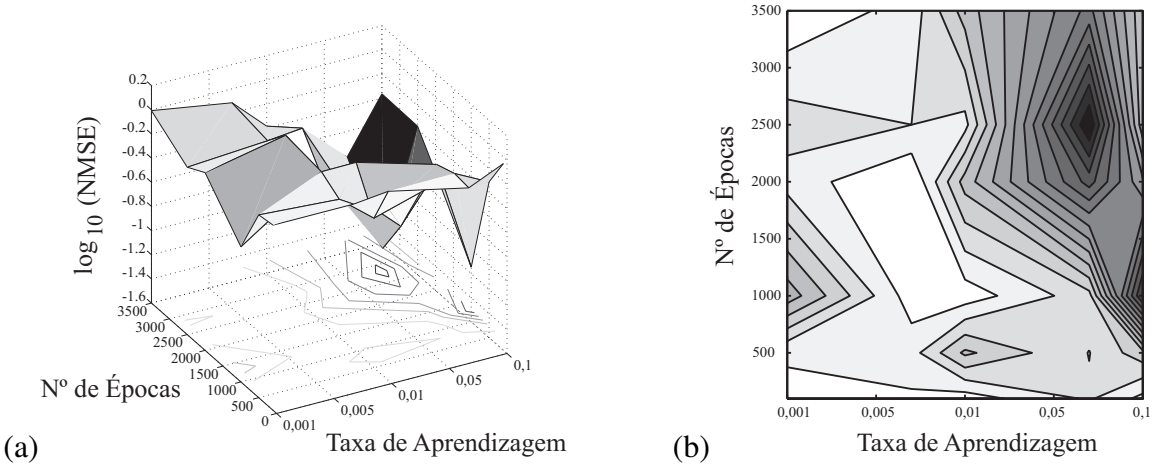


Figura 45 – Série do laser caótico. NARX-MISO com duas camadas ocultas. (a) e (b) Número de épocas de treinamento e taxa de aprendizagem.

Para o caso da escolha do número de neurônios do modelo NARX-MISO com duas camadas ocultas, é analisada a faixa de [2, 40], com os resultados gráficos vistos nas Figuras 46(a) e 46(b). Para o modelo NARX-MISO o erro mínimo é encontrado com 30 e 20 neurônios, na primeira e segunda camada oculta respectivamente. Também é verificado que os valores de 20 e 10 neurônios podem ser escolhidos para este modelo, principalmente se for levado em conta que este par permite uma redução do números de pesos da rede.

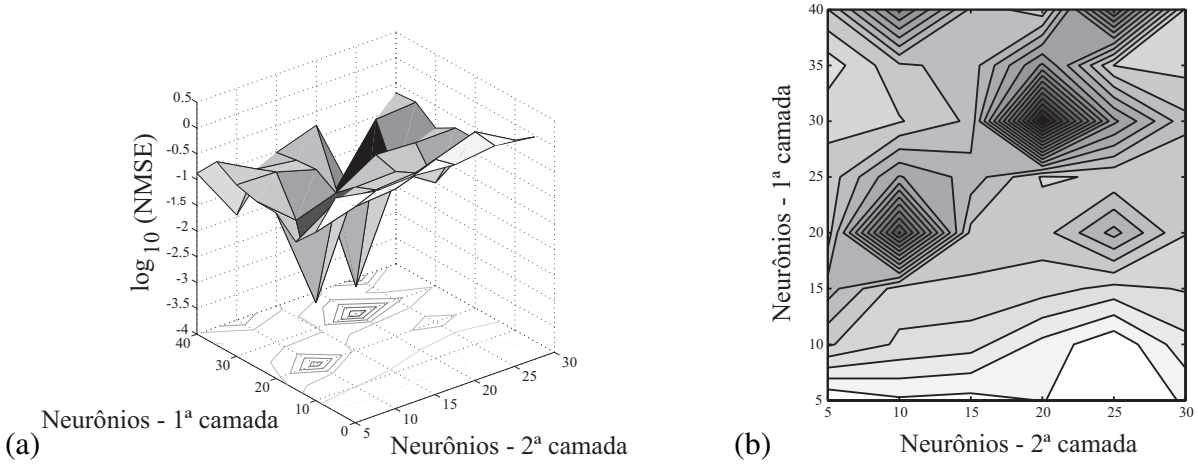


Figura 46 – Série do laser caótico. NARX-MISO com duas camadas ocultas. (a) e (b) Número de neurônios na primeira e na segunda camada oculta.

Na Figura 47, é analisada a ordem do regressor de saída da rede NARX-MISO na faixa de [1, 40]. Pode-se verificar que o melhor valor encontrado é $d_y = 40$. Embora este seja o

último dos valores testados e por ventura um maior valor pudesse ser escolhido, nos experimentos evitou-se utilizar valores maiores para d_y. Esta escolha se deve porque um valor muito alto de d_y poderia aumentar demasiadamente a entrada da rede NARX-MISO e acarretar uma rede mais custosa de ser treinada.

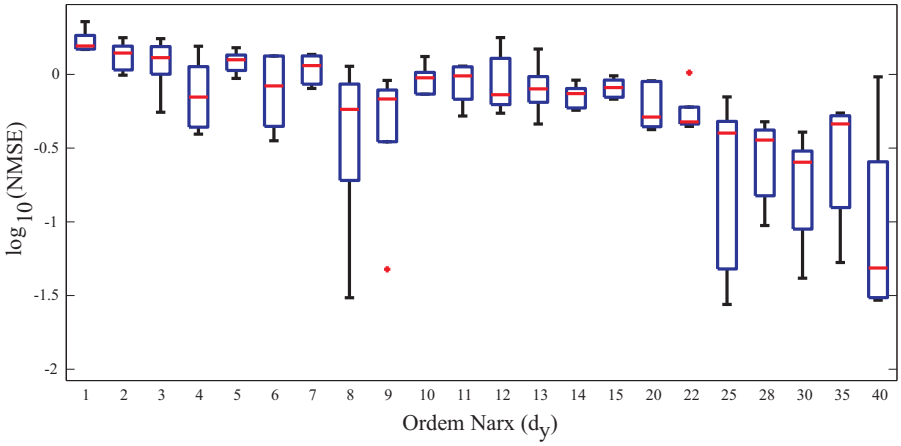


Figura 47 – Série do laser caótico. Variação da ordem do regresso de saída (d_y) da rede NARX-MISO com duas camadas ocultas.

Na Tabela 7 são listados resultados previamente publicados em Gers (2001) e Bontempi, Birattari e Bersini (1999) para a série do laser caótico, tanto para predição UPA e HPA. Nesta mesma tabela constam também os resultados obtidos no presente trabalho para as redes MLP, Elman e NARX-MISO. Na Figura 48 são mostrados apenas os resultados em ordem de magnitude do NMSE para a predição recursiva, onde pode-se observar o bom desempenho da rede NARX-MISO quando comparado com outros modelos de predição.

Tabela 7 – Resultados do erro de predição (NMSE) da série do laser caótico. Valores entre parênteses representam a variância dos resultados.

Referência	NMSE UPA	NMSE HPA
LSTM (GERS, 2001)	0,3642	0,9683
MLP (GERS, 2001)	0,0996	0,8569
Rede FIR (WAN, 1994)	0,023	0,0551
Rede sFIR (WAN, 1994)	-	0,0273
MLP (KOSKELA *et al.*, 1998)	0,0177	-
RSOM (KOSKELA *et al.*, 1998)	0,0833	-
EP-MLP (BAKKER *et al.*, 2000)	-	0,2159
(SAUER, 1994)	-	0,077
(WEIGEND; GERSHEFELD, 1994)	0,0198	0,016
(BONTEMPI; BIRATTARI; BERSINI, 1999)	-	0,029
MLP (16-14-10-1)	0,1717 (8,6 x 10^{-4})	0,7548 (0,042)
Elman (16-14-2-1)	0,0340 (1,6 x 10^{-6})	0,8479 (0,142)
NARX-MISO (8-20-10-1))	0,0339 (1.3 x 10^{-7})	0,0565 (0,0011)

Os detalhes das redes que geraram os resultados da Tabela 7 são apresentados a seguir. Wan (1994) reportou o melhor resultado submetido para a competição promovida pelo instituto Santa Fé. Ele usou uma rede FIR-MLP com 25 unidades de entradas e 12 neurônios ocultos. Esta rede modela cada neurônio como um filtro FIR, ou seja, cada neurônio possui memória. A principal consequência desta abordagem é que a rede torna-se difícil de sintonizar e extremamente instável. Para predição recursiva, o autor teve que utilizar observações suavisadas dando origem à rede *smoothed* FIR-MLP. Koskela *et al.* (1998) compara a rede SOM recorrente (RSOM) com a rede MLP treinada com o algoritmo Levenberg-Marquardt (LM) usando em ambas uma janela de entrada com $d_E = 9$. Bakker *et al.* (2000) usa uma mistura de valores preditos e valores observados como entrada, a fim de diminuir a propagação de erro devido à realimentação. Além disso, este autor usa PCA (Análise de Componentes Principais) para reduzir a dimensionalidade da entrada dos 40 mais recentes valores para 16 componentes principais. Uma rede MLP foi usada para predição utilizando o algoritmo *backpropagation through time* (BPTT) para treinamento. Na etapa de teste recursivo, apenas os valores preditos são realimentados para a entrada.

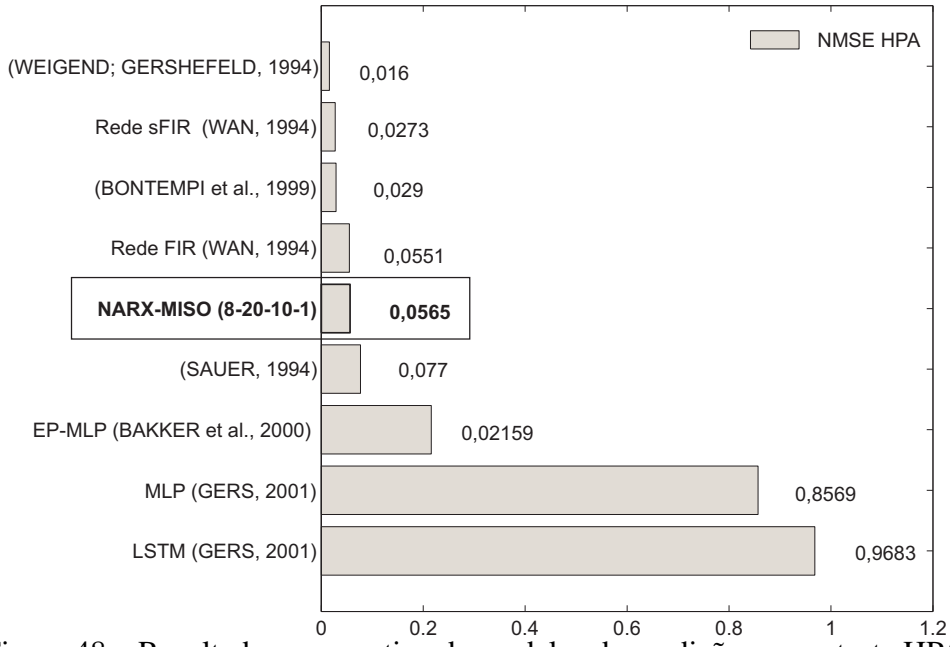


Figura 48 – Resultado comparativo de modelos de predição para o teste HPA.

O melhor resultado para a predição recursiva relatado em Gers (2001) foi o obtido por Weigend e Gershefeld (1994) utilizando uma rede MLP com 25 unidades de entrada, 12 neurônios ocultos e com uma saída adicional para estimar o erro de predição. Bontempi, Birattari e Bersini (1999) utiliza métodos de aprendizagem local para a predição recursiva. Finalmente,

Gers (2001) utiliza a rede recorrente *Long Short-Term Memory* (LSTM), que possui blocos de memória para guardar informação temporal. Deve-se destacar que o bom desempenho da rede NARX-MISO para predição recursiva foi obtido usando o algoritmo de treinamento *backpropagation* padrão, enquanto todas as outras redes cujos resultados estão mostrados na Tabela 7 usam métodos de treinamentos mais sofisticados (e.g. LM e BPTT).

A Figura 49 apresenta a ilustração da predição da rede NARX-MISO para 500 passos adiante. A linha sólida representa os 500 valores da amplitude estimados recursivamente e a linha tracejada são os valores exatos da série. É importante salientar que a situação crítica do laser caótico ocorre por volta do instante de tempo 60, quando ocorrem colapsos da intensidade do laser repentinamente (passam de um valor alto para um valor baixo), para então começar uma recuperação gradual da mesma. Assim, pode ser observado que o modelo NARX-MISO é capaz de reproduzir as dinâmicas do laser caótico muito fielmente, inclusive no ponto crítico, após o colapso.

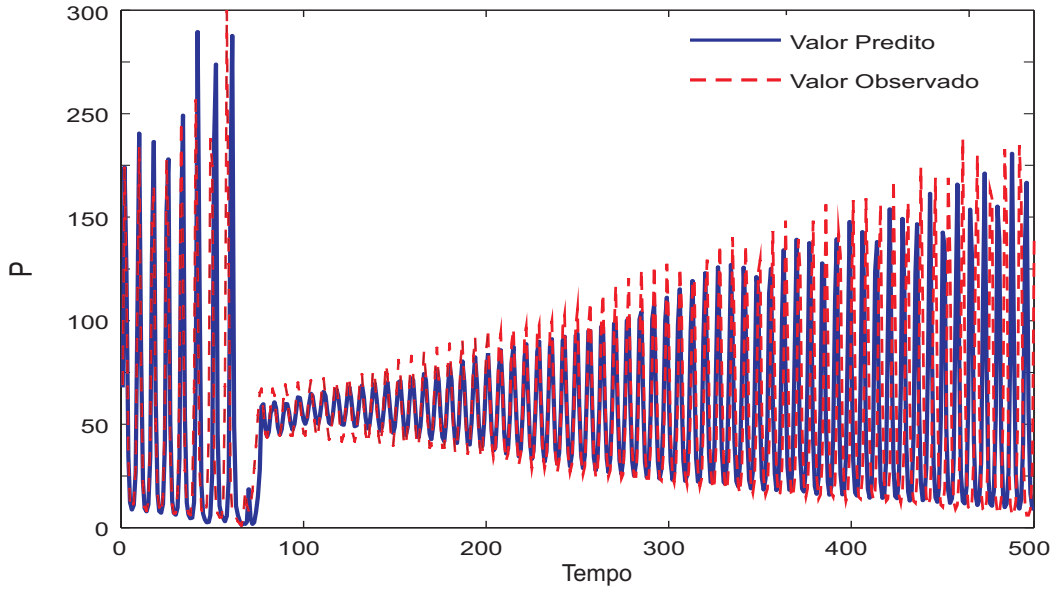


Figura 49 – Predição HPA para a série do laser caótico utilizando a rede NARX-MISO.

Embora o modelo NARX-MISO não tenha apresentado o menor erro de predição para o horizonte de 100 passos, não implica que ele seja “pior” que os outros modelos comparados. Na Figura 49, observa-se que a rede NARX-MISO é capaz de reproduzir (modelar) as dinâmicas do laser caótico para horizontes mais longos, melhor do que, por exemplo, o modelo de Wan (1994).

8.4 Resultados para a Rede NARX-MIMO

O objetivo deste experimento é avaliar o desempenho da rede NARX-MIMO, introduzida na Seção 5.3, em tarefa de predição recursiva. Tenta-se verificar se o aumento do número de neurônios na camada de saída, um para cada horizonte de predição desejado, aumenta a capacidade preditiva da rede NARX.

Duas séries caóticas clássicas (Hénon e Mackey-Glass) são utilizadas pra avaliar o desempenho da rede NARX-MIMO. Para o sistema caótico de Hénon foram adotados os valores a =1,4 e b =0,3, para produzir uma dinâmica caótica. As observações da série de Mackey-Glass são utilizados os seguintes valores: α = 0,2, β = - 0,1 e Δ = 17.

Para a série de Hénon, são gerados 200 valores, dos quais 150 são utilizados para treino e o restante para teste. Já para a série de Mackey-Glass, são utilizadas 500 amostras, sendo que, 200 são para treino e o restante para teste. Todas as séries são reescalonadas para a faixa de $[-1,+1]$.

A heurística discutida na Seção 4.4 é adotada para seleção dos melhores parâmetros da rede NARX-MISO. São otimizados os valores da dimensão de imersão, atraso de imersão, ordem do regressor de saída (d_y), número de épocas de treinamento, taxa de aprendizagem e número de neurônios em cada camada oculta. Para a série de Hénon é encontrada a rede NARX-MISO(2 + 2,20,18,1) e para a série de Mackey-Glass é utilizada a rede NARX-MISO(3 + 11,30,12,1), já com os valores dos parâmetros otimizados inseridos nas denotações.

Os melhores parâmetros encontrados pela heurística citada foram fixados e testou-se a rede NARX-MIMO variando o número de saídas da rede. Assim, cada rodada de treino/teste de cada modelo MIMO é repetido $K = 10$ vezes e o erro de predição é calculado via NMSE. Vale observar que se a rede NARX-MIMO é definida com apenas uma saída, esta rede é reduzida a uma rede NARX-MISO.

A Figura 50 apresenta, por meio de *boxplots*, o resultado produzido pela variação do número de saídas do modelo NARX-MIMO para a série de Hénon. Os erros de predição estão em escala logarítmica de base 10 para suavizar o NMSE. Pode-se observar que, para duas saídas, a rede NARX-MIMO obteve melhor resultado que para o modelo com apenas um nó da camada de saída, isto é, uma rede NARX-MISO.

A Figura 51 apresenta o resultado produzido pela variação do número de saídas do modelo NARX-MIMO para a série de Mackey-Glass. Analisando os erros de predição deste experimento, pode-se observar que para duas ou três saídas são gerados os melhores resultados

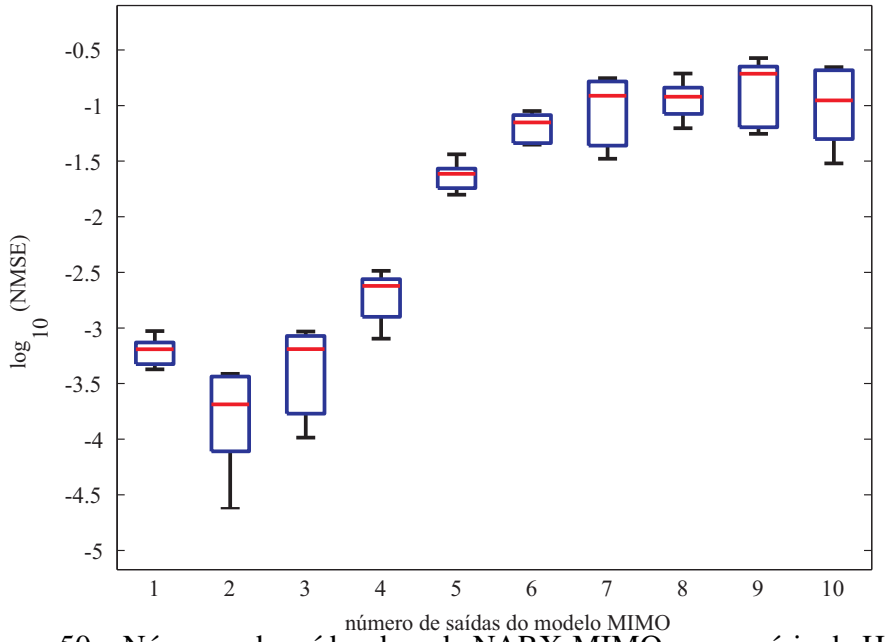


Figura 50 – Números de saídas da rede NARX-MIMO para a série de Hénon.

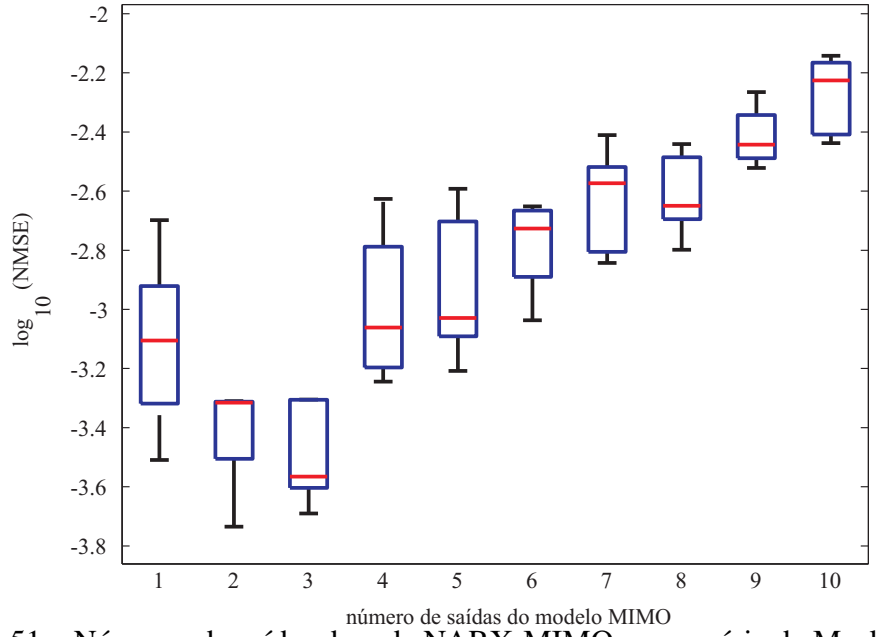


Figura 51 – Números de saídas da rede NARX-MIMO para a série de Mackey-Glass.

na tarefa de predição da rede NARX-MIMO.

8.5 Conclusão

Nesta tese são avaliados os desempenhos dos modelos neurais recorrentes e a rede NARX-MISO na tarefa de predição de recursiva. É utilizado para avaliar estas redes a série de precipitação de chuva e a série do laser caótico. Os resultados obtidos são comparados em um contexto mais amplo com aqueles obtidos por diversas arquiteturas para a mesma tarefa.

Uma análise dos resultados obtidos mostrou que os modelos NARX-MISO apresentam bom desempenho quando comparados com as redes FTDNN e Elman.

Como primeira conclusão deste capítulo, pode-se afirmar que as redes neurais NARX-MISO têm um melhor desempenho na predição UPA, predição HPA e na tarefa de modelagem do que as redes FTDNN e Elman. Em particular, além de possui desempenho superior, a rede NARX-MISO é mais flexível na escolha de alguns parâmetros da rede neural.

Este melhor desempenho das redes NARX-MISO pode ser explicado pela sua capacidade de extrair as memórias de curto e longo prazo. Em especial, o caso do modelo de treinamento série paralelo da rede NARX-MISO obter um desempenho melhor do que todas as redes testadas, pode ser explicado por dois motivos. Primeiro, o regressor de saída durante o treino é composto de amostras exatas da série temporal de interesse e não de valores estimados, deixando este modelo mais preciso. Em segundo, a rede NARX-MISO tem puramente uma arquitetura *feedforward*, e pode ser treinada pelo algoritmo *backpropagation*.

A heurística adotada para encontrar os melhores modelos confirmou, assim como no capítulo anterior, os valores sugeridos para a dimensão e atraso de imersão, estimados pelo método de Cao e informação mútua, respectivamente. Para os outros parâmetros, observa-se que nenhuma das heurísticas citadas na Seção 4.4 sugeriu valores condizentes com os que são determinados aqui. Por fim, deve-se observar que o método de busca demonstrou ser necessário um número pequeno de neurônios na camada oculta da rede NARX-MISO, além dessa escolha ser mais flexível em relação as outras duas redes testadas, FTDNN e Elman.

Por fim, a abordagem NARX-MIMO gerou erros de predição recursiva menores que os da rede NARX-MISO. Pode-se explicar esse melhor desempenho pela forma como o método calcula os valores futuros da predição. Isto faz com que a rede NARX-MIMO reduza o acúmulo dos erros de predição, teoricamente de forma mais eficiente que na rede NARX-MISO, que utiliza o método de predição recursiva.

No próximo capítulo, são avaliados os resultado das técnicas baseadas em projeções aleatórias. O objetivo é apresentar os resultados tanto para a rede ESN, como também para a rede ELM.

9 RESULTADOS PARA AS VARIANTES DA REDE NARX BASEADAS EM PROJEÇÕES ALEATÓRIAS

9.1 Introdução

Este capítulo tem como propósito avaliar o desempenho das redes NARX-ESN e NARX-ELM na tarefa de predição recursiva, usando as séries de Hénon, de Mackey-Glass e do laser caótico. O objetivo deste estudo é ressaltar as diferenças no projeto de tais arquiteturas neurais recorrentes, a fim de oferecer subsídios ao usuário no momento da escolha da arquitetura mais adequada à tarefa de interesse.

9.2 Resultados para a Rede NARX-ESN

Esta seção é dedicada a análise dos resultados da rede NARX-ESN e de suas variantes.

9.2.1 Resultados para a Série de Hénon

O objetivo deste primeiro experimento é avaliar as variações da rede NARX-ESN, entrada da rede e regressor de saída, conforme descritas na Seção 6.3.1. Também é avaliada a influência dos parâmetros da rede NARX-ESN, conforme discutido na Seção 6.3.2. As arquiteturas da redes NARX-ESN utilizadas possuem um único neurônio na camada de saída ($L = 1$) que fornece a predição do próximo valor da série temporal.

Cada um destes modelos tem os parâmetros encontrados pela metodologia apresentada na Seção 4.4, sendo que cada rodada de treino/teste de cada modelo é repetido $K = 20$ vezes. A Figura 52 apresenta o resultado para a predição recursiva com 50 passos-adiante de todos os 8 modelos avaliados. Pode-se observar que o uso da janela de Takens como informação da unidade de entrada da rede não traz benefícios, visto que nenhuma combinação da dimensão e atraso de imersão gera resultados significativamente melhores na tarefa de predição. O modelo que gera o menor erro na predição de múltiplos-passos-adiante é o que utiliza informação da entrada e da saída para a camada de saída (ESN(1, d_y ǀ 1, d_y)). Este modelo não utiliza a janela de Takens e sim um valor fixo como unidade de entrada, ficando a unidade de saída projetada para os neurônios do reservatório como a única fonte geradora de informação para a rede (ver Tabela 4).

Para esta melhor configuração, são geradas algumas figuras que ilustram os resultados

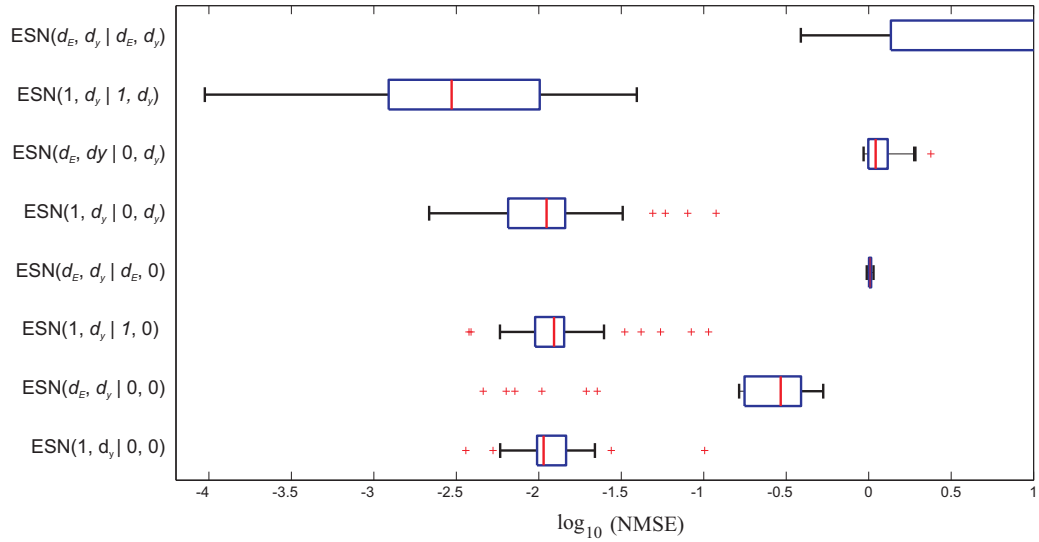


Figura 52 – Valores do NMSE em logaritmo fornecidos pelos diversos modelos avaliados da rede NARX-ESN com diferentes configurações (teste recursivo, $H = 50$) da série de Hénon).

da variação dos parâmetros utilizados na rede NARX-ESN. A Figura 53 apresenta o desempenho da variação da ordem do regressor de saída e do número de neurônios do reservatório. A variação do número de neurônios do reservatório não gera muita diferença no desempenho da predição HPA para valores superiores a 60 neurônios. O menor NMSE é obtido para 40 neurônios. Já o teste com a ordem do regressor de saída, isto é, o número de valores atrasados realimentados da saída para a entrada, indica $d_y = 1$ como sendo o que gera o menor erro na predição da série de Hénon.

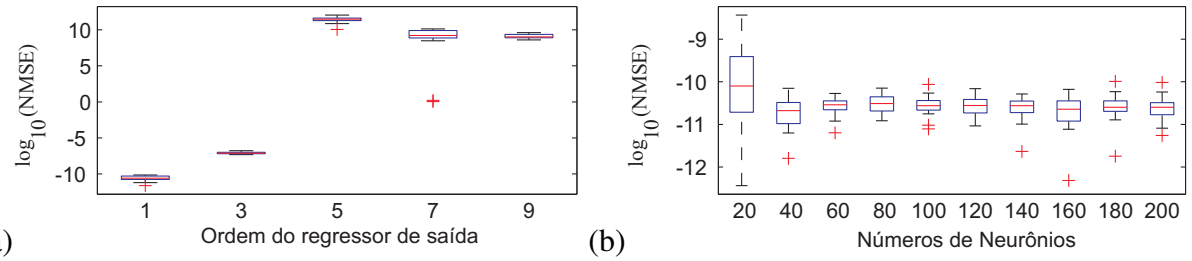
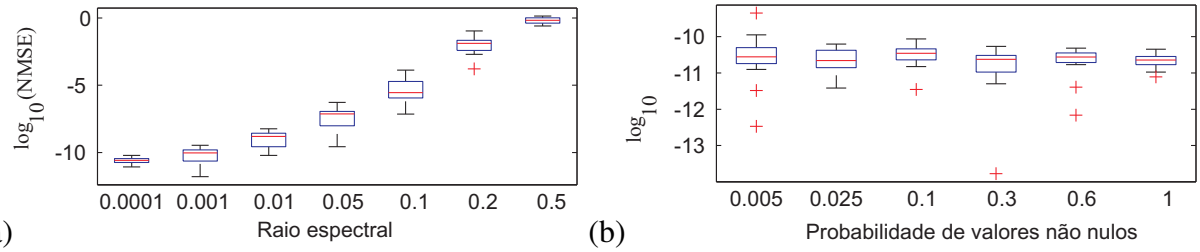


Figura 53 – Parâmetros para a rede ESN(1, d_y | 1, d_y) com série de Hénon: (a) ordem do regressor de saída; (b) números de neurônios do reservatório.

A Figura 54 mostra os resultados da variação do raio espectral do reservatório, demonstrando que valores baixos devem ser utilizados, no caso deste problema o valor $0,0001$ é escolhido. A probabilidade de valores não nulos nas unidades do reservatório, faixa de testes de $[0,5\% - 100\%]$, não é um parâmetro relevante, visto que sua variação não traz diferença na tarefa de predição da série de Hénon.

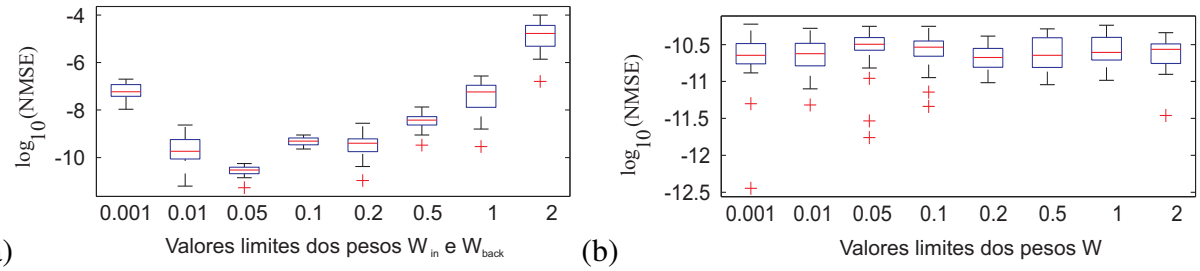
Valores para amplitude dos pesos do reservatório ($\mathbf{W}$) escolhidos dentro do intervalo



(a) (b)

Figura 54 – Parâmetros para a rede ESN(1, d_y ǀ 1, d_y) com série de Hénon: (a) raio espectral do reservatório; (b) probabilidade de valores não nulos nas unidades do reservatório.

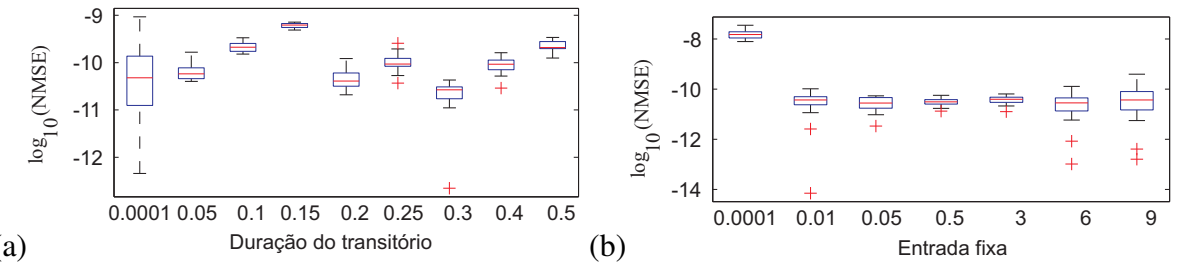
$[0,001-2]$ não gera diferenças nos resultados. Desta forma, qualquer valor dentro da faixa testada pode ser utilizado, como pode ser visto da Figura 55(a). Já para a amplitude dos pesos $\mathbf{W}^{in}$ e $\mathbf{W}^{back}$, existe uma região ótima, por volta de $0,05$, em que a rede NARX-ESN possui o melhor desempenho preditivo, como pode ser visto na Figura 55(b) .



(a) (b)

Figura 55 – Parâmetros para a rede ESN(1, d_y ǀ 1, d_y) com série de Hénon: (a) amplitude dos pesos $\mathbf{W}^{in}$ e $\mathbf{W}^{back}$; (b) amplitude dos pesos do reservatório $\mathbf{W}$.

Por fim, a Figura 56 apresenta os resultados da variação da duração do transitório e o valor da entrada fixa. A arquitetura da rede ESN(1, d_y ǀ 1, d_y) não utiliza o teorema de Takens para compor a entrada, sendo utilizado como unidade de entrada um valor fixo (ver Tabela 4). A varição entre $[0,001-9]$ indica que valores próximos de 1 (0,5 e 3 entre os valores avaliados) geram os menores erros, isto é, aqueles com menor mediana e dispersão. Para a duração do transitório, os testes indicaram que 30% dos dados devam ser descartados para a etapa de estimação dos pesos de saída, como uma forma de excluir os efeitos do período transitório inicial do reservatório.



(a) (b)

Figura 56 – Parâmetros para a rede ESN(1, d_y ǀ 1, d_y) com série de Hénon: (a) duração do transitório; (b) valor da entrada fixa.

Na Figura 57 tem-se o resultado da predição múltiplos-passos-adiante da série Hénon. Utiliza-se a rede NARX-ESN com o melhor modelo avaliado, a rede ESN(1, d_y | 1, d_y). Verifica-se que a predição consegue acompanhar os valores reais da série em quase toda totalidade do horizonte utilizado, $H = 50$.

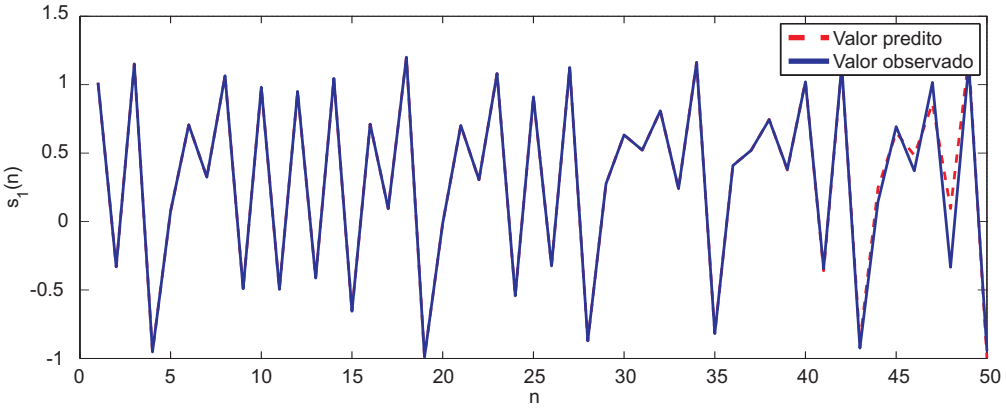


Figura 57 – Predição da série Hénon com a rede ESN(1, d_y | 1, d_y), teste recursivo, $H = 50$, com informação da saída e da entrada para a camada de saída, sem informação de Takens na entrada.

9.2.2 Resultados para a Série de Mackey-Glass

A fim de continuar a investigação do desempenho das variantes da rede NARX-ESN e das diversas possíveis arquiteturas desta rede, é avaliada a resposta da predição múltiplos-passos-adiante utilizando a série de Mackey-Glass. Todas as arquiteturas utilizadas possuem um único neurônio na camada de saída ($L = 1$),s que fornece a predição do próximo valor da série.

Utilizando as diversas configurações possíveis com as alimentações e realimentações, pode-se construir oito modelos com a rede NARX-ESN, como visto anteriormente. Cada um destes modelos tem os parâmetros encontrados pela metodologia apresentada na Seção 4.4, sendo que cada rodada de treino/teste de cada modelo é repetido $K = 20$ vezes.

Tendo sido selecionados e testados todos os modelos, o próximo objetivo é avaliar o desempenho da predição em termos do valor NMSE de todas as redes ESNs avaliadas. O *Boxplot* para os valores de $NMSE(h,l)$, obtidos para $h = 30$ e $l = 1,\ldots,20$, é mostrado na Figura 58. Esta figura ilustra os resultados para a predição recursiva de todos os 8 modelos. Pode-se observar, novamente, que o uso da janela de Takens como entrada para a rede não traz benefícios, nenhuma combinação da dimensão e atraso de imersão gerou resultados significativamente melhores na tarefa de predição. O modelo que gera o menor erro na predição múltiplos-passos-adiante é a rede ESN(1, d_y | 1, d_y), em que é utilizado a informação da entrada e a realimentação da saída

para a camada de saída (ver Tabela 4).

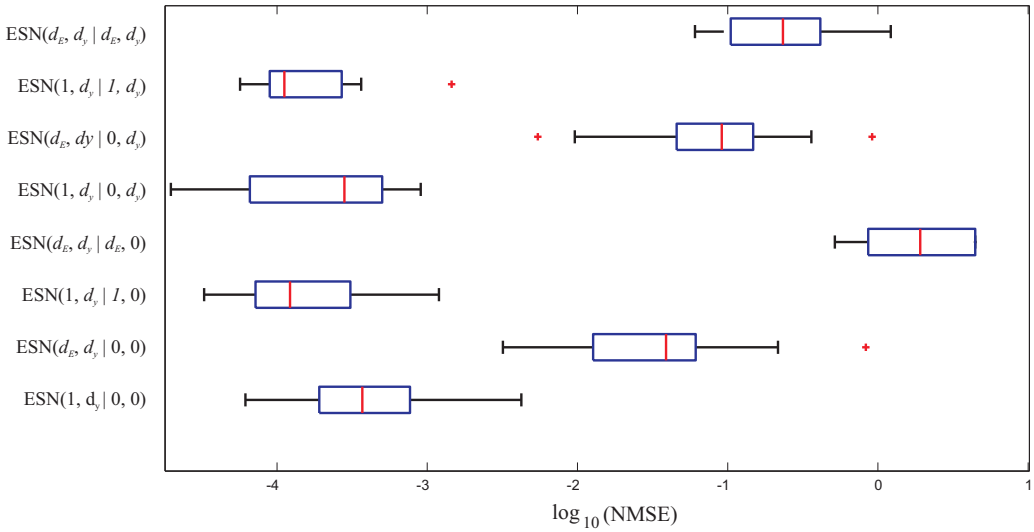


Figura 58 – Valores do NMSE em logaritmo fornecidos pelos diversos modelos avaliados da rede NARX-ESN com diferentes configurações (teste HPA, $H = 30$) da série de Mackey-Glass).

Na Figura 59, tem-se o horizonte de predição múltiplos-passos-adiante da série de Mackey-Glass utilizando a rede NARX-ESN com os melhores modelos encontrados no experimento anterior. Esta figura é gerada com base na Equação (4.9), fazendo-se variar o valor de H, tomando horizonte de predição até o instante $H = 100$. Esta figura não apresenta o desempenho dos modelos com a janela de Takens, pois estes modelos, como discutido anteriormente, possuem resultados inferiores. Desta forma, pode-se verificar novamente que o modelo utilizando informações para a camada de saída, ESN(1, d_y | 1, d_y), obtém o melhor resultado em relação aos outros modelos avaliados.

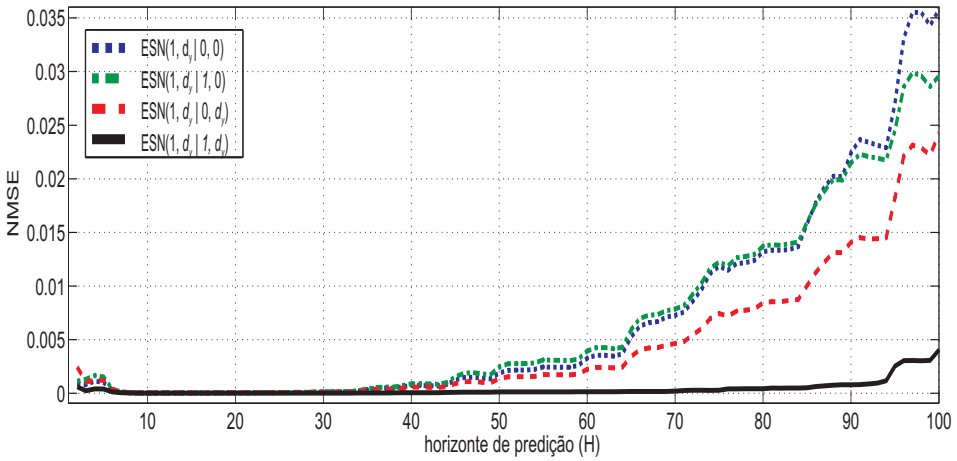


Figura 59 – Horizonte de predição para a série de Mackey-Glass com a rede NARX-ESN ($H = 100$).

Na Figura 60, tem-se o resultado da predição múltiplos-passos-adiante da série Mackey-Glass utilizando a rede NARX-ESN com o melhor modelo avaliado. Verifica-se que os valores da predição conseguem acompanhar os valores reais da série até em torno do horizonte $H = 100$. Após isso, embora a predição não seja precisa, não se pode dizer que este resultado seja errado, principalmente no ponto de vista de aprendizado da dinâmica.

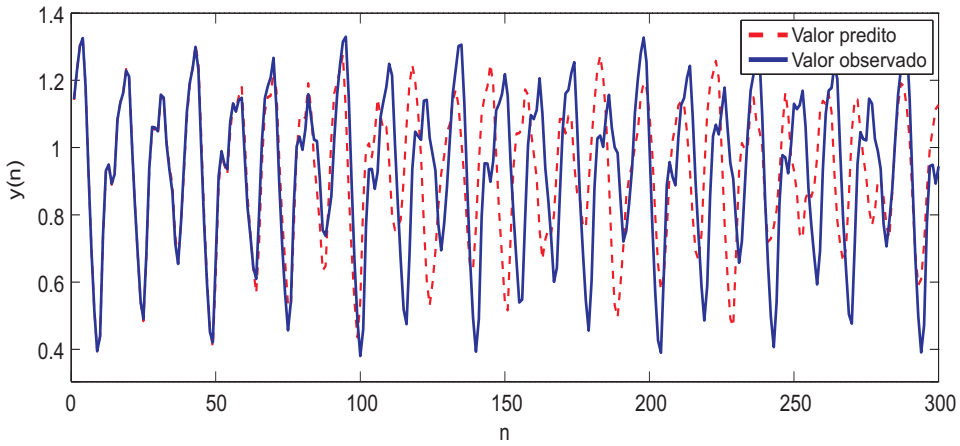


Figura 60 – Predição da série Mackey-Glass com a rede ESN(1, d_y ǀ 1, d_y), teste Recursivo, $H = 300$. Com informação da saída e da entrada para a camada de saída. Sem informação de Takens na entrada.

9.2.3 Resultados para a Série do Laser Caótico

No próximo teste, avaliam-se as redes NARX-ESN para a série do laser caótico. Esta série temporal possui 1100 amostras, sendo que as 1000 primeiras são destinadas para o treino e as 100 últimas para o teste. As observações da série são normalizadas para o intervalo $[-1,+1]$. Os resultados da predição são calculados para a mediana de 10 rodadas de treino/teste com reinicializações aleatórias dos pesos da rede.

Neste experimento foram utilizadas apenas as redes que não fazem uso da janela de Takens na unidade de entrada. Estas redes escolhidas utilizaram a heurística de busca pelos melhores parâmetros. A cada repetição, os pesos da rede são iniciados com valores aleatórios de média 0 e desvio-padrão 0,25. Por fim, para cada modelo é calculada uma estatística utilizando a mediana dos valores do NMSE obtidos pelas repetições treino/teste para cada horizonte de predição.

A Figura 61 apresenta por meio de *boxplots* os resultados produzidos pelas diversas redes NARX-ESN analisadas. As quatro redes avaliadas possuem resultados de predição de

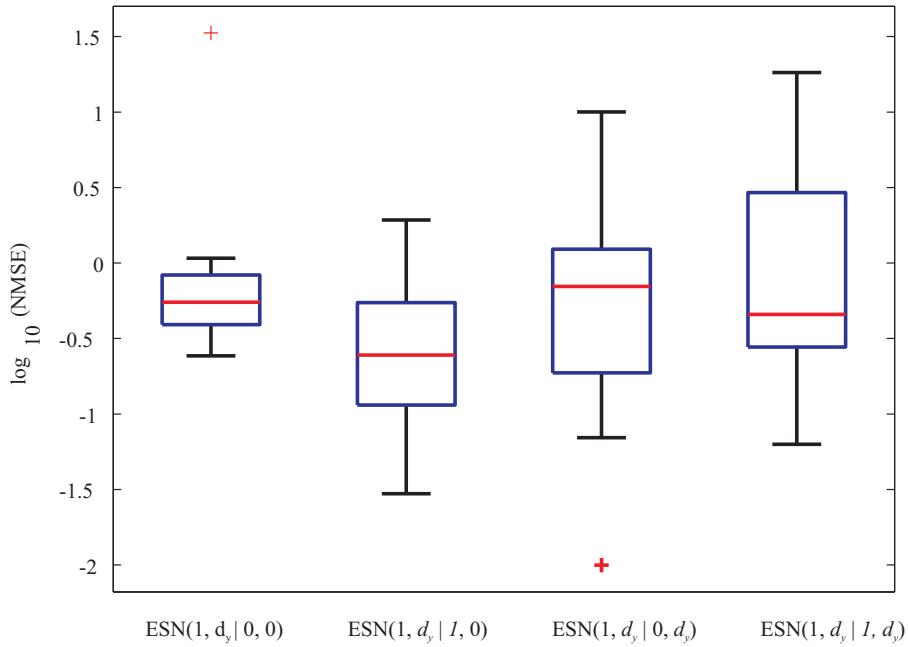


Figura 61 – Valores do NMSE em logaritmo fornecidos pelos diversos modelos avaliados da rede NARX-ESN com diferentes configurações (teste HPA, $H = 50$) da série do laser caótico.

múltiplos-passos-adiante muito equivalentes. Observa-se porém que a rede ESN(1, d_y | 1, 0) alcançou o melhor resultado dentre todas as outras redes analisadas.

Na Figura 62 tem-se a predição recursiva da série do laser caótico utilizando a rede ESN(1, d_y | 1, 0). Verifica-se que os valores da predição conseguem acompanhar os valores observados da série até em torno do horizonte $H = 40$. Vale ressaltar que a rede NARX-ESN, em nenhuma das configurações de parâmetros, consegue acompanhar o colapso da série do laser caótico.

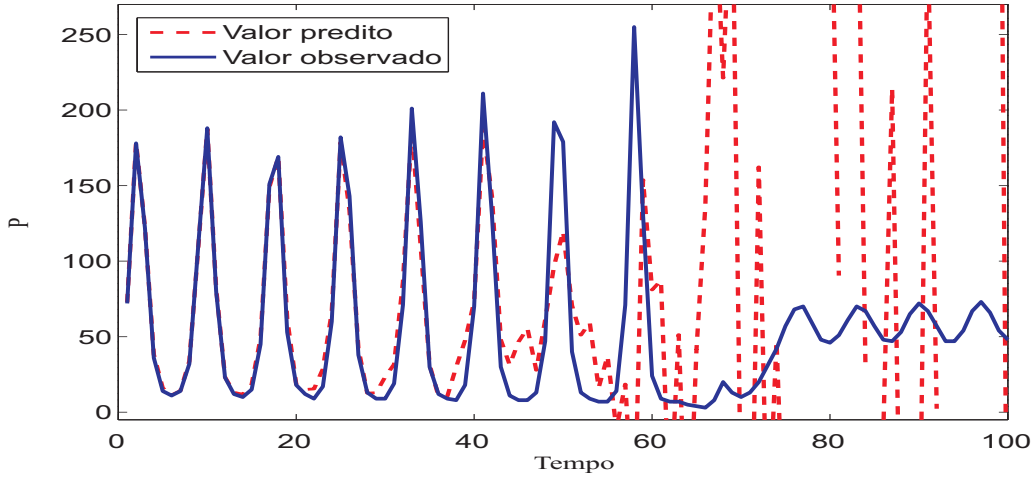


Figura 62 – Predição recursiva da série do laser caótico com a rede NARX-ESN.

9.3 Resultados para a Rede ELM

Esta seção é dedicada a análise dos resultados da rede da rede ELM. É observado o desempenho da rede ELM padrão, sem realimentações, como também o desempenho das variantes NARX-ELM, ELMAN-ELM e ELMAN/NARX-ELM.

9.3.1 Resultados para Série Hénon

A rede ELM foi a que obteve os melhores resultados na tarefa de predição recursiva da série Hénon dentre as outras redes discutidas nesta tese. A rede ELM obteve erros muito próximos da rede NARX-ESN, mas com a vantagem de possuir menos parâmetros a serem configurados.

Na Figura 63 encontram-se os resultados comparativos das quatro variações da rede ELM. Estas quatro redes se diferem basicamente no modo como a entrada de informação é inserida na rede. Pode-se destacar o bom desempenho da rede NARX-ELM, que obteve o menor NMSE comparado com as outras redes avaliadas na tarefa de predição múltiplos-passos-adiante, embora a rede ELM tenha gerado resultados muito próximos.

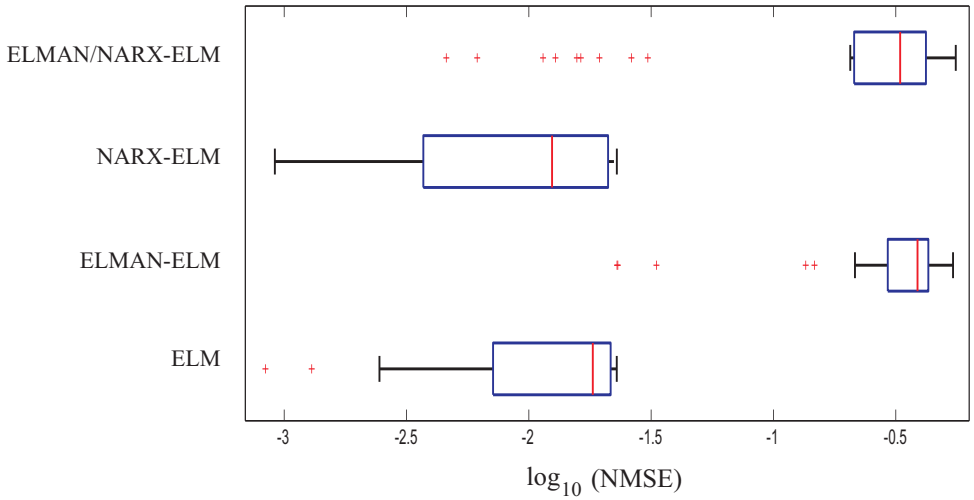


Figura 63 – Valores do NMSE em logaritmo fornecidos pelas diversas variantes da rede ELM (teste recursivo, $H = 50$) para série de Hénon): rede ELM, ELMAN-ELM, NARX-ELM, ELMAN/NARX-ELM.

Por razões de precisão estatística, cada rodada de treino/teste de cada modelo é repetido $K = 20$ vezes. Quantitativamente, para cada k-th rodada de treino/teste, os modelos são avaliados em termos do NMSE. Assim, para encontrar a melhor configuração do modelo, é utilizada a metodologia apresentada na Seção 4.4 e os resultados com os parâmetros escolhidos

para cada rede podem ser visualizados na Tabela 8. Verifica-se que a rede NARX-ELM, além de possuir o menor NMSE na predição múltiplos-passos-adiante, necessita de um número menor de neurônios na camada oculta. É importante destacar que este parâmetro é o mais importante da rede ELM, pois é diretamente ligado ao tamanho da rede e desta forma influencia o tempo de treinamento.

Tabela 8 – Variáveis ótimas para com as redes ELM, ELMAN-ELM, NARX-ELM e ELMAN/NARX-ELM, teste recursivo.

dimensão de imersão	atraso de imersão	número de neurônios	variância dos pesos	dimensão d_y	mediana do NMSE
ELM					
2	1	150	0,01	-	0,0183
ELMAN-ELM					
1	1	110	0,01	-	0,3881
NARX-ELM					
2	1	50	0,01	3	0,0124
ELMAN/NARX-ELM					
2	3	170	0,001	5	0,3296

Para a melhor configuração da rede NARX-ELM, são geradas algumas figuras que apresentam os resultados da variação dos parâmetros. A Figura 64 mostra os resultados da variação da dimensão e atraso de imersão, demonstrando que valores baixos devem ser utilizados para a dimensão de imersão, sendo neste caso, $d_E < 2$ o que leva aos menores erro de predição. A escolha do atraso de imersão não gerou grandes modificações do desempenho da rede, tal como a dimensão de imersão. O melhor par encontrado para a rede NARX-ELM e para as outras redes (ver Tabela 8) é $(d_E, \tau) = (2, 1)$, confirmando assim os valores sugeridos na Seção 4.2.1 para a dimensão e atraso de imersão da série caótica de Hénon.

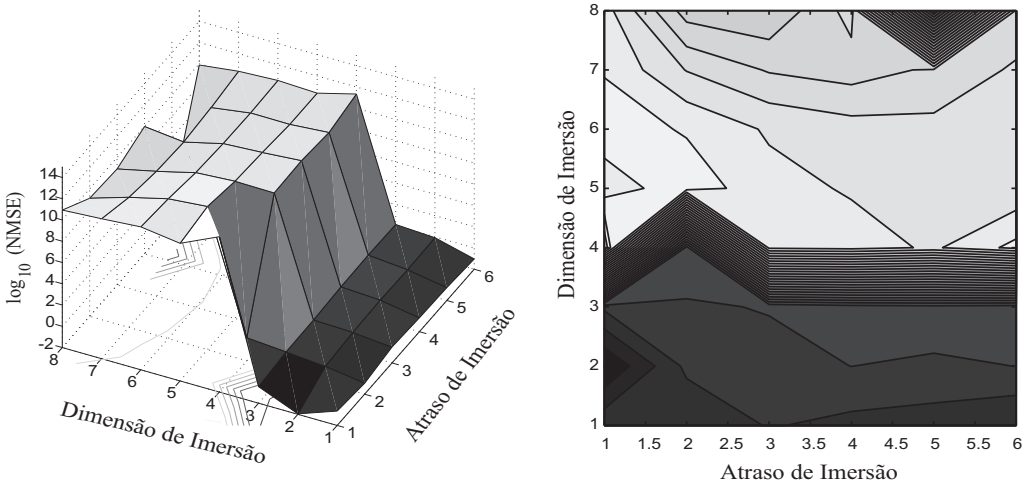


Figura 64 – Variação da dimensão e atraso de imersão para a rede NARX-ELM.

Para o caso da escolha do número de neurônios e amplitude dos pesos da camada oculta da rede NARX-ELM é analisada a Figura 65. É verificado uma região de parâmetros ótimos, região mais escura da figura, indicando o valor de $\sigma^2 = 0,01$ como parâmetro para a inicialização dos pesos. Visualmente não é possível determinar o melhor número de neurônios, mas como dito anteriormente, este valor ficou próximo de 50 neurônios para a rede NARX-ELM.

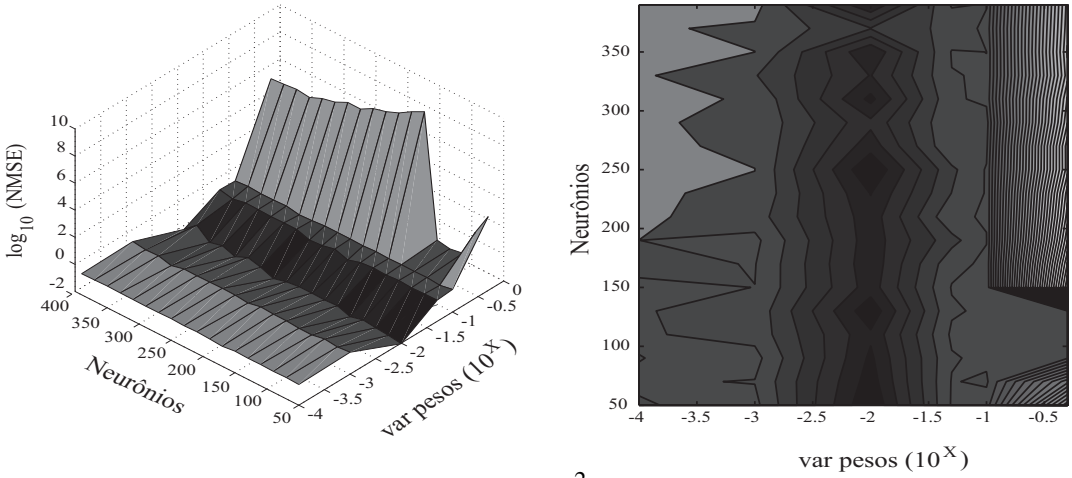


Figura 65 – Variação do número de neurônios e σ^2 dos pesos aleatórios da rede NARX-ELM.

A Figura 66 mostra os resultados da variação da ordem do regressor de saída da rede NARX-ELM. Pode-se verificar que a escolha de valores entre 1 e 3 para d_y geram os menores erros, além de possuírem menores dispersões estatísticas nos resultados.

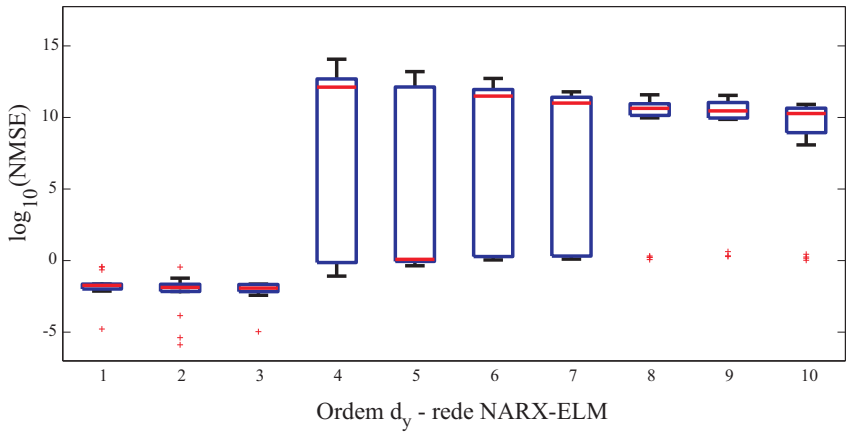


Figura 66 – Variação da ordem do contexto (d_y) para a rede NARX-ELM.

Na Figura 67, observa-se o resultado da predição múltiplos-passos-adiante da série Hénon, utilizando a rede NARX-ELM. Verifica-se que os valores da predição conseguem acompanhar os valores reais da série em quase toda totalidade do horizonte utilizado, $H = 50$.

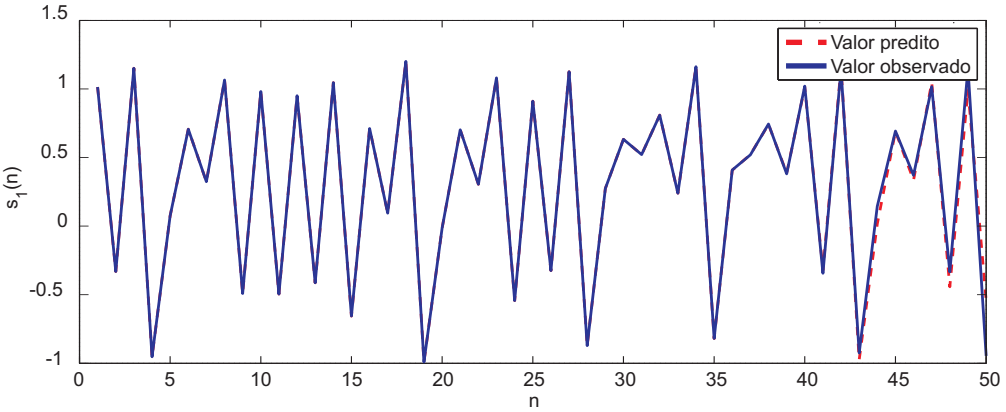


Figura 67 – Predição da série Hénon com a rede NARX-ELM, teste recursivo, $H = 50$.

9.3.2 Resultados para a Série Mackey-Glass

A rede ELM, assim como na predição para série Hénon, é a que gera os melhores resultados na tarefa de predição recursiva para a série Mackey-Glass, dentre as arquiteturas avaliadas. Novamente, vale ressaltar que, esta rede neural obteve erros muito próximos da rede NARX-ESN, mas com a vantagem de possuir menos parâmetros a serem definidos.

Tendo sido selecionados os melhores parâmetros e testados todos os modelos, o próximo objetivo é avaliar o desempenho da predição em termos do valor NMSE de todas as redes ELM avaliadas. Na Figura 68 encontra-se o *Boxplot* comparativo das quatro variantes da rede ELM, sendo que cada rodada de treino/teste de cada modelo é repetido $K = 20$ vezes. Estas quatro redes se diferem basicamente no modo como a entrada de informação é inserida na rede. Entre os preditores baseados na rede ELM, pode-se destacar o resultado da rede NARX-ELM, que obteve o menor valor do NMSE na tarefa de predição múltiplos-passos-adiante.

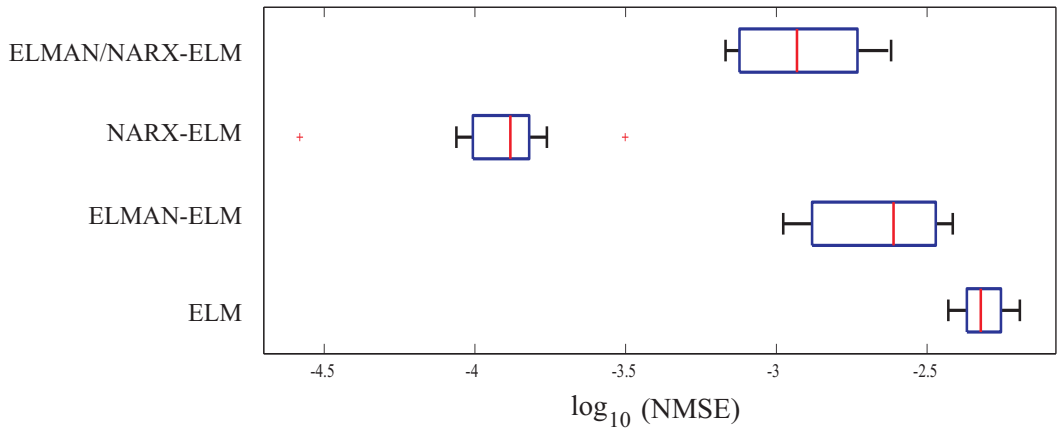


Figura 68 – Valores do NMSE em logaritmo fornecidos pelas variantes da rede ELM (teste recursivo, $H = 30$) para a série de Mackey-Glass): rede ELM, ELMAN-ELM, NARX-ELM, ELMAN/NARX-ELM.

O resultado da série Mackey-Glass com a rede ELM encontra-se na Tabela 9, onde possui os parâmetros e os valores dos NMSE para cada modelo analisado. Observa-se o desempenho superior da rede NARX-ELM em relação às outras redes, tanto em relação ao NMSE como também em relação ao número de neurônios na camada oculta. A rede ELM necessitou de um maior número de neurônios na camada oculta.

Tabela 9 – Variáveis ótimas para com as rede ELM, ELMAN-ELM, NARX-ELM e ELMAN/NARX-ELM, teste recursivo.

dimensão de imersão	atraso de imersão	número de neurônios	variância dos pesos	dimensão (d_y)	mediana do NMSE
ELM					
2	4	570	0,1	-	0,0048
ELM-Elman					
4	1	210	0,0001	-	0,0025
NARX-ELM					
4	1	130	0,01	5	1,312e-004
ELMAN/NARX-ELM					
1	10	290	0,001	4	0,0012

A Figura 69 apresenta o horizonte de predição múltiplos-passos-adiante para a série de Mackey-Glass utilizando as redes ELMs propostas. Esta figura é gerada com base na Equação 4.9, variando o horizonte de predição até o instante $H = 100$. Este resultado comprova mais uma vez a superioridade da rede NARX-ELM. A rede ELMAN/NARX-ELM também obteve resultado em destaque, possivelmente decorrente da utilização da informação da rede NARX, já que a rede ELMAN-ELM não apresentou uma boa predição.

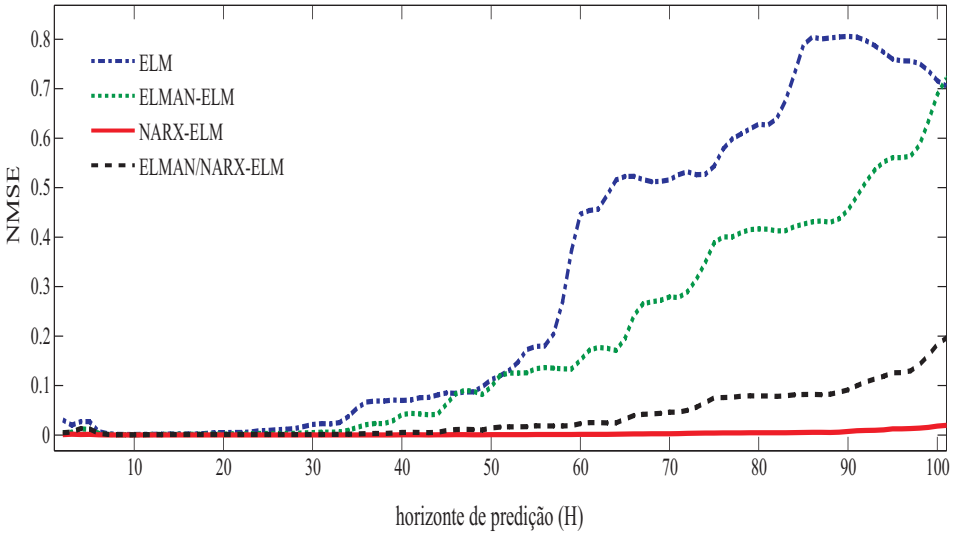


Figura 69 – Horizonte de predição para a série Mackey-Glass com a rede NARX-ELM, ($H = 100$).

A Figura 70 mostra o resultado da predição múltiplos-passos-adiante da série de Mackey-Glass utilizando a rede NARX-ELM, a partir do melhor modelo avaliado. Verifica-se que os valores da predição conseguem acompanhar os valores reais da série até em torno do horizonte $H = 120$.

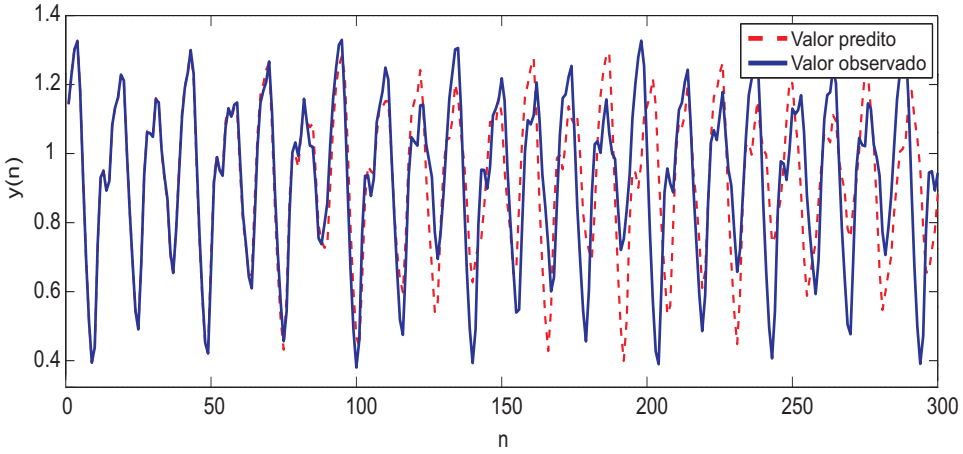


Figura 70 – Predição para a série de Mackey-Glass com a rede NARX-ELM, teste recursivo, $H = 300$.

9.3.3 Resultados para a Série do Laser Caótico

No próximo teste, avaliam-se as redes ELM para a série do laser caótico. Esta série temporal possui 1100 amostras, sendo que as 1000 primeiras são destinadas para o treino e as 100 últimas para o teste. As observações da série são normalizadas para o intervalo $[-1,+1]$. Os resultados da predição são calculados para a mediana de 10 rodadas de treino/teste com reinicializações aleatórias dos pesos da rede.

Neste experimento as redes escolhidas utilizaram a heurística de busca pelos melhores parâmetros. A cada repetição, os pesos da rede são iniciados com valores aleatórios de média 0 e desvio-padrão 0,25. Por fim, para cada modelo é construído uma estatística utilizando a mediana dos valores do NMSE obtidos pelas repetições treino/teste para cada horizonte de predição.

A Figura 71 apresenta por meio de *boxplots* os resultados produzidos pelas diversas redes ELM analisadas. A rede NARX-ELM e NARX/Elman-ELM alcançaram os melhores resultados, dentre as redes analisadas.

Na Figura 72 observa-se a predição recursiva da série do laser caótico utilizando a rede NARX-ELM. Verifica-se que os valores da predição conseguem acompanhar os valores observados da série até em torno do horizonte $H = 50$. Mas a rede ELM, assim como a rede

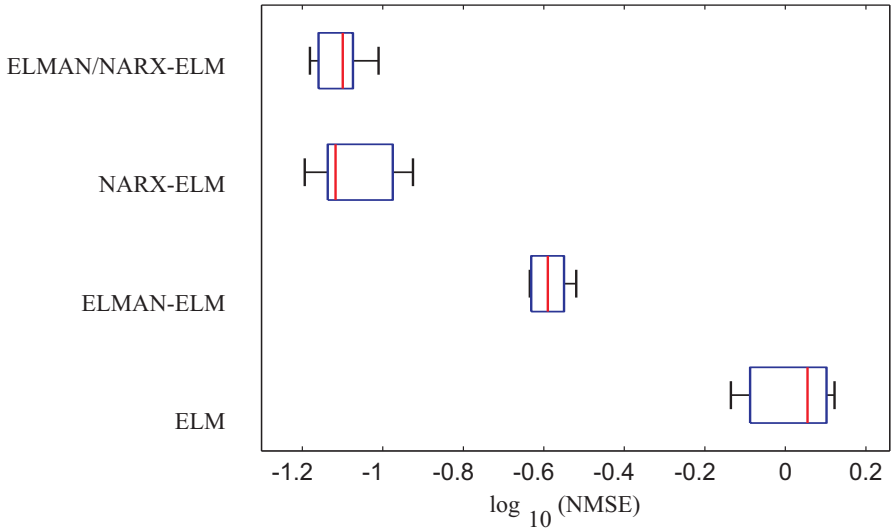


Figura 71 – Valores do NMSE em logaritmo fornecidos pelas variantes da rede ELM (teste recursivo, $H = 50$) para a série do laser caótico): rede ELM, ELMAN-ELM, NARX-ELM, ELMAN/NARX-ELM.

NARX-ESN, em nenhuma das configurações de parâmetros, consegue acompanhar o colapso da série do laser caótico.

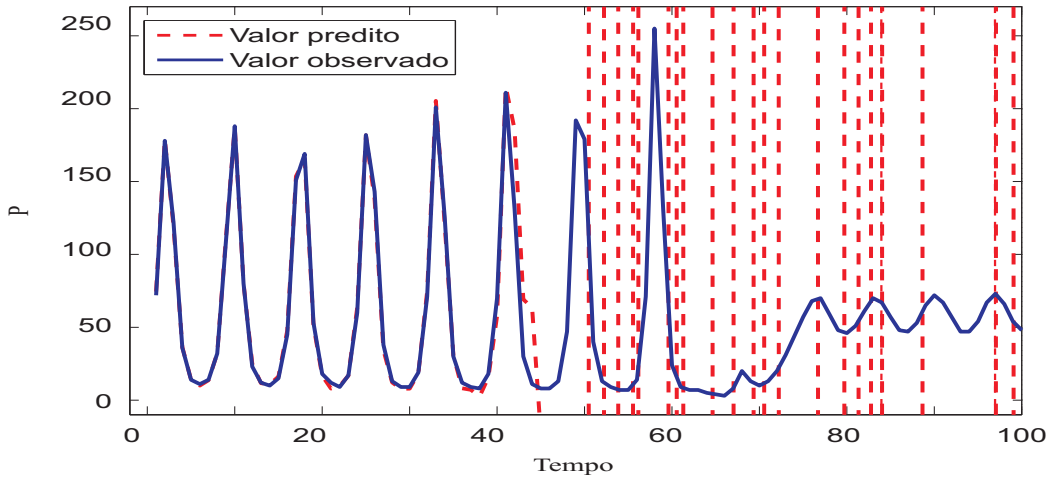


Figura 72 – Predição recursiva da série do laser caótico com a rede NARX-ELM

9.4 Conclusão

Este capítulo apresentou os resultados das arquiteturas de redes neurais baseadas em projeções aleatórias (redes NARX-ESN e rede ELM). Foram avaliadas as variantes tanto da rede NARX-ESN como da rede NARX-ELM. Para a rede NARX-ESN foram analisados os laços de alimentação e realimentação da arquitetura, como também feito um amplo estudo nos parâmetros de treinamento da rede. Já para a rede NARX-ELM foram analisadas as variantes baseadas nas redes de Elman e NARX.

O objetivo dos experimentos com a rede NARX-ESN é avaliar as diversas arquiteturas e os parâmetros necessários para sua configuração. Os resultados obtidos com as duas séries temporais utilizadas, indicaram ser suficiente como informação de entrada da rede apenas um valor (escalar), isto é, as extensões da rede NARX-ESN propostas não resultaram em melhorias na tarefa de predição de séries temporais. O uso da janela de Takens conduziu aos piores resultados dentre o modelos testados e a ordem da unidade de saída projetada para as unidades internas necessitou de apenas um único atrasador.

Outro ponto que deve ser destacado é sobre as conexões que se projetam da camada de saída e/ou da camada de entrada diretamente para a camada de saída, não passando pelo reservatório. Tendo como base os experimentos efetuados neste capítulo, verificou-se que em ambos os experimentos fez-se necessário o uso destas conexões, isto é, fazendo com que estes modelos gerassem os menores erros de predição.

Já para a escolha dos diversos parâmetros concluiu-se que realmente é uma tarefa importante para a configuração da rede NARX-ESN. Número de neurônios no reservatório, raio espectral do reservatório, valores limites dos pesos $\mathbf{W}^{in}$ e $\mathbf{W}^{back}$, e o valor da entrada fixa, mostraram que devem ser utilizados valores específicos para a rede NARX-ESN tirar o melhor proveito da capacidade preditiva. Já os valores limites dos pesos $\mathbf{W}$, a duração do transitório e a probabilidade de valores não nulos das unidades do reservatório, não resultaram num impacto importante na predição, mostrando que se pode dar menos atenção para estes parâmetros.

Para os experimentos com a rede NARX-ELM, o objetivo foi avaliar as variantes propostas aplicadas à tarefa de predição de séries temporais univariadas. Na rede ELM nenhuma recorrência é utilizada. Já nas redes baseadas na rede Elman, existe recorrência da saída da camada oculta para a entrada da rede. As redes baseadas no modelo NARX não podem ser tratadas como redes recorrentes, visto que o modelo NARX adotado é o modo Série-Paralelo, em que a saída do regressor é formado somente por valores atuais da série temporal durante o treinamento. Os resultados obtidos para as duas séries temporais utilizadas indicaram que a utilização da janela de Takens trouxe bons resultados, diferentemente do que é encontrado para a rede NARX-ESN. A rede NARX-ELM foi a que possui o melhor desempenho preditivo dentre as redes ELM avaliadas.

Embora se tenha menos parâmetros a serem definidos na rede ELM, estes devem ser escolhidos com cuidado, visto que uma melhor faixa de valores foi encontrado em cada experimento. O teste com a da σ^2 dos pesos durante a inicialização aleatória mostra que se

deve utilizar valores em torno de $0,01$ tanto para a rede ELM quanto para a rede NARX-ESN. Já os números de neurônios na camada oculta são sempre menores para a rede NARX-ELM, conseguindo esta rede obter o melhor resultado na tarefa de predição múltiplos-passos-adiante.

As redes ESN e ELM conseguiram os melhores resultados para as séries de Hénon e para a série de Mackey-Glass. Elas obtiveram uma melhor predição de longo prazo, conseguindo fazer predições mais corretas por um horizonte maior do que as redes que utilizam o *backpropagation*. Este fato foi observado principalmente para a série de Hénon, onde as predições das redes de projeções aleatórias conseguiram acompanhar por quase todo o horizonte testado ($H = 50$).

Embora as redes NARX-ESN e NARX-ELM tenham conseguindo excelentes resultados para as séries de Hénon e para a série de Mackey-Glass, o mesmo não aconteceu para a série do laser caótico. Esta série possui uma situação crítica, que ocorre por volta do instante de tempo 60, quando ocorrem súbitos colapsos da intensidade do laser, para então começar uma recuperação gradual da mesma. Desta forma, pode ser observado que as redes de projeções aleatórias não conseguiram aprender este colapso, como acontece com a rede NARX-MISO.

No próximo capítulo, são avaliados os resultados das variantes da rede de Elman. O objetivo é apresentar o desempenho das modificações da rede de Elman em problemas de predição de séries temporais univariadas.

10 RESULTADO PARA AS EXTENSÕES DA REDE DE ELMAN

10.1 Introdução

Este capítulo tem o propósito de avaliar as variantes da rede de Elman comparando-os com os desempenhos de outras arquiteturas neurais, tais como as redes NARX-MISO, NARX-MIMO e FTDNN, na tarefa de predição recursiva, usando as séries de Hénon, Mackey-Glass e do laser caótico. O objetivo deste estudo é ressaltar as diferenças no projeto de tais arquiteturas neurais recorrentes, a fim de oferecer subsídios ao usuário no momento da escolha da arquitetura mais adequada à tarefa de interesse.

10.2 Resultados para a Série de Hénon

Neste experimento é utilizando a série de Hénon, a fim de fazer a avaliação das extensões propostas da rede de Elman e de outras redes neurais comumente utilizadas em predição de séries temporais. No sistema caótico de Hénon foram adotados os valores a =1,4 e b =0,3, para produzir uma dinâmica caótica.

Para a série de Hénon são gerados 200 valores, dos quais 150 são utilizados para treino e o restante para teste. A série foi reescalonadas para a faixa de $[-1,+1]$. A heurística discutida na Seção 4.4 é adotada para seleção dos melhores parâmetros de cada modelo. São otimizados os valores da dimensão de imersão, atraso de imersão, número de épocas de treinamento, taxa de aprendizagem e número de neurônios em cada camada oculta.

Cada rodada de treino/teste de cada modelo (combinação dos parâmetros da rede neural) é repetido $K = 10$ vezes. A cada repetição, os pesos da rede são iniciados com valores aleatórios de média 0 e desvio-padrão 0,25. Quantitativamente, para a l-ésima repetição de treino/teste, o modelo é avaliado em termos do NMSE obtido após a predição recursiva no horizonte de H passos-adiante:

$$NMSE(H,l) = \frac{1}{H \cdot \sigma_x^2} \sum_{h=1}^{K} \left(x(n+h) - \hat{x}^{(l)}(n+h) \right)^2, \tag{10.1}$$

onde $x(n+h)$ é o valor observado da série temporal no instante $n+h$, $\hat{x}^{(l)}(n+h)$ é o valor predito no tempo $n+h$ para a l-ésima repetição de treino/teste, e $\hat{\sigma}_x^2$ é a variância amostral dos valores observados da série temporal.

Para cada modelo é construído uma estatística utilizando a mediana dos valores do

NMSE obtidos pelas repetições treino/teste para cada horizonte de predição h, ou seja

$$NMSE_{md} = \text{mediana}[NMSE(h,1), NMSE(h,2),\ldots,NMSE(h,N)]. \quad (10.2)$$

A Figura 73 apresenta por meio de *boxplots* os resultados produzidos pelas diversas redes neurais analisadas. Deve-se destacar que é utilizada uma escala logarítmica de base 10 para suavizar o NMSE. Analisando a predição recursiva da série de Hénon, destaque deve ser dado à rede NARX($d_E + d_y$,q_1,q_2,1), que corresponde à rede NARX-MISO com duas camadas ocultas. Esta rede obteve o melhor resultado dentre todas as redes na tarefa de predição múltiplos-passos-adiante.

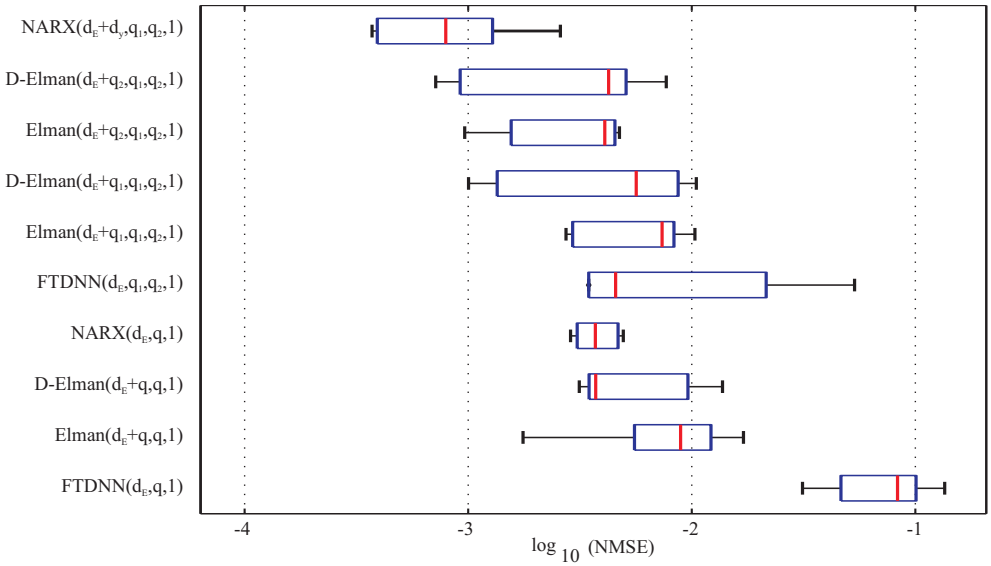


Figura 73 – Resultados obtidos com diversas RNAs para a série de Hénon (teste HPA, $H = 10$).

Na Figura 74 tem-se a predição recursiva da série de Hénon para o melhor modelo obtido (NARX-MISO). Para este resultado, a série é testada com a rede NARX($2+2$,20,18,1). Pode-se verificar visualmente que a predição recursiva consegue prever com exatidão a série observada até o horizonte $H = 20$. Embora a série predita não consiga acompanhar por mais tempo a série observada, vale notar que o modelo consegue reproduzir a dinâmica do sistema caótico.

10.3 Resultados para a série de Mackey-Glass

Neste segundo experimento deste capítulo, é analisada a série de Mackey-Glass. Novamente são avaliadas as extensões propostas da rede de Elman e de outras redes neurais

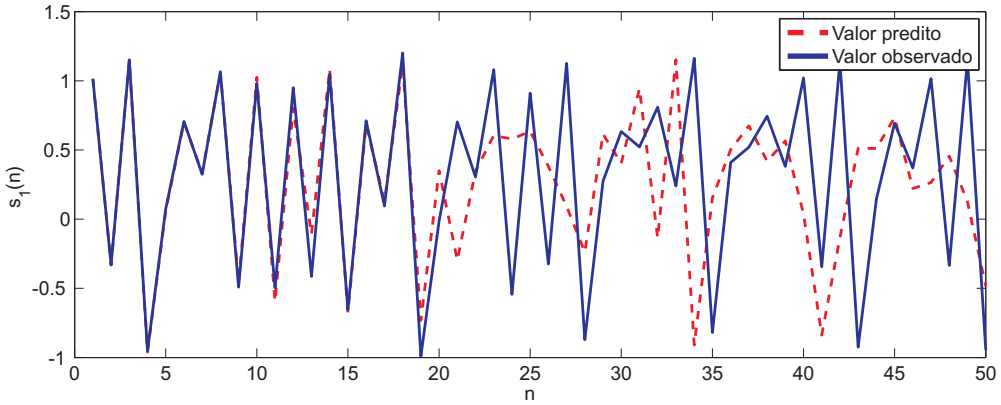


Figura 74 – Predição recursiva da série de Hénon com a rede NARX-MISO com duas camadas ocultas.

comumente utilizadas em predição de séries temporais.

As observações da série de Mackey-Glass são são utilizados os seguintes valores: α = 0,2, β = - 0,1 e Δ = 17. Para esta série são utilizadas 500 amostras, sendo que, 200 são para treino e o restante para teste. A série é reescalonada para a faixa de $[-1,+1]$.

A heurística discutida na Seção 4.4 é adotada para seleção dos melhores parâmetros de cada modelo. São otimizados os valores da dimensão de imersão, atraso de imersão, número de épocas de treinamento, taxa de aprendizagem e número de neurônios em cada camada oculta.

Cada rodada de treino/teste de cada modelo (combinação dos parâmetros da rede neural) é repetido $K = 10$ vezes. A cada repetição, os pesos da rede são iniciados com valores aleatórios de média 0 e desvio-padrão 0,25. Para cada modelo é construído uma estatística utilizando a mediana dos valores do NMSE obtidos pelas repetições treino/teste para cada horizonte de predição h.

A Figura 75 apresenta por meio de *boxplots* os resultados produzidos pelas diversas redes neurais analisadas. Deve-se destacar que é utilizada uma escala logarítmica de base 10 para suavizar o NMSE. Observa-se que as redes com duas camadas ocultas, de um modo geral, produzem melhores predições que as redes com uma camada oculta.

A rede NARX($d_E + d_y$,q_1,q_2,1) alcança resultados superiores dentre todas as outras redes analisadas. Deve-se frisar que a rede D_1-Elman($d_E + q_1$,q_1,q_2,1), rede que realimenta as derivadas das ativações da primeira camada oculta, também alcança bom desempenho, produzindo o NMSE próximo da rede NARX com duas camadas.

Na Figura 76 tem-se a predição recursiva da série de Mackey-Glass. Para gerar a predição recursiva $H = 300$ utilizou-se a rede D_1-Elman(2 + 20,20,18,1), já com os valores dos parâmetros otimizados inseridos nas denotações. Pode-se verificar visualmente que a predição

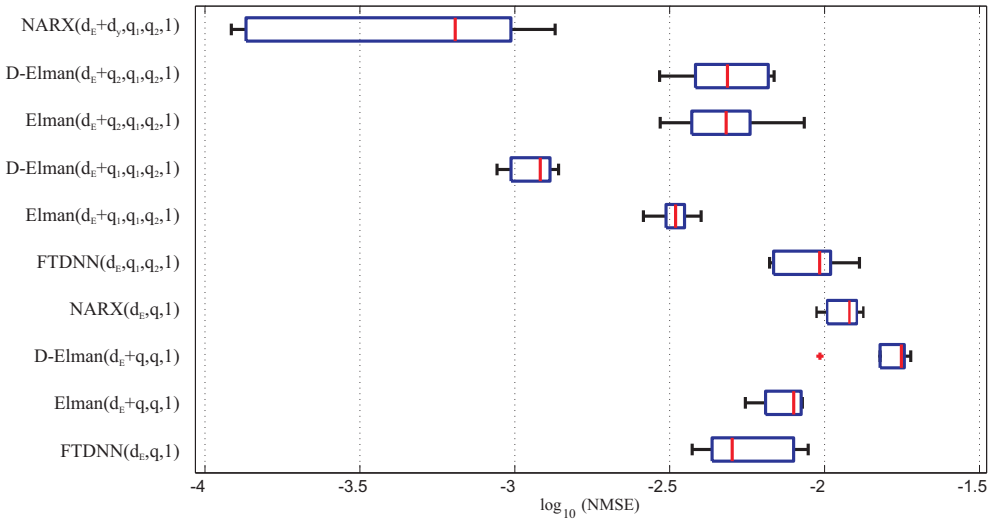


Figura 75 – Resultados obtidos com diversas RNAs para a série de Mackey-Glass (teste HPA, $H = 30$).

recursiva consegue acompanhar grande parte desta série caótica e, principalmente, consegue reproduzir a dinâmica caótica.

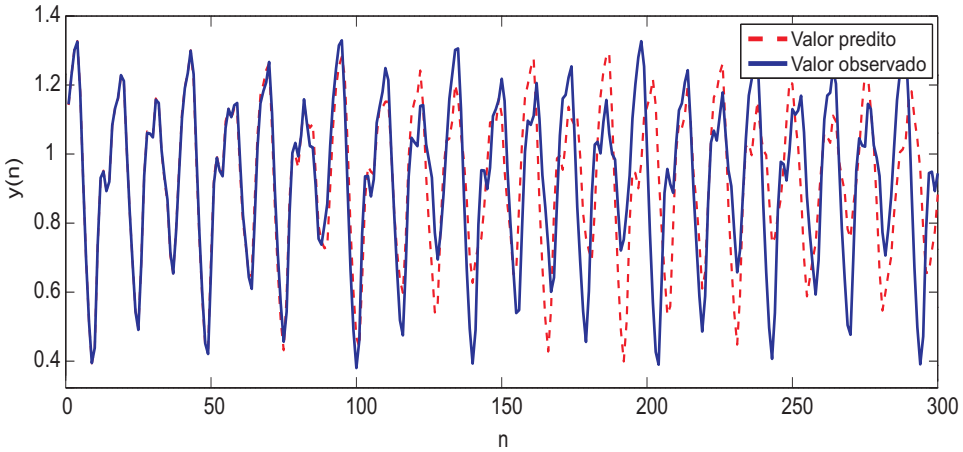


Figura 76 – Predição recursiva da série de Mackey-Glass com a rede D_1-Elman($d_E + d_1$,q_1,q_2,1).

10.4 Resultados para a Série do Laser Caótico

No próximo teste, avalia-se as redes neurais recorrentes, redes de Elman e rede NARX, para a série do laser caótico. Esta série temporal possui 1100 amostras, sendo que as 1000 primeiras são destinadas para o treino e as 100 últimas para o teste. As observações da série são normalizadas para o intervalo $[-1,+1]$. Os resultados da predição são calculados para a mediana de 10 rodadas de treino/teste com reinicializações aleatórias dos pesos da rede.

Tendo vários ciclos da heurística de busca pelos melhores parâmetros sido efetuada e satisfeita uma condição de parada, a metodologia retornará os melhores parâmetros de cada

modelo neural testado. Cada rodada de treino/teste de cada modelo (combinação dos parâmetros da rede neural) é repetido $K = 10$ vezes. A cada repetição, os pesos da rede são iniciados com valores aleatórios de média 0 e desvio-padrão 0,25. Por fim, para cada modelo é construído uma estatística utilizando a mediana dos valores do NMSE obtidos pelas repetições treino/teste para cada horizonte de predição.

A Figura 77 apresenta por meio de *boxplots* os resultados produzidos pelas diversas redes neurais analisadas. Deve-se destacar que é utilizada uma escala logarítmica de base 10 para suavizar o NMSE.

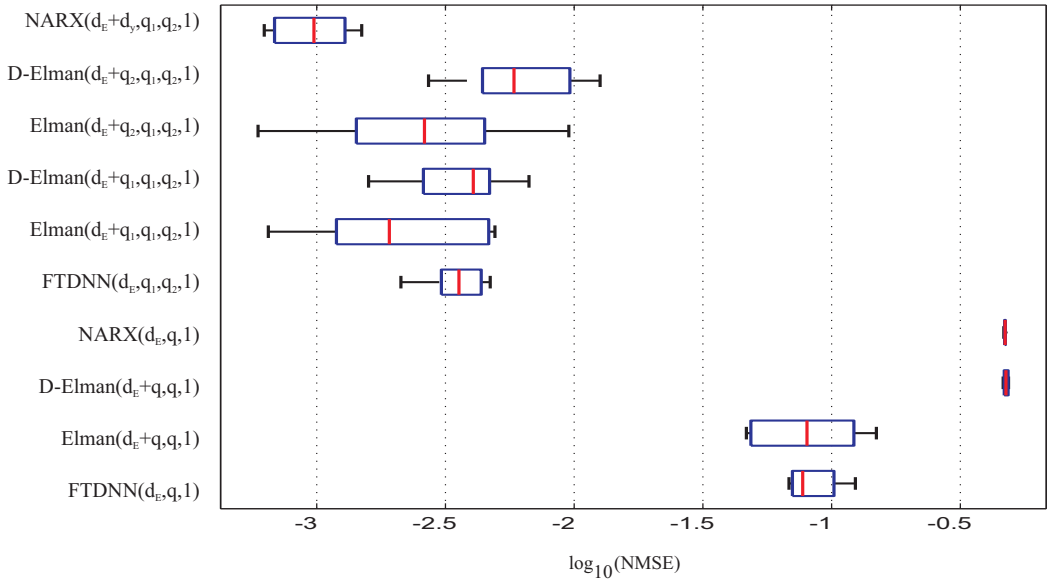


Figura 77 – Resultados obtidos com diversas RNAs para a série do laser caótico (teste HPA, $H = 50$).

Observa-se que as redes com duas camadas ocultas produzem melhores predições que as redes com uma camada oculta. Além disso, pode-se afirmar que somente as redes com duas camadas conseguem prever a região crítica da série do laser caótico, que ocorre por volta do instante de tempo 60. Embora as redes com duas camadas ocultas tenham bom desempenho na predição de múltiplos-passos-adiantes, observa-se que a rede NARX($d_E + d_y$,q_1,q_2,1), rede NARX-MISO com duas camadas ocultas, alcança resultados superiores dentre todas as outras redes analisadas. No caso rede Elman que realimenta as ativações da 1^a camada oculta, Elman($d_E + q_1$,q_1,q_2,1), pode-se verificar que esta rede também produz um baixo NMSE, mas com dispersão dos resultados maior do que a rede NARX-MISO com duas camadas ocultas.

Na Figura 78 tem-se a predição recursiva da série do laser caótico. Esta predição utiliza a rede Elman($20 + 35$,35,30,1) já com os valores dos parâmetros otimizados inseridos

na denotação. Pode-se verificar visualmente que a predição recursiva consegue acompanhar grande parte desta série caótica e, principalmente, próximo do ponto crítico. Vale ressaltar por fim que, embora a rede Elman tenha obtido este bom desempenho, a rede NARX-MISO com duas camadas ocultas ainda assim consegue reproduzir mais fielmente o colapso e reproduzir por um maior horizonte de tempo a dinâmica do sistemas, como pode ser visualizado na Figura 49.

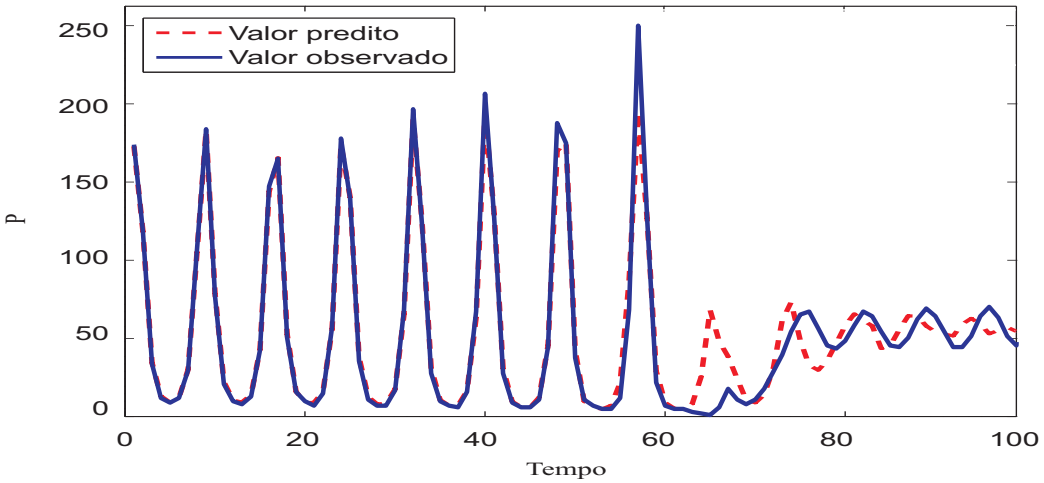


Figura 78 – Predição recursiva da série do laser caótico com a rede Elman($d_E + q_1$,q_1,q_2,1).

10.5 Conclusão

Neste capítulo foram analisados cinco extensões da rede recorrente de Elman voltadas para predição de séries temporais univariadas e comparadas com o desempenho da rede NARX-MISO. A análise dos resultados obtidos mostrou que as extensões da rede de Elman com duas camadas ocultas apresentam bom desempenho, gerando resultados superiores do que das redes com um camada oculta.

Destaque deve ser dado para a rede de Elman que realimentam as ativações ou a derivadas das ativações da primeira camada oculta. Estas redes de Elman geraram resultados muito próximos da rede NARX-MISO com duas camadas ocultas. Desta forma, as novas modificações do contexto da rede de Elman são importantes para extrair informações de curto prazo de séries temporais, conseguindo assim reproduzir a dinâmica de séries caóticas mais fielmente do que RNAs sem realimentações.

11 CONCLUSÕES E TRABALHOS FUTUROS

11.1 Resumo das Contribuições da Tese

Este último capítulo apresenta as conclusões, considerações finais, um resumo das contribuições científicas e identifica trabalhos futuros relacionados com os assuntos abordados nesta tese.

A Metodologia de Projeto e Avaliação, introduzida no Capítulo 4, implementa uma busca sistemática do melhor modelo, ou pelo menos, o que mais se adequa ao problema de interesse (no caso, predição recursiva). Esta busca mostrou-se fundamental na tentativa de estimar parâmetros dos modelos neurais, reduzir o tempo de procura por variáveis ótimas, maximizar o desempenho dos modelos e comparar preditores sob as mesmas condições. Em particular, a utilização da metodologia de seleção de parâmetros permitiu verificar que as redes NARX-ESN e NARX-ELM são sensíveis à inicialização dos pesos dos neurônios ocultos. Os testes realizados recomendam utilizar pesos aleatórios com σ^2 em torno de $0,01$, tanto para a rede NARX-ELM quanto para a rede NARX-ESN.

A rede NARX-MIMO, proposta no Capítulo 5, é uma extensão da rede NARX-MISO para predição recursiva de vários valores futuros da série simultaneamente em um dado instante de tempo. A rede NARX-MIMO gerou erros de predição menores do que os da rede NARX-MISO para as séries de Hénon e Mackey-Glass. Pode-se explicar esta melhoria pela forma como o método calcula os valores futuros da predição, fazendo com que a rede NARX-MIMO reduza o efeito da propagação dos erros de predição, de forma mais eficiente que na rede NARX-MISO.

Discutida no Capítulo 6, a rede NARX-ESN, possui diversos parâmetros a serem ajustados, o que dificulta a configuração e convergência do modelo. Foram então estudadas algumas variações da arquitetura ESN original, com a introdução da idéia do teorema de Takens e do regressor de saída do modelo NARX. Em particular, a introdução de um vetor de entrada baseado no Teorema de Takens não trouxe melhoria ao desempenho do modelo NARX-ESN, uma vez que a metodologia de seleção de parâmetros sempre selecionou apenas uma entrada por vez. Por outro lado, a realimentação dos valores preditos de saída, à guisa da rede NARX, teve papel importante na obtenção de arquiteturas com desempenho muito bom.

A rede NARX-ELM, também proposta no Capítulo 6, é a que exige menos parâmetros a serem ajustados. A rede NARX-ELM proposta apresentou o melhor desempenho dentre todas as arquiteturas neurais avaliadas para as séries de Hénon e Mackey-Glass, apresentando o

treinamento mais rápido, com menos neurônios na camada oculta e gerando os menores erros de predição.

Dentre as modificações propostas para a rede de Elman, introduzidas no Capítulo 7, destaque deve ser dado para a rede de Elman de duas camadas com duas camadas ocultas, com realimentação das ativações ou das derivadas das ativações da primeira camada oculta. Estas variantes geraram resultados muito próximos aos da rede NARX-MISO com duas camadas ocultas. Particularmente interessante, as extensões propostas da rede de Elman conseguiram reproduzir a situação crítica na predição recursiva da série do laser caótico.

A rede híbrida ELMAN/NARX-ELM, também proposto no Capítulo 7, não apresentou vantagens adicionais sobre a rede NARX-ELM, pelo menos para os conjuntos de dados avaliados.

Por fim, a título de comparação, na Tabela 10 são mostrados os resultados dos desempenhos das redes NARX-MISO, NARX-ESN e NARX-ELM na predição recursiva das séries de Hénon, Mackey-Glass e Laser Caótico, para três diferentes horizontes de predição. São mostrados os valores médios do erro e as respectivas variâncias (entre parênteses).

Tabela 10 – Análise comparativa dos desempenhos das redes NARX-MISO, NARX-ESN e NARX-ELM.

	NARX-MISO	NARX-ESN	NARX-ELM
Série de Hénon $H = 10$	8,1 x 10^{-4} (2,3 x 10^{-7})	3,3 x 10^{-12} (6,9 x 10^{-23})	**1,74 x 10^{-14}** (**4,73 x 10^{-28}**)
Série de Mackey-Glass $H = 30$	6,42 x 10^{-4} (2,88 x 10^{-7})	1,23 x 10^{-4} (7,36 x 10^{-8})	**1,31 x 10^{-4}** (**5,63 x 10^{-9}**)
Série do Laser Caótico $H = 50$	**9,76 x 10^{-4}** (**1,31 x 10^{-4}**)	4,2 x 10^{-2} (0,1177)	7,6 x 10^{-2} (4,97 x 10^{-4})

De um modo geral, pode-se concluir que as redes NARX-ESN e NARX-ELM alcançaram resultados melhores dos que os da rede NARX-MISO (treinada com backpropagation) para as séries de Hénon e Mackey-Glass. O mesmo não aconteceu para a série do laser caótico. Isto se deve basicamente ao fato de as redes baseadas em projeções aleatórias não terem conseguido aprender o colapso do sinal que ocorre por volta do instante de tempo 60, quando os valores de intensidade do laser sofrem variações súbitas. Já a rede NARX-MISO e as extensões da rede de Elman (com duas camadas ocultas) conseguiram aprender tal situação crítica, obtendo resultados superiores na tarefa de predição de múltiplos-passos-adiante da série do laser caótico.

Os baixos desempenhos das redes baseadas em projeções aleatórias obtidos nesta

tese para a série do laser caótico, estão em consonância com indícios semelhantes reportados por Sheng *et al.* (2012) para predição de séries caóticas ruidosas, por Steil (2007) para modelagem de sistemas não-lineares, e por Xue, Yang e Haykin (2007) para predição de séries não-lineares. Em todos estes trabalhados tem sido observado o baixo desempenho da rede ESN original para as séries reais ou séries ruidosas.

11.2 Propostas para Trabalhos Futuros

Esta tese possui desdobramentos que podem ser investigados como uma continuação deste trabalho de pesquisa. Com este intuito, são listados a seguir algumas potenciais linhas de pesquisa:

- Seleção dos parâmetros de cada modelo proposto nesta tese via algoritmos de Computação Evolucionária Mattos e Barreto (2011).
- Predição usando *ensemble learning* Ahmed, Athanasopoulos e Shang (2011) através de uma combinação das saídas de vários preditores baseados na rede NARX.
- Proposição de extensões das redes estudadas nesta tese que sejam robustas ao ruído.

REFERÊNCIAS

ABARBANEL, H. D.; FRISON, T. W.; TSIMRING, L. Obtaining order in a world of chaos. **IEEE Signal Processing Magazine**, v. 15, n. 3, p. 49–65, 1998.

ABARBANEL, H. D. I. *et al.* The analysis of observed chaotic data in physical systems. **Reviews of Modern Physics**, v. 65, n. 4, p. 1331–1392, 1993.

AGUIRRE, L. A. **Introdução à Identificação de Sistemas**. Belo Horizonte, MG: Editora UFMG, 2000.

AHMED, R. J. H. ans R. A.; ATHANASOPOULOS, G.; SHANG, H. L. Optimal combination forecasts for hierarchical time series. **Computational Statistics and Data Analysis**, v. 55, n. 9, p. 2579–2589, 2011.

ARDALANI-FARSA, M.; ZOLFAGHARI, S. Chaotic time series prediction with residual analysis method using hybrid elman-NARX neural networks. **Neurocomputing**, v. 73, n. 13–15, p. 2540–2553, 2010.

ATIYA, A. F.; ALY, M. A.; PARLOS, A. G. Sparse basis selection: New results and application to adaptive prediction of video source traffic. **IEEE Transactions on Neural Networks**, v. 16, n. 5, p. 1136–1146, 2005.

ATIYA, A. F. *et al.* A comparison between neural-network forecasting techniques-case study: River flow forecasting. **IEEE Transactions on Neural Networks**, v. 10, n. 2, p. 402–409, 1999.

BAKKER, R. *et al.* Learning chaotic attractors by neural networks. **Neural Comput.**, MIT Press, Cambridge, MA, USA, v. 12, n. 10, p. 2355–2383, 2000. ISSN 0899-7667.

BAUM, E. B.; HAUSSLER, D. What size net gives valid generalization? **Neural Computation**, v. 1, n. 1, p. 151–160, 1989.

BENGIO, Y.; SIMARD, P.; FRASCONI, P. Learning long-term dependencies with gradient descent is difficult. **IEEE Transactions on Neural Networks**, v. 5, n. 2, p. 157–166, 1994.

BONTEMPI, G. Long term time series prediction with multi-input multi-output local learning. In: **2nd European Symposium on Time Series Prediction ESTSP08**. [S.l.: s.n.], 2008. p. 145–154.

BONTEMPI, G.; BIRATTARI, M.; BERSINI, H. Local learning for iterated time-series prediction. In: **ICML '99: Proceedings of the Sixteenth International Conference on Machine Learning**. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1999. p. 32–38. ISBN 1-55860-612-2.

BOUCHACHIA, A. Radial basis function nets for time series prediction. **International Journal of Computational Intelligence Systems**, v. 2, n. 2, p. 147–157, 2009.

BOX, G.; JENKINS, G. M.; REINSEL, G. **Time Series Analysis: Forecasting & Control**. San Francisco: Holden-Day, 1970.

BOX, G.; JENKINS, G. M.; REINSEL, G. **Time Series Analysis: Forecasting & Control**. 3rd. ed. [S.l.]: Prentice Hall, 1994.

BROOMHEAD, D.; KING, G. Extracting qualitative dynamics from experimental data. **Physica D**, v. 20, p. 217–236, 1986.

CAO, L. Practical method for determining the minimum embedding dimension of a scalar time series. **Physica D**, v. 110, n. 1–2, p. 43–50, 1997.

CASDAGLIA, M. *et al.* State space reconstruction in the presence of noise. **Physica D: Nonlinear Phenomena**, v. 51, n. 1–3, p. 52–98, 1991.

CASTRO, M. C. F. de. **Predição não-linear de series temporais usando redes neurais RBF por decomposição em componentes principais**. Tese (Doutorado) — Universidade Estadual de Campinas . Faculdade de Engenharia Elétrica e de Computação, Campinas, SP, 2001.

CHARYTONIUK, W.; CHEN, M. Neural network design for short-term load forecasting. In: . [S.l.: s.n.], 2000. p. 554–561.

CHEN, S.; BILLINGS, S. A.; GRANT, P. M. Nonlinear system identification using neural networks. **International Journal of Control**, v. 11, n. 6, p. 1191–1214, 1990.

CHTOUROU, S.; CHTOUROU, M.; HAMMAMI, O. A hybrid approach for training recurrent neural networks: application to multi-step-ahead prediction of noisy and large data sets. **Neural Computing and Applications**, v. 17, n. 3, p. 245–254, 2008.

CORREIA, M. M. R. da L. **Memória longa, agrupamento de valores extremos e assimetrias em séries financeiras**. Dissertação (Mestrado) — Mestrado em Economia, Departamento de Economia da Faculdade de Economia, Administração e Contabilidade da Universidade de São Paulo, São Paulo, SP, 1997.

COYLE, D.; PRASAD, G.; MCGINNITY, T. M. A time-series prediction approach for feature extraction in a brain-computer interface. **IEEE Transactions on Neural Systems and Rehabilitation Engineering**, v. 13, n. 4, p. 461–467, 2005.

CRONE, S.; DHAWAN, R. Forecasting seasonal time series with neural networks: A sensitivity analysis of architecture parameters. In: **International Joint Conference on Neural Networks (IJCNN2007)**. [S.l.: s.n.], 2007. p. 2099–2104.

CRONE, S. F.; HIBON, M.; NIKOLOPOULOSC, K. Advances in forecasting with neural networks? empirical evidence from the NN3 competition on time series prediction. **International Journal of Forecasting**, v. 27, n. 3, p. 635–660, Setembro 2011.

CYBENKO, G. Approximation by superpositions of a sigmoidal function. **Mathematics of Control, Signals, and Systems**, v. 2, p. 303–314, 1989.

DABLEMONT, S. *et al.* Time series forecasting with SOM and local non-linear models - Application to the DAX30 index prediction. In: **Proceedings of the 4th Workshop on Self-Organizing Maps, (WSOM)'03**. [S.l.: s.n.], 2003. p. 340–345.

DOULAMIS, A. D.; DOULAMIS, N. D.; KOLLIAS, S. D. An adaptable neural network model for recursive nonlinear traffic prediction and modelling of MPEG video sources. **IEEE Transactions on Neural Networks**, v. 14, n. 1, p. 150–166, 2003.

ELMAN, J. Finding structure in time. **Cognitive Science**, v. 14, p. 179–211, 1990.

ENGLE, R. F. Autoregressive conditional heteroskedasticity with estimates of the variance of u.k. inflation. **Physica D: Nonlinear Phenomena**, v. 50, p. 987–1008, 1982.

FARMER, J. D. Chaotic attractors of an infinite-dimensional dynamical system. **Physica D**, v. 4, p. 66–393, 1982.

FRASER, A. M.; SWINNEY, H. L. Independent coordinates for strange attractors from mutual information. **Physical Review A**, v. 33, p. 1134–40, 1986.

GERS, F. **Long Short-Term Memory in Recurrent Neural Networks**. Dissertação (Mestrado) — Universitat Hannover, Hannover, 2001.

GÓMEZ-GIL, P. *et al.* A neural network scheme for long-term forecasting of chaotic time series. **Neural Processing Letters**, v. 33, n. 3, p. 215–233, 2011.

GOOIJER, J. G. D.; HYNDMAN, R. J. 25 years of time series forecasting. **International Journal of Forecasting**, v. 22, n. 3, p. 443–473, 2006.

GRANGER, C. W. J.; ANDERSEN, A. P. Introduction to bilinear time series models. **Vandenhoeck and Ruprecht, Göttingen**, 1978.

GRASSBERGER, P.; PROCACCIA, I. Characterization of strange attractors. **Physical Review Letters**, v. 50, n. 5, p. 346–349, 1983.

GRASSBERGER, P.; PROCACCIA, I. Measuring the strangeness of strange attractors. **Physica D: Nonlinear Phenomena**, v. 9, n. 1–2, p. 189–208, 1983.

GRAVES, D.; PEDRYCZ, W. Fuzzy prediction architecture using recurrent neural networks. **Neurocomputing**, v. 72, n. 7–9, p. 1668–1678, 2009.

GROSSGLAUSER, M.; BOLOT, J. C. On the relevance of long-range dependence in network traffic. **IEEE/ACM Transactions on Networking**, v. 7, n. 4, p. 329–640, 1998.

GURESEN, E.; KAYAKUTLUA, G.; DAIM, T. U. Using artificial neural network models in stock market index prediction. **Expert Systems with Applications**, v. 38, n. 8, p. 10389–10397, 2011.

HAYKIN, S. **Neural Networks: A Comprehensive Foundation**. 2. ed. [S.l.]: Prentice Hall, 1999.

HAYKIN, S.; PRINCIPE, J. C. Making sense of a complex world. **IEEE Signal Processing Magazine**, v. 15, n. 3, p. 66–81, 1998.

HERTZ, J.; KROGH, A.; PALMER, R. G. **Introduction to the theory of neural computation**. Redwood City, CA: Addison-Wesley, 1991.

HILL, T. *et al.* Artificial neural networks for forecasting and decision making. **International Journal of Forecasting**, v. 10, p. 5–15, March 1994.

HINKLEY, D. Miscellanea: On quick choice of power transformation. **Applied Statistics**, v. 26, n. 1, p. 67–69, 1977.

HIPPERT, H. S.; PEDREIRA, C. E.; SOUZA, R. C. Neural networks for short-term load forecasting: a review and evaluation. **IEEE Transactions on Power Systems**, v. 16, n. 1, p. 44–55, 2001.

HéNON, M. A two-dimensional mapping with a strange attractor. **Communications in Mathematical Physics**, v. 50, n. 1, p. 69–77, 1976.

HORNE, B. G.; GILES, C. L. An experimental comparison of recurrent neural networks. In: TESAURO, G.; TOURETZKY, D.; LEEN, T. (Ed.). **Advances in Neural Information Processing Systems**. [S.l.]: MIT Press, 1995. v. 7, p. 697–704.

HORNIK, K. Approximation capabilities of multilayer feedforward networks. **Neural Networks**, v. 4, p. 251–257, 1991.

HORNIK, K.; STINCHCOMBE, M.; WHITE, H. Multilayer feedforward networks are universal approximators. **Neural Networks**, v. 2, p. 359–366, 1989.

HU, M. J. C. **Application of the Adaline System to Weather Forecasting**. Tese (Doutorado) — Stanford Electronic Laboratories, Stanford, CA, June 1964.

HUANG, G.; ZHU, Q.; SIEW, C. Extreme learning machine: Theory and applications. **Neurocomputing**, v. 70, n. 1-3, p. 489–501, 2006.

HUANG, G.-B.; WANG, D. H.; LAN, Y. Extreme learning machines: A survey. **International Journal of Machine Learning and Cybernetics**, v. 2, n. 2, p. 107–122, 2011.

HUBER, P. J. **Robust Statistics**. [S.l.]: John Wiley & Sons, 1981.

HÜBNER, U.; ABRAHAM, N. B.; WEISS, C. O. Dimensions and entropies of chaotic intensity pulsations in a single-mode far-infrared NH_3 laser. **Physical Review**, A 40, p. 6354–6365, 1989.

HURST, H. E. Long term storage capacity of reservoirs. **Transactions of the American Society of Civil Engineers**, v. 116, p. 770–799, 1951.

JAEGER, H. **The "echo state" approach to analysing and training recurrent neural networks**. GMD Report 148, German National Research Center for Information Technology, 2001.

JAEGER, H.; HAAS, H. Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication. **Science**, April 2004.

JORDAN, M. I. Attractor dynamics and parallelism in a connectionist sequential machine. In: **Proceedings of the 8th Annual Conference of the Cognitive Science Society**. Amherst, MA: [s.n.], 1986. p. 531–546.

KANTZ, H.; SCHREIBER, T. **Nonlinear time series analysis**. Cambridge: Cambridge University Press, 1997.

KAPLAN, D.; GLASS, L. **Understanding Nonlinear Dynamics**. New York: Springer, 1995.

KENNEL, M. B.; BROWN, R.; ABARBANEL, H. D. I. Determining embedding dimension for phase-space reconstruction using a geometrical construction. **Phys. Rev. A**, v. 45, n. 6, p. 3403–3411, 1992.

KOSKELA, T. *et al.* Time series prediction using recurrent som with local linear models. **International Journal of Knowledge-based Intelligent Engineering Systems**, v. 2, p. 60–68, 1998.

KUGIUMTZIS, D.; LILLEKJENDLIE, B.; CHRISTOPHERSEN, N. Chaotic time series - part I: Estimation of some invariant properties in state space. **Modeling, Identification and Control**, v. 15, n. 4, p. 205–224, 1994.

LAPEDES, A.; FARBER, R. **Nonlinear Signal Processing Using Neural Networks: Prediction and System Modeling**. Los Alamos, NM, 1987.

LEONTARITIS, I. J.; BILLINGS, S. A. Input-output parametric models for nonlinear systems - Part I: deterministic nonlinear systems. **International Journal of Control**, v. 41, n. 2, p. 303–328, 1985.

LIN, T.; HORNE, B. G.; GILES, C. L. How embedded memory in recurrent neural network architectures helps learning long-term temporal dependencies. **Neural Networks**, v. 11, n. 5, p. 861–868, 1998.

LIN, T. *et al.* Learning long-term dependencies in NARX recurrent neural networks. **IEEE Transactions on Neural Networks**, v. 7, n. 6, p. 1424–1438, 1996.

LIN, T. *et al.* A delay damage model selection algorithm for NARX neural networks. **IEEE Transactions on Signal Processing**, v. 45, n. 11, p. 2719–2730, 1997.

LJUNG, L. **System Identification: Theory for the user**. 2nd. ed. Englewood Cliffs, NJ: Prentice-Hall, 1999.

MACKEY, M. C.; GLASS, L. Oscillations and chaos in physiological control systems. **Science**, v. 197, p. 287–289, 1977.

MARQUEZ, L. *et al.* Neural network models for forecast: a review. In: **Proceedings of the Twenty-Fifth Hawaii International Conference on System Sciences**. [S.l.: s.n.], 1992. iv, p. 494–498.

MATTOS, C. L. C.; BARRETO, G. A. ARTIE and MUSCLE models: building ensemble classifiers from fuzzy ART and SOM networks. **Neural Computing & Applications**, 2011.

MENEZES-JÚNIOR, J. M.; BARRETO, G. A. Long-term time series prediction with the NARX network: An empirical evaluation. **Neurocomputing**, v. 71, n. 16–18, p. 3335–3343, 2008.

MENEZES-JÚNIOR, J. M.; BARRETO, G. A. Multistep-ahead prediction of rainfall precipitation using the NARX network. In: **Proceedings of the 2nd European Symposium on Time Series Prediction (ESTSP'2008)**. [S.l.: s.n.], 2008. p. 87–96.

MENEZES-JÚNIOR, J. M.; BARRETO, G. A. Extensões da rede recorrente de elman para predição não-linear de séries temporais caóticas: Um estudo comparativo. In: **Anais da X Conferência Brasileira de Dinâmica, Controle e Aplicações (DINCON'2011)**. [S.l.: s.n.], 2011. p. 1–6.

MENEZES-JÚNIOR, J. M.; BARRETO, G. A.; FREIRE, A. L. Redes neurais recorrentes para predição recursiva de séries temporais caóticas: Um estudo comparativo. In: **Anais do XI Congresso Brasileiro de Redes Neurais (CBRN'2009)**. [S.l.: s.n.], 2009. p. 1–6.

MENEZES-JÚNIOR, J. M. P. **Redes Neurais Dinâmicas para Predição e Modelagem Não-linear de Séries Temporais**. Dissertação (Mestrado) — Programa de Pós-Graduação em Engenharia de Teleinformática, Universidade Federal do Ceará, 2006.

MICHE, Y.; SCHRAUWEN, B.; LENDASSE, A. Machine learning techniques based on random projections. In: VERLEYSEN, M. (Ed.). **ESANN2010: 18th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning**. Bruges, Belgium: [s.n.], 2010. p. 295–302.

MORETTIN, P. A.; TOLOI, C. M. C. **Análise de Séries Temporais**. São Paulo, SP: Editora Edgard Blücher, 2004.

NARENDRA, K. S.; PARTHASARATHY, K. Identification and control of dynamical systems using neural networks. **IEEE Transactions on Neural Networks**, v. 1, n. 1, p. 4–27, 1990.

NATSCHLAGER, T.; MAAS, W.; MARKRAM, H. The liquid computer: A novel strategy for real-time computing on time series. **Special Issue on Foundations of Information Processing of Telematik**, v. 8, p. 39–43, 2002.

NORGAARD, M. *et al.* **Neural Networks for Modelling and Control of Dynamic Systems**. [S.l.]: Springer, 2000.

PALIT, A. K.; POPOVIC, D. **Computational Intelligence in Time Series Forecasting**. 1st. ed. [S.l.]: Springer Verlag, 2005.

PRINCIPE, J. C.; EULIANO, N. R.; LEFEBVRE, W. C. **Neural Adaptive Systems: Fundamentals Through Simulations**. [S.l.]: John Willey and Sons, 2000.

PUSKORIUS, G. V.; FELDKAMP, L. A. Neurocontrol of nonlinear dynamical systems with kalman filter-trained recurrent networks. **IEEE Transactions on Neural Networks**, v. 5, n. 2, p. 279–297, 1994.

REDONDO, M. F.; ESPINOSA, C. H. Generalization capability of one and two hidden layers. In: **In Proceedings International Joint Conference on Neural Networks**. [S.l.: s.n.], 1999. v. 3, p. 1840–1843.

RUMELHART, D. E.; HINTON, G. E.; WILLIAMS, R. J. Learning internal representation by back- propagating errors. **Parallel Distributed Processing: Explorations in the Microstructure of Cognition**, 1986.

SAUER, T. Time series prediction by using delay coordinate embedding. In: WEIGEND, A. S.; GERSHENFELD, N. A. (Ed.). **Time Series Prediction: Forecasting the Future and Understanding the Past**. Harlow, UK: Addison Wesley, 1994. p. 175–193.

SAUER, T.; YORKE, J.; CASDAGLI, M. Embedology. **Journal of Statistical Physics**, v. 65, p. 579–616, 1991.

SAVI, M. A. **Dinâmica Não Linear e Caos**. Universidade Federal do Rio de Janeiro - COPPE, Engenharia Mecânica: [s.n.], 2004.

SCHREIBER, T. Interdisciplinary application of nonlinear time series methods. **Physics Reports**, v. 308, n. 1, p. 1–64, 1999.

SHANNON, C. A mathematical theorinproceedingsy of communication. **Bell System Technical Journal**, v. 27, p. 379–423 and 623–656, 1948.

SHENG, C. *et al.* Prediction for noisy nonlinear time series by echo state network based on dual estimation. **Neurocomputing**, v. 82, p. 186–195, 2012.

SIEGELMANN, H. T.; HORNE, B. G.; GILES, C. L. Computational capabilities of recurrent NARX neural networks. **IEEE Transactions On Systems, Man, and Cybernetics**, B-27, n. 2, p. 208–215, 1997.

SINGH, R.; BALASUNDARAM, S. Application of extreme learning machine method for time series analysis. **International Journal of Computer Systems Science and Engineering**, v. 2, n. 4, p. 256–262, 2007.

SORJAMAA, A. *et al.* Methodology for long-term prediction of time series. **Neurocomputing**, v. 70, n. 16–18, p. 2861–2869, 2007.

SOVILJ, D. *et al.* OPELM and OPKNN in long-term prediction of time series using projected input data. **Neurocomputing**, v. 73, n. 10–12, p. 148–156, 2010.

STEIL, J. J. Online reservoir adaptation by intrinsic plasticity for backpropagation-decorrelation and echo state learning. **Neural Networks**, v. 20, n. 3, p. 353–364, 2007.

TAIEB, S. B.; SORJAMAA, A.; BONTEMPI, G. Multiple-output modeling for multi-step-ahead time series forecasting. **Neurocomputing**, v. 73, n. 10–12, p. 1950–1957, 2010.

TAKENS, F. Detecting strange attractors in turbulence. In: RAND, D. A.; YOUNG, L.-S. (Ed.). **Dynamical Systems and Turbulence**. [S.l.]: Springer, 1981. (Lecture Notes in Mathematics, v. 898), p. 366–381.

TAMPELINI, L. G. *et al.* An application of elman networks in treatment and prediction of hydrologic time series. **Learning and Nonlinear Models (L&NLM) - Journal of the Brazilian Neural Network Society**, v. 9, n. 3, p. 148–156, 2011.

TIKKA, J.; HOLLMÉN, J. Sequential input selection algorithm for long-term prediction of time series. **Neurocomputing**, v. 71, n. 13–15, p. 2604–2615, 2008.

TONG, H.; LIM, K. S. Threshold autoregression, limit cycles and cyclical data. **Journal of the Royal Statistical Society**, v. 42, p. 245–292, 1980.

TSOI, A. C.; BACK, A. D. Discrete-time recurrent neural network architectures: a unifying review. **Neurocomputing**, v. 15, n. 3, p. 183–223, 1997.

VILLIERS, J. de; BARNARD, E. Backpropagation neural nets with one and two hidden layers. **IEEE Transactions on Neural Networks**, v. 4, n. 1, p. 136–141, 1993.

WAIBEL, A. *et al.* Phoneme recognition using time-delay neural networks. **IEEE Transactions on Acoustics, Speech, and Signal Processing**, v. 37, n. 3, p. 328–339, 1989.

WAN, E. A. Temporal backpropagation for FIR neural networks. In: **Proceedings of the IEEE International Joint Conference on Neural Networks**. [S.l.: s.n.], 1990. v. 1, p. 575–580.

WAN, E. A. **Finite impulse response neural networks with applications in time series prediction**. Tese (Doutorado), Stanford, CA, USA, 1994.

WANG, H.; GAO, Y. Elman's recurrent neural network applied to forecasting the quality of water diversion in the water source of lake taihu. In: **International Conference on Energy and Environmental Science (ICEES 2011)**. [S.l.: s.n.], 2011. v. 11, p. 2139–2147.

WEIGEND, A.; GERSHEFELD, N. **Time Series Prediction: Forecasting the Future and Understanding the Past**. Reading: Addison-Wesley, 1994.

WERBOS, P. **Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences**. Tese (Doutorado) — Harvard University, Cambridge, MA, 1974.

WERBOS, P. Generalization of backpropagation with application to a recurrent gas market model. **Neur. Net.**, v. 1, p. 339–356, 1986.

WERBOS, P. J. Backpropagation through time: what it does and how to do it. **Proceedings of the IEEE**, v. 78, n. 10, p. 1550–1560, 1990.

WHITNEY, H. Differentiable manifolds. **Annals of Mathematics**, v. 37, n. 3, p. 645–680, 1936.

WILLIAMS, G. P. **Chaos Theory Tamed**. Washington, DC: National Academies Press, 1997.

WILLIAMS, R. J.; ZIPSER, D. A learning algorithm for continually running fully recurrent neural networks. **Neural Computation**, v. 1, p. 270–280, 1989.

XUE, Y.; YANG, L.; HAYKIN, S. Decoupled echo state networks with lateral inhibition. **Neural Networks**, v. 20, n. 3, p. 365–376, 2007.

YULE, G. U. On a method of investigating periodicities in disturbed series, with special reference to wolfer's sunspot numbers. **Philosophical Transactions of the Royal Society of London (A)**, v. 226, p. 267–298, 1927.

ZHANG, G.; PATUWO, B. E.; HU, M. Y. Forecasting with artificial neural networks: The state of the art. **International Journal of Forecasting**, v. 14, n. 1, p. 35–62, 1998.

ZHANG, G. P. Avoiding pitfalls in neural network research. **IEEE Transactions on Systems, Man, and Cybernetics, Part C**, p. 3–16, 2007.

ZHANG, X. Time series analysis and prediction by neural networks. **Neural Networks via Mathematical Programming**, v. 4, n. 2, p. 151–170, 1994.

APÊNDICE A – MÉTODO DE CAO

A.1 Definições Preliminares

Seja uma série temporal composta por N amostras, $x(t)$, $t = 1,2,\ldots,N$:

$$\mathbf{X}_N = \{x(1),x(2),\ldots,x(N)\}. \tag{A.1}$$

A trajetória do do sinal temporal $x(t)$ no espaço de fases d-dimensional é reconstruída a partir de vetores d-dimensionais, $\mathbf{y}_t(d)$, definidos como:

$$\mathbf{y}_t(d) = [x(t),x(t+\tau),\ldots,x(t+(d-1)\tau)], \quad t = 1,2,\ldots,N-(d-1)\tau, \tag{A.2}$$

onde d é a dimensão de imersão (*embedding dimension*) e τ é o passo ou atraso de reconstrução (*time delay*). Este método de reconstrução de atratores, chamado "método dos atrasos temporais", foi proposto por (TAKENS, 1981). Este autor demonstrou que a trajetória (ou atrator) reconstruída não é idêntica à trajetória real geradora da série temporal observada, mas as características topológicas do atrator reconstruído permanecem preservadas.

A utilização dos vetores $\mathbf{y}_t(d)$ na reconstrução de atratores no espaço de fase só é possível se forem determinados valores adequados para o passo de reconstrução e para a dimensão de imersão.

Embora, em princípio, a dimensão de imersão é independente do atraso τ, a dimensão de imersão *mínima* o é. Assim, diferentes valores de τ resultam em diferentes valores para a dimensão de imersão mínima. A seguir, descreve-se um método que tem sido bastante utilizado, graças a sua simplicidade, para determinar a dimensão de imersão mínima.

A.2 Cálculo da Dimensão de Imersão pelo Método de Cao

Usando como base o método dos falsos vizinhos, descrito brevemente na Seção 2.7.1, Cao (1997) fez a seguinte definição:

$$a(t,d) = \frac{\|\mathbf{y}_t(d+1) - \mathbf{y}_{n(t,d)}(d+1)\|}{\|\mathbf{y}_t(d) - \mathbf{y}_{n(t,d)}(d)\|}, \qquad t = 1,2,\ldots,N-d\tau. \tag{A.3}$$

Onde:

- o vetor $\mathbf{y}_t(d+1)$ é o t-ésimo vetor de reconstrução com dimensão $d+1$,

$$\mathbf{y}_t(d) = [x(t),x(t+\tau),\ldots,x(t+d\tau)]. \tag{A.4}$$

- O número inteiro $n(t,d)$, $1 \leq n(t,d) \leq N - d\tau$, é tal que $\mathbf{y}_{n(t,d)}(d)$ é o vizinho mais próximo de $\mathbf{y}_t(d)$ no espaço de fase d-dimensional reconstruído, no sentido determinado pela função distância $\|\cdot\|$.
- A função distância $\|\cdot\|$ é definida como a norma máxima de seu argumento, ou seja,

$$\|\mathbf{y}_k(m) - \mathbf{y}_l(m)\| = \max_{0 \leq j \leq m-1} |x(k+j\tau) - x(l+j\tau)|. \tag{A.5}$$

É importante fazer algumas observações sobre os componentes da Equação A.3 antes de continuar a apresentação do método de Cao:

1. O número inteiro $n(t,d)$ que aparece no numerador desta equação é o mesmo que o do denominador.
2. Se $\mathbf{y}_{n(t,d)}(d)$ é igual a $\mathbf{y}_t(d)$, toma-se o segundo vizinho mais próximo em seu lugar.
3. Se d é qualificada como uma dimensão de imersão pelos teoremas de (TAKENS, 1981), então dois pontos que estão próximos no espaço de fases d-dimensional reconstruído, permanecerão próximos no espaço de fases $d+1$-dimensional. Tal par de pontos são chamados de "vizinhos verdadeiros", caso contrário são "vizinhos falsos". Esta é a idéia subjacente ao método dos "vizinhos falsos", proposto por (ABARBANEL *et al.*, 1993).

O método de Cao se baseia na definição do valor médio de todos os $a(t,d)$'s, ou seja,

$$E(d) = \frac{1}{N - d\tau} \sum_{t=1}^{N-d\tau} a(t,d), \tag{A.6}$$

onde $E(d)$ depende apenas da dimensão d e do passo τ. Para investigar a variação de $E(d)$ quando a dimensão aumenta de d para $d+1$, define-se a seguinte quantidade:

$$E_1(d) = \frac{E(d+1)}{E(d)}. \tag{A.7}$$

Cao verificou que $E_1(d)$ para de variar quando d é maior que um certo valor d_0 se a série provém de um atrator. Assim, o valor d_0 é tomado como a mínima dimensão de imersão.

APÊNDICE B – ITENS IMPORTANTES NO PROJETO DE UMA REDE NEURAL

B.1 Introdução

O término do treinamento da rede reural é, em geral, avaliado com base no valor da média do erro quadrático calculado ao final de cada época de treinamento

$$\varepsilon_{med} = \frac{1}{N}\sum_{n=1}^{N}\varepsilon(n) = \frac{1}{2N}\sum_{t=1}^{N}\sum_{k=1}^{m}e_k^2(n), \tag{B.1}$$

em que uma *época* de treinamento corresponde à apresentação de todos os pares entrada-saída disponíveis. Neste trabalho, a rede neural é treinada por várias épocas até que um número máximo de épocas permitido seja alcançado. O gráfico de ε_{med} pelo número de épocas é chamado de curva de aprendizagem da rede neural.

Para avaliar o desempenho da rede treinada é importante avaliar a sua resposta aos dados de entrada diferentes daqueles utilizados durante o treinamento, calculando-se o valor de ε_{med} para estes vetores. Na fase de teste, os pesos da rede não são ajustados. Para este fim, o procedimento mais adotado consiste em treinar a rede apenas com uma parte dos dados, guardando a parte restante para ser usada no teste do desempenho da rede. Assim, ter-se-á dois conjuntos de dados, um para treinamento de tamanho $N_1 < N$, e outro de tamanho $N_2 = N - N_1$ para o teste.

O valor de ε_{med} calculado para os dados de teste é chamado de erro médio de generalização da rede, pois testa a capacidade da mesma em extrapolar o conhecimento aprendido durante o treinamento para novos casos. É importante ressaltar que, geralmente, o erro de generalização é maior do que o erro de treinamento, pois trata-se de um novo conjunto de dados, mas seu valor deve ser suficientemente baixo de modo a garantir o bom desempenho da rede neural.

Os procedimentos de treinamento e teste são repetidos por um número K $(K \gg 1)$ de vezes, a fim de se ter uma noção da variabilidade estatística das taxas de erro. Para cada bateria de treinamento e teste, os pesos são iniciados aleatoriamente. O valor final da taxa de acerto é dado então pela média das taxas obtidas para as K baterias. O intervalo de confiança da taxa de acerto também pode ser estimado a partir da amostra obtida para as K baterias de treinamento e teste.

Apesar das muitas características positivas do uso de RNAs, a construção de um preditor utilizando redes neurais não é uma tarefa trivial. Questões de modelagem que afetam

o desempenho de uma RNA devem ser consideradas com cuidado (ZHANG; PATUWO; HU, 1998). Uma decisão crítica é determinar a arquitetura apropriada, ou seja, o vetor de entrada, o número de camadas, o número de neurônios em cada camada, bem como o número de saídas. Outras decisões de projeto de rede incluem a seleção de funções de ativação dos nós das camadas ocultas e de saída, o algoritmo de treinamento, transformação ou métodos de normalização de dados, conjuntos de treinamento e teste, e medidas de desempenho. A seguir, certos itens importantes para o bom funcionamento de uma rede neural em tarefas de predição de séries temporais são analisados.

B.2 Vetor de Entrada (p)

Os dados de entrada de treinamento de uma rede são na forma de vetores de entrada ou de padrões de treinamento. Correspondentemente, cada elemento do vetor de entrada é um nó de entrada da camada de entrada da rede. Portanto, o número de nós de entrada é igual à dimensão do vetor de entrada.

Em predição de séries temporais esta dimensão se confunde com a ordem da memória ou regressão de entrada, visto que o vetor de entrada da rede MLP é construído a partir da amostra atual da série $x(n)$ e $p-1$ amostras passadas[1]

$$\begin{aligned} \mathbf{x}(n) &= [x_0(n)\ x_1(n)\ \cdots\ x_p(n)]^T, \\ &= [-1\ x(n)\ \cdots\ x(t-p+1)]^T, \end{aligned} \tag{B.2}$$

em que se nota que o valor mínimo permitido de p é 1. O limite superior para p está associado à ordem do sistema que gerou a série temporal. É importante ter em mente que um valor alto para p não indica necessariamente um melhor desempenho para a rede neural, pois pode haver redundância na informação provida.

Em problemas de predição de séries temporais, o número apropriado de nós de entrada não é fácil de se determinar. Na predição não-linear, a dimensão p está associada ao conceito de dimensão de imersão visto no Capítulo 2. Lembrando que, seja qual for a dimensão, o vetor de entrada será quase sempre composto por uma janela móvel de tamanho fixo ao longo da série.

[1] Além do termo constante $x_0 = -1$, é claro!

B.3 Número de Camadas Ocultas

Em geral, escolhe-se redes com uma ou duas camadas de neurônios escondidos. E é claro que, sem camadas ocultas, *perceptrons* simples com camada de saída linear são equivalentes aos modelos lineares de predição. Conforme já mencionado, redes de uma camada oculta são capazes de aproximar com precisão arbitrária funções contínuas, enquanto redes MLP de duas ou mais camadas podem aproximar até funções descontínuas. Muitos pesquisadores usam apenas uma camada oculta para fins de predição de séries temporais. De acordo com (REDONDO; ESPINOSA, 1999; VILLIERS; BARNARD, 1993), a capacidade de generalização da rede de uma camada oculta é melhor do que a da rede de duas camadas ocultas. Além disto, redes de duas camadas ocultas são mais propensas a cair em um mínimo local ruim.

No entanto, redes com uma camada oculta podem exigir um elevado número de neurônios ocultos, o que não é desejável, na medida em que o tempo de treinamento e a capacidade de generalização da rede se agravarão. Zhang, Patuwo e Hu (1998) aponta que uma rede não precisa de mais do que duas camadas ocultas para resolver a maioria dos problemas, incluindo os de predição. Ainda segundo os mesmos autores, uma camada oculta pode ser suficiente para solucionar a maioria dos problemas de predição. Contudo, usando duas camadas ocultas pode dar melhores resultados para alguns problemas específicos, especialmente quando uma camada de rede oculta é sobrecarregada com muitos neurônios escondidos para obter resultados satisfatórios.

O princípio da Navalha de Occam (*Occam's Razor*) sugere que se comece os testes com uma rede MLP de uma camada oculta. Caso esta não tenha produzido bons resultados, parte-se para a inclusão de uma outra camada oculta. Este é o procedimento adotado nesta pesquisa, chegando-se à conclusão de que, em algumas aplicações, as redes a serem utilizadas devem ter duas camadas ocultas.

B.4 Número de Neurônios em Cada Camada Oculta

Este item, juntamente com o anterior, definem o poder computacional da rede MLP. Mas encontrar o número ideal de neurônios da camada oculta não é uma tarefa fácil, porque depende de uma série de fatores, muito dos quais não se tem controle total. Entre os fatores mais importantes, podem-se destacar os seguintes:

1. Quantidade de dados disponíveis para treinar e testar a rede.

2. Qualidade dos dados disponíveis (ruidosos, com elementos faltantes, etc.)
3. Número de variáveis ajustáveis (pesos e limiares) da rede.
4. Nível de complexidade do problema (não-linear, descontinuo, etc.).

Um valor subótimo para o número de neurônios em cada camada oculta é geralmente encontrado por experimentação, em função da capacidade de generalização da rede. Grosso modo, esta grandeza mede o desempenho da rede neural ante situações não-previstas, ou seja, que valor de erro médio quadrático ela produz quando novos dados de entrada são apresentados. Se muitos neurônios existirem na camada oculta, a generalização é boa para os dados de treinamento, mas tende a ser ruim para os novos dados. Se existirem poucos neurônios, o desempenho é ruim também para os dados de treinamento. O valor ideal é aquele que permite atingir as especificações de desempenho adequadas tanto para os dados de treinamento, quanto para os novos dados.

Existem algumas fórmulas heurísticas que sugerem valores para o número de neurônios na camada oculta da rede MLP, porém estas regras devem ser usadas apenas para dar um valor inicial para q. O projetista deve sempre treinar e testar várias vezes uma dada rede MLP para diferentes valores de q, a fim de se certificar que a rede neural generaliza bem para dados novos, ou seja, não usados durante a fase de treinamento.

Dentre as regras heurísticas citamos a seguir três, que são comumente encontradas na literatura especializada:

Regra do valor médio - De acordo com esta fórmula o número de neurônios da camada oculta é igual ao valor médio do número de entradas (p) e o número de saídas da rede (m), ou seja:

$$q = \frac{p+m}{2}. \tag{B.3}$$

Regra da raiz quadrada - De acordo com esta fórmula o número de neurônios da camada oculta é igual a raiz quadrada do produto do número de entradas pelo número de saídas da rede, ou seja:

$$q = \sqrt{p \cdot m}. \tag{B.4}$$

Regra de *Kolmogorov* - De acordo com esta fórmula o número de neurônios da camada oculta é igual a duas vezes o número de entradas da rede adicionado de 1, ou seja:

$$q = 2p + 1. \tag{B.5}$$

Percebe-se que as regras só levam em consideração características da rede em si, como número de entradas e número de saídas, desprezando informações úteis, tais como número de dados disponíveis para treinar/testar a rede e o erro de generalização máximo aceitável.

Uma regra que define um valor inferior para q levando em consideração o número de dados de treinamento/teste é dada por:

$$q \geq \frac{N-1}{p+2}. \tag{B.6}$$

A regra geral que se deve sempre ter em mente é a seguinte: deve-se sempre ter muito mais dados que variáveis ajustáveis. Assim, se o número total de variáveis (pesos + limiares) da rede é dado por $Z = (p+1) \cdot q + (q+1) \cdot m$, então deve-se sempre tentar obedecer à seguinte relação:

$$N \gg Z. \tag{B.7}$$

Um refinamento da Equação (B.7), proposto por Baum e Haussler (1989), sugere que a relação entre o número total de variáveis da rede (Z) e a quantidade de dados disponíveis (N) deve obedecer à seguinte relação:

$$N > \frac{Z}{\varepsilon}, \tag{B.8}$$

em que $\varepsilon > 0$ é o erro percentual máximo aceitável durante o teste da rede; ou seja, se o erro aceitável é 10%, então $\varepsilon = 0,1$. Para o desenvolvimento desta equação, os autores assumem que o erro percentual durante o treinamento não deverá ser maior que $\varepsilon/2$.

Para exemplificar, assumindo que $\varepsilon = 0,1$, então tem-se que $N > 10Z$. Isto significa que para uma rede de Z variáveis ajustáveis, deve-se ter uma quantidade dez vezes maior de padrões de treinamento.

Note que se for substituído Z na Equação (B.8) e for isolado para q, chega-se à seguinte expressão que fornece o valor aproximado do número de neurônios na camada oculta:

$$q \approx \left\lceil \frac{\varepsilon N - m}{p+m+1} \right\rceil, \tag{B.9}$$

em que $\lceil u \rceil$ denota o menor inteiro maior que u.

A Equação (B.9) é bastante completa, visto que leva em consideração não só aspectos estruturais da rede MLP (número de entradas e de saídas), mas também o erro máximo tolerado para teste e o número de dados disponíveis. Portanto, seu uso é bastante recomendado.

Para o caso de uma rede neural com duas camadas ocultas, pode-se extrapolar as heurísticas discutidas anteriormente e utilizar para encontrar o número de neurônios da primeira camada oculta (q_1). Já para determinar o número de neurônios da segunda camada oculta (q_2), pode ser utilizada a seguinte fórmula

$$q_2 = \sqrt{q_1}, \tag{B.10}$$

em que o valor resultante é arredondado para o maior valor inteiro mais próximo. Para um exemplo em que $q_1 = 20$, ter-se-ía $q_2 = \sqrt{20} = 4{,}472$, resultando em $q_2 = 5$.

B.5 Funções de Ativação (ϕ)

Em tese, cada neurônio pode ter a sua própria função de ativação diferente de todos os outros neurônios. Contudo, para simplificar o projeto da rede é comum adotar a mesma função para todos os neurônios. Em geral, escolhe-se a função logística ou a tangente hiperbólica para os neurônios da camada oculta. A sigmóide logística dada por

$$v(n) = \phi(u(n)) = \frac{1}{1+\exp\{-u(n)\}}, \tag{B.11}$$

e a tangente hiperbólica é dada por

$$v(n) = \phi(u(n)) = \frac{1-\exp\{-u(n)\}}{1+\exp\{-u(n)\}}. \tag{B.12}$$

O domínio destas funções é a reta dos números reais. Contudo, a imagem da função sigmóide logística está restrita ao intervalo $[0,1]$, enquanto a imagem da tangente hiperbólica está restrita a $[-1,+1]$. De um extremo a outro, ambas são monotonicamente crescentes.

Aquela função que for escolhida para os neurônios da camada oculta será adotada também para os neurônios da camada de saída. Contudo, em algumas aplicações é comum adotar uma função de ativação linear para os neurônios da camada de saída, ou seja, $\phi(u(n)) = C \cdot u(n)$, onde C é uma constante (ganho) positiva. Neste caso, tem-se que $\phi(u(n)) = C$. O fato de $\phi(u(n))$ ser linear não altera o poder computacional da rede, o que devemos lembrar sempre é que os neurônios da camada oculta devem ter uma função de ativação não-linear, obrigatoriamente.

B.6 Inicialização dos Pesos

Os pesos w_{ij} e m_{ki}, de um RNA devem ser inicializados com valores aleatórios. Formalmente, pode-se escrever:

$$\begin{aligned} &w_{ij} \sim U(a,b) \text{ ou } w_{ij} \sim N(\mu,\sigma^2) \\ &m_{ki} \sim U(a,b) \text{ ou } m_{ki} \sim N(\mu,\sigma^2) \end{aligned} \tag{B.13}$$

em que $U(a,b)$ é um número (pseudo-)aleatório uniformemente distribuído no intervalo (a,b), enquanto $N(\mu,\sigma^2)$ é um número (pseudo-)aleatório normalmente distribuído com média μ e variância σ^2.

Os valores de a, b, μ,σ^2 dependem da função de ativação utilizada. Caso a escolha seja feita pela sigmóide logística estes valores devem produzir pesos com média 0,5. Caso a escolha seja feita pela tangente hiperbólica os valores devem gerar pesos com média 0. É recomendado que a variância dos valores dos pesos seja baixa, próxima da média.

Pela experimentação, observa-se que a escolha dos valores para inicializar os pesos não é uma tarefa muito relevante. Na prática o que acontece é que, independentemente dos valores escolhidos para os pesos na inicialização, o processo de aprendizagem tende sempre a encontrar valores para os pesos que minimizem o problema em questão. Contudo, deve-se ter em mente que, se forem gerados pesos com grandes valores, pode-se levar à paralisia do treinamento, onde a rede passa a operar em uma região que a derivada da função de ativação é nula ou muito pequena e assim o ajuste dos pesos sinápticos serão nulos.

B.7 Normalização dos Dados

Antes de apresentar os exemplos de treinamento para a rede MLP é comum mudar a escala original das componentes dos vetores de entrada e de saída para a escala das funções de ativação logística (0 e 1) ou da tangente hiperbólica (-1 e 1). Algumas maneiras de se fazer esta mudança de escala são apresentadas a seguir:

- Transformação linear, [0, 1]:

$$x_t^* = \frac{x_t - x_{min}}{x_{max} - x_{min}} \tag{B.14}$$

- Normalização estatística:

$$x_t^* = \frac{x_t - \overline{x}}{s} \tag{B.15}$$

- Transformação linear simples, [0, 1]:

$$x_t^* = \frac{x_t}{x_{max}} \tag{B.16}$$

- Transformação linear, [a, b]:

$$x_t^* = (b-a) \cdot \left(\frac{x_t - x_{min}}{x_{max} - x_{min}} \right) + a \tag{B.17}$$

- Transformação para um caso particular do anterior, quando $a = -1$ e $b = +1$:

$$x_t^* = 2 \cdot \left(\frac{x_t - x_{min}}{x_{max} - x_{min}} \right) - 1 \tag{B.18}$$

A escolha do intervalo ao qual as entradas e saídas são normalizadas depende da função de ativação dos neurônios de saída, com tipicamente [0, 1] para a função logística e [-1, 1] para a função tangente hiperbólica. Muitos pesquisadores fazem a mudança de escala dos dados para o intervalo de [0,1, 0,9] para a função logística e [-0,9, 0,9] para a função tangente hiperbólica baseado no fato de que as funções de ativação não-linear geralmente têm limites assintóticos (ZHANG; PATUWO; HU, 1998).

Deve-se destacar que, com a normalização dos dados de entrada e saída, a saída da rede predita corresponderá ao intervalo normalizado. Assim, para interpretar os resultados obtidos a partir da rede, as saídas devem ser redimensionadas à escala original. Por conseguinte, as medidas de desempenho devem ser calculadas com base nas saídas redimensionadas. No entanto, apenas alguns autores especificam claramente se as medidas de desempenho são calculadas com a escala original ou transformadas.

B.8 Taxa de Aprendizagem Variável

Nas expressões de ajuste de pesos sinápticos do algoritmo de retropropagação do erro, Equações (3.16) e (3.17), é usada uma taxa de aprendizagem variável no tempo $\eta(n)$, que decai linearmente a zero com o passar das iterações de treinamento

$$\eta(n) = \eta_0 \left(1 - \frac{n}{n_{max}} \right), \tag{B.19}$$

em que η_0 é o valor inicial da taxa de aprendizagem e n_{max} é o número máximo de iterações, dado por

$$n_{max} = N \times \text{Número máximo de épocas.} \tag{B.20}$$

A ideia representada na Equação (B.19) está em começar o treinamento da rede MLP com um valor alto para η (e.g. $\eta_0 \approx 0{,}5$), para então ir decaindo o valor de $\eta(n)$ a fim de estabilizar o processo de aprendizado (HAYKIN, 1999).