UNIVERSIDADE FEDERAL DO CEARÁ

CENTRO DE TECNOLOGIA

DEPARTAMENTO DE ENGENHARIA DE TELEINFORMÁTICA

PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE TELEINFORMÁTICA

JACKSON UCHOA PONTE

REVISITANDO A ESTIMAÇÃO DO NÚMERO DE NEURÔNIOS OCULTOS DA REDE MLP USANDO ANÁLISE DE COMPONENTES PRINCIPAIS BASEADA EM KERNEL

FORTALEZA

2020

JACKSON UCHOA PONTE

REVISITANDO A ESTIMAÇÃO DO NÚMERO DE NEURÔNIOS OCULTOS DA REDE MLP USANDO ANÁLISE DE COMPONENTES PRINCIPAIS BASEADA EM KERNEL

Dissertação apresentada a Coordenação do Programa de Pós-Graduação em Engenharia de Teleinformática do Centro de Tecnologia da Universidade Federal do Ceará, como requisito parcial para obtenção do título de mestre em Engenharia de Teleinformática. Área de Concentração: Sinais e Sistemas - Reconhecimento de Padrões

Orientador: Prof. Dr. Guilherme de Alencar Barreto

FORTALEZA

2020

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)

P857r Ponte, Jackson Uchoa.
Revisitando a Estimação do Número de Neurônios Ocultos da Rede MLP Usando Análise de Componentes Principais Baseada em Kernel / Jackson Uchoa Ponte. – 2020.
60 f. : il. color.

Dissertação (mestrado) – Universidade Federal do Ceará, Centro de Tecnologia, Programa de Pós-Graduação em Engenharia de Teleinformática, Fortaleza, 2020.
Orientação: Prof. Dr. Guilherme de Alencar Barreto.

1. Perceptron multicamadas. 2. Perceptron multicamadas. 3. Model building. 4. Métodos de kernel. 5. Análise de componentes principais. I. Título.

CDD 621.38

JACKSON UCHOA PONTE

REVISITANDO A ESTIMAÇÃO DO NÚMERO DE NEURÔNIOS OCULTOS DA REDE MLP USANDO ANÁLISE DE COMPONENTES PRINCIPAIS BASEADA EM KERNEL

Dissertação apresentada a Coordenação do Programa de Pós-Graduação em Engenharia de Teleinformática do Centro de Tecnologia da Universidade Federal do Ceará, como requisito parcial para obtenção do título de mestre em Engenharia de Teleinformática. Área de Concentração: Sinais e Sistemas - Reconhecimento de Padrões

Aprovada em: 27 de Fevereiro de 2020

BANCA EXAMINADORA

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

Prof. Dr. José Manoel de Seixas
Universidade Federal do Rio de Janeiro - UFRJ

Prof. Dr. José Daniel de Alencar Santos
Instituto Federal de Educação, Ciência e Tecnologia do Ceará - IFCE

Prof. Dr. Elineudo Pinho de Moura
Universidade Federal do Ceará - UFC

Prof. Dr. Tarcisio Ferreira Maciel
Universidade Federal do Ceará - UFC

A Deus. Aos meus pais, Jocele Ribeiro Ponte e Sarah Uchoa Ponte. Aos meus irmãos, Jefferson Uchoa Ponte e Jéssica Uchoa Ponte.

AGRADECIMENTOS

Ao Prof. Dr. Guilherme de Alencar Barreto, pelas orientação, seu grande desprendimento em ajudar-nos e amizade sincera.

À Universidade Federal do Ceará, seu corpo docente, direção e administração que oportunizaram a janela que hoje vislumbro um horizonte superior, eivado pela acendrada confiança no mérito e ética aqui presentes.

À Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) e ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq). O presente trabalho foi realizado com apoio da CAPES (código de Financiamento 001) e do CNPq (processo 309451/2015-9).

RESUMO

A rede perceptron multicamadas (MLP, *multilayer perceptron*) é uma importante arquitetura clássica de redes neurais artificiais, que encontra aplicação em diversos problemas complexos de classificação de padrões e aproximação de funções. Apesar do seu amplo uso, sabe-se que o desempenho da rede MLP é fortemente dependente do número de neurônios ocultos escolhido, sendo a estimação deste hiperparâmetro responsável por boa parte do tempo gasto no projeto dessa topologia de rede neural. Isto posto, nesta dissertação é introduzida uma nova técnica para estimar de forma rápida o número de neurônios ocultos da rede MLP com uma camada oculta usando KPCA (*kernel principal componentes analysis*). Esta técnica é aplicada a três conjuntos de variáveis de estado, a saber, (*i*) saídas dos neurônios ocultos, (*ii*) erros retropropagados, e (*iii*) gradientes locais dos erros retropropagados, com o objetivo de reduzir o nível de redundância da informação carregada por estas variáveis. Uma avaliação comparativa abrangente do método proposto usando quatro conjuntos de dados reais e um conjunto artificial é levada a cabo nesta dissertação tendo como alvo problemas de classificação de padrões e aproximação de funções. Os resultados alcançados ora reportados indicam claramente um desempenho superior da técnica proposta em comparação a uma versão anteriormente proposta que usa técnicas lineares.

Palavras-Chave: Perceptron multicamadas, Classificação de padrões, Model building, Métodos de kernel, Análise de componentes principais.

ABSTRACT

The multilayer perceptron (MLP) neural network is an important classical architecture of artificial neural networks that finds application in many complex pattern classification and function approximation problems. Despite its wide use, it is known that the performance of the MLP network is strongly dependent on the number of hidden neurons, and the estimation of this hyperparameter is responsible for much of the time spent such topology. In this work we introduced a new technique for quickly estimating the number of hidden neurons in the MLP network using KPCA (Kernel Principal Components Analysis). This technique is applied to three sets of state variables, (*i*) hidden neuron outputs, (*ii*) back-propagated errors, and (*iii*) local gradients back-propagated errors, with the aim of reducing the information redundancy on these variables. A comprehensive comparative evaluation of the proposed method using four real datasets and one synthetic dataset is carried out targeting pattern classification and function approximation problems. The results achieved clearly indicate a superior performance of the proposed technique compared to a previously proposed version that uses linear techniques.

Keywords: Multilayer Perceptron, Pattern classification, Model building, Kernel methods, Principal component analysis.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

RNAs	Redes Neurais Artificiais
PCA	Análise de Componentes Principais
KPCA	Análise de Componentes Principais Kernalizada
MLP	Rede Perceptron Multicamadas
VE	Variância Explicada

LISTA DE SÍMBOLOS

$\mathbf{X}$	Matriz de padrões
$\mathbf{D}$	Matriz de rótulos
$\mathbf{W}$	Matriz de conexões sinápticas entre os padrões de entrada e os neurônios da camada oculta
$\mathbf{M}$	Matriz de conexões sinápticas entre os neurônios da camada oculta e os neurônios de saída
$\mathbf{Y}^{(h)}$	Matriz de saídas dos neurônios ocultos
$\mathbf{E}^{-(h)}$	Matriz de erros retropropagados
$\mathbf{\Delta}^{(h)}$	Matriz de gradientes locais dos erros retropropagados
$\mathbf{C}$	Matriz de covariância
$\bar{\mathbf{C}}$	Matriz de covariância no espaço de características
$\mathbf{K}$	Matriz de kernel
$\mathbf{G}_{ij}$	Matriz de Gram
$\mathbf{w}_i^T$	Vetor de pesos sinápticos entre o i-ésimo neurônio oculto e as $p+1$ unidades de entrada
$\mathbf{m}_k^T$	Vetor de pesos sinápticos do k-ésimo neurônio de saída
$\mathbf{x}$	Vetor de atributos de entrada
$\mathbf{y}$	Vetor de saída efetiva da rede
$\mathbf{d}$	Vetor de rótulos
v_t	Velocidade normal de translação
v_r	Velocidade rotacional
λ_i	i-ésimo autovetores
$\tilde{\lambda}_j$	j-ésimo autovalor da matriz de correlação
q	Número de neurônios sugerido para camada oculta da rede neural MLP
$\varphi(\cdot)$	Função de ativação
ε	Erro quadrático médio

SUMÁRIO

1 INTRODUÇÃO

A rede MLP e o algoritmo de retropropagação do erro ainda desempenham relevante papel na área de aprendizado de máquinas mesmo após mais 30 anos de sua introdução (RUMELHART; HINTON; WILLIAMS, 1986), seja na forma de um classificador com umas poucas camadas ocultas, seja integrada em arquiteturas de aprendizado profundo (SCHMIDHUBER, 2015). Porém, apesar dessa importância, derivada em grande parte da propriedade de aproximação universal (HORNIK, 1991), uma limitação dessa rede que ainda persiste em dias atuais, é a necessidade de especificação a priori (i.e., antes do treinamento) do número de neurônios da camada oculta (q).

O valor de q é geralmente encontrado por tentativa-e-erro, em função da capacidade de *generalização* da rede. Esta propriedade avalia o desempenho da rede neural ante situações não-previstas, ou seja, que resposta é obtida quando novos dados de entrada forem apresentados. Se a quantidade de neurônios na camada oculta for superestimada, o desempenho será muito bom para os dados de treinamento, mas tende a ser ruim para os novos dados. Já a quantidade de neurônios da camada oculta for subestimada, o desempenho será 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.

Embora haja um número considerável de técnicas que podem ajudar neste processo (ver Medeiros e Barreto (2013) e referências), a especificação desse hiperparâmetro não é trivial e exige muita experimentação com o conjunto de dados.

Um desses métodos, proposto por Santos, Barreto e Medeiros (2010), tem por base à aplicação de PCA a três conjuntos de variáveis, a saber, (*i*) saídas dos neurônios ocultos, (*ii*) erros retropropagados, e (*iii*) gradientes locais dos erros retropropagados, a fim de reduzir o nível de redundância da informação carregada por estas variáveis. Aspectos positivos desta técnica residem em sua linearidade e na consequente facilidade de aplicação, sendo necessário apenas construir as matrizes associadas às variáveis supracitadas e, em seguida, estimar as matrizes de covariância correspondentes. O número de neurônios ocultos é então definido como o número de autovalores mais relevantes; ou seja, àqueles associados às componentes principais. Santos, Barreto e Medeiros (2010) generalizaram a técnica proposta por Teoh, Tan e Xiang (2006), que, por sua vez, estenderam a técnica proposta por Weigend e Rumelhart (1991), estes os primeiros a usar PCA na estimação de neurônios ocultos da rede MLP em uma tarefa de predição de séries temporais.

Apesar de a proposta de Santos, Barreto e Medeiros (2010) ter sido aplicada com sucesso a vários problemas de classificação de padrões, sua melhor característica (i.e., linearidade) é justamente sua maior limitação. Sabe-se que as variáveis mencionadas no parágrafo anterior resultam do processamento não-linear da informação ao longo das camadas sucessivas da rede MLP. Assim, a aplicação de PCA padrão a estas variáveis consiste em uma aproximação, uma vez que correlações não-lineares entre as variáveis de interesse podem não ser capturadas em sua plenitude. Em suma, pode haver redundância oriunda de relações não-lineares entre as variáveis e estas não serem capturadas adequadamente pela técnica linear.

Isto posto, nesta dissertação, é introduzida um nova método para estimação do número de neurônios ocultos de redes MLP com uma camada oculta, que consiste na aplicação de um tipo de PCA não-linear, conhecido como kernel PCA (KPCA), proposto por Schölkopf, Smola e Müller (1998). A metodologia proposta é muito similar àquela descrita em Santos, Barreto e Medeiros (2010), com a troca das matrizes de covariância das variáveis de interesse pelas matrizes de kernel correspondentes. Os experimentos e resultados a serem apresentados na presente dissertação buscam, assim, averiguar uma possível superioridade da técnica proposta em comparação ao método anterior baseado em PCA. Por desempenho superior entende-se aquele que produzir maior acurácia usando o menor número de neurônios ocultos possível.

Em suma, a hipótese de trabalho a ser desenvolvida nesta dissertação é a de que o uso de KPCA leva a uma estimativa mais fiel do número adequado de neurônios ocultos da rede MLP, pois é capaz de diminuir a redundância não-linear resultante da representação dos dados de entrada pelas variáveis de interesse.

A fim de testar tal hipótese, uma avaliação comparativa abrangente do método proposto usando quatro conjuntos de dados reais e um conjunto artificial é levada a cabo nesta dissertação, tendo como alvo problemas de classificação de padrões e de aproximação de funções.

1.1 Objetivo Geral

Face ao exposto, o objetivo dessa dissertação é o desenvolvimento de uma proposta de estimação do número de neurônios ocultos da rede MLP, de uma camada oculta, que estenda a técnica linear proposta por Santos, Barreto e Medeiros (2010) por meio do uso da técnica não linear KPCA.

1.2 Objetivos Específicos

- Implementar e testar o uso da técnica kernel PCA em problemas de classificação de padrões.
- Implementar e testar o uso da técnica kernel PCA em problemas de regressão.
- Comparar os resultados obtidos pela aplicação da técnica kernel PCA com aqueles obtidos via PCA linear.

1.3 Produção científica

Ao longo do desenvolvimento desta dissertação, o seguinte artigo científico foi produzido e publicado:

- PONTE, J. U.; BARRETO, G. A, "Estimação do Número de Neurônios Ocultos da Rede MLP Usando Kernel PCA", apresentado no Congresso Brasileiro de Inteligência Computacional (CBIC'2019), 03/11/2019 - 06/11/2019, Belém - Pará.

1.4 Organização da Dissertação

Este trabalho é formado por mais sete capítulos capítulos, além deste, organizados na seguinte sequência: A rede MLP e o Algoritmo de Retropropagação dos Erros (Capítulo 2), PCA para Estimação de Neurônios Ocultos (Capítulo 3), Kernel PCA para Estimação de Neurônios Ocultos (Capítulo 4), Metodologia de Simulação (Capítulo 5), Resultados e Discussões (Capítulo 6) e Conclusões e Trabalhos Futuros (Capítulo 7).

No Capítulo 2, descreve-se a rede MLP, seu funcionamento e o algoritmo de retro-propagação dos erros. São apresentados também alguns dos métodos heurísticos de estimação do número de neurônios da camada oculta existentes na literatura especializada.

No Capítulo 3 é introduzido o conceito de variáveis de estado, usando-se de uma abordagem matricial para a descrição matemática do funcionamento da rede MLP. Aborda-se, também, a teoria da análise de componentes principais (PCA) e, por fim, são apresentados os métodos de estimação do número de neurônios ocultos da rede MLP através das técnicas baseada em PCA, que foram propostas por Santos, Barreto e Medeiros (2010).

O Capítulo 4 trata da fundamentação, propriedades e aplicações da Análise de Componentes Principais Kernalizada (KPCA). São apresentados o conceito de função de kernel e matriz de kernel. Finalmente, são introduzidas as técnicas não-lineares de estimação do número

de neurônios da camada oculta da rede MLP.

No Capítulo 5 são apresentadas os parâmetros de treinamento da rede MLP, o parâmetro da função kernel e a função kernel adotada para as simulações realizadas durante o desenvolvimento desta dissertação . Também é tópico deste capítulo a apresentação dos bancos de dados que foram usados para testar os métodos propostos nesta dissertação.

No Capítulo 6 são apresentados os resultados numéricos resultantes de uma comparação de desempenho das técnicas baseadas em PCA e KPCA para estimação do número de neurônios ocultos a partir das matrizes de estado. Também são apresentadas as discussões dos resultados para cada conjunto de dados, descritos no Capítulo 5.

No Capítulo 7 são apresentadas as conclusões que podem ser tiradas a partir dos resultados apresentados no Capítulo 6, além de trazer um tópico sobre sugestões de trabalhos futuros.

2 A REDE MLP E O ALGORITMO DE RETROPROPAGAÇÃO DOS ERROS

Neste capítulo, descreve-se a rede MLP, seu funcionamento e o algoritmo de retropropagação dos erros. São apresentados também alguns dos métodos heurísticos de estimação do número de neurônios da camada oculta existentes na literatura especializada.

2.1 Perceptron Multicamadas

De acordo com Haykin (2001, p. 183), a rede MLP é consiste tipicamente de "um conjunto de unidade sensoriais (nós de fonte) que constituem a camada de entrada, uma ou mais camadas ocultas de nós computacionais e uma camada de saída de nós computacionais." Estes nós computacionais são chamados de neurônios artificiais. Os neurônios das camadas intermediárias são chamados de neurônios ocultos ou escondidos pelo fato de não terem acesso direto à saída da rede.

O enfoque nesta dissertação é a rede MLP *feedforward* (alimentada adiante) totalmente conectada e com apenas uma camada de neurônios ocultos, como representado pela Figura 1. De acordo com Haykin (2001, p. 186), o fato da rede ser totalmente conectada "[...] significa que um neurônio em qualquer camada da rede é conectado a todas as unidades/neurônios da camada anterior, e que um sinal de entrada progride na rede para frente, da esquerda para direita e de camada em camada."

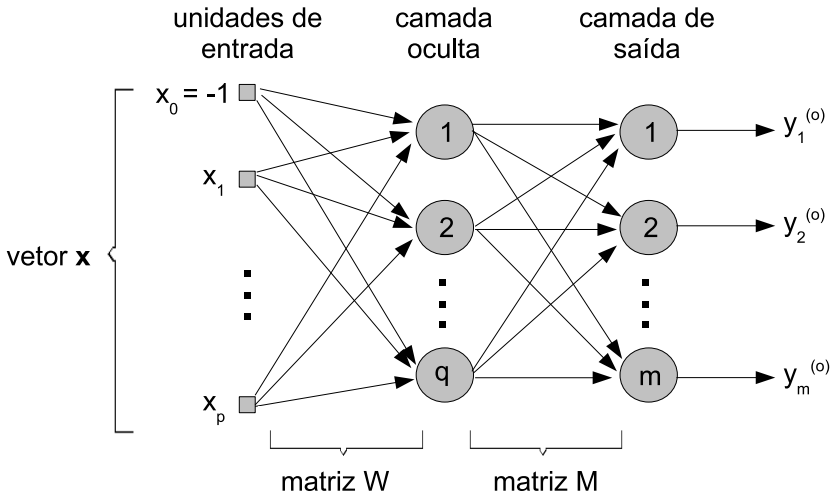


Figura 1 – Rede MLP com uma camada oculta.

Fonte – (SANTOS, 2010, p. 9)

Nesta arquitetura, $x_1, x_2, \ldots, x_p$ são as componentes do vetor de entrada **x**, o qual

deve ser acrescido da componente fixa $x_0 = -1$, correspondente ao limiar ou *bias*. Portanto o vetor $\mathbf{x}$, de dimensionalidade $p+1$ (incluindo o limiar), pode então ser representado na iteração t por

$$\mathbf{x}(t) = \begin{bmatrix} -1 \\ x_1(t) \\ \vdots \\ x_p(t) \end{bmatrix}. \tag{1}$$

É possível definir a matriz $\mathbf{X} \in \mathbb{R}^{(p+1)\times N}$ reunindo todos os vetores-coluna acima, para as N amostras do conjunto de dados utilizado na etapa de treinamento da rede. A matriz $\mathbf{X}$ pode então ser escrita como

$$\mathbf{X} = [\mathbf{x}(1) \mid \mathbf{x}(2) \mid \ldots \mid \mathbf{x}(N)]. \tag{2}$$

O número de neurônios da camada oculta, representado por q $(2 \leq q < \infty)$ na Figura 1, desempenham um papel muito importantes na rede MLP, pois agem como extratores de características (HAYKIN, 2001). Conforme o processo de aprendizagem da rede avança, estes neurônios começam gradualmente a "descobrir" características salientes presentes nos dados de treinamento. Daí então realizam uma transformação não-linear nos dados de entrada para um novo espaço, chamado espaço oculto ou espaço de características. Neste novo espaço, as classes de interesse em uma tarefa de classificação de padrões podem ser mais facilmente separadas entre si, quando se compara com o espaço original de entrada (HAYKIN, 2001).

Na camada de saída da rede MLP, m $(m \geq 1)$ designa o número de neurônios na camada de saída, os quais produzem as saídas reais obtidas pela rede. Normalmente, o número de neurônios de saída é igual ao número C de classes em um problema de classificação de padrões, na chamada codificação *1-out-of-m*. Se o interesse está em problemas de aproximação de funções, por exemplo: $\mathbf{y} = F(\mathbf{x})$, o número de neurônios deve refletir diretamente a quantidades de funções de saída desejadas (ou seja, a dimensão de $\mathbf{y}$).

A matriz $\mathbf{W} \in \mathbb{R}^{q\times(p+1)}$ é formada por todas as conexões (ou pesos) sinápticas entre as unidades de entrada e os neurônios da camada oculta. Portanto, cada elemento w_{ij} de $\mathbf{W}$ na Figura 1 representa a conexão sináptica entre a j-ésima entrada e o i-ésimo neurônio da camada

oculta. A matriz $\mathbf{W}$ pode então ser representada por

$$\mathbf{W} = \begin{bmatrix} \mathbf{w}_1^T \\ \mathbf{w}_2^T \\ \vdots \\ \mathbf{w}_q^T \end{bmatrix}, \tag{3}$$

em que cada linha $\mathbf{w}_i^T = [w_{i0}\ w_{i1}\ \ldots\ w_{ip}]$ corresponde aos pesos sinápticos entre o i-ésimo neurônio oculto e as $p+1$ unidades de entrada, incluindo o limiar.

Seguindo o mesmo princípio, define-se $\mathbf{M} \in \mathbb{R}^{m\times(q+1)}$ como a matriz formada pelas conexões sinápticas entre os neurônios da camada oculta e os neurônios da camada de saída. Cada entrada m_{ki} de $\mathbf{M}$ representa a conexão sináptica entre o i-ésimo neurônio oculto e o k-ésimo neurônio de saída. $\mathbf{M}$ pode então ser escrita como

$$\mathbf{M} = \begin{bmatrix} \mathbf{m}_1^T \\ \mathbf{m}_2^T \\ \vdots \\ \mathbf{m}_m^T \end{bmatrix}, \tag{4}$$

em que cada linha $\mathbf{m}_k^T = [m_{k0}\ m_{k1}\ \ldots\ m_{kq}]$ corresponde ao vetor de pesos sinápticos do k-ésimo neurônio de saída, incluindo o limiar.

Nesta Dissertação, o algoritmo de retropropagação do erro (*error back-propagation*), é utilizado na etapa de aprendizagem supervisionada da rede MLP. Portanto, faz-se necessária uma breve discussão sobre este algoritmo. Tal discussão apresentada não seção 2.2.

2.2 O Algoritmo de Retropropagação do Erro

Dentre os algoritmos de treinamento da rede MLP pode-se destacar o algoritmo de retropropagação do erro, o qual está descrito com maiores detalhes em Principe, Euliano e Lefebvre (2000), Haykin (2001) e Bishop (2005). Segundo (HAYKIN, 2001, p. 183), "[...] a aprendizagem por retropropagação de erro consiste de dois passos através das diferentes camadas da rede: um passo para frente, a propagação, e um passo para trás, a retropropagação." As seções seguintes discutem o treinamento da rede MLP com uma camada oculta nos sentidos direto e reverso.

2.2.1 Sentido Direto

Esta etapa de funcionamento da rede MLP 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 das unidades de entrada para os neurônios de saída, passando, obviamente, pelos neurônios da camada oculta. Por isso, diz-se que o informação está fluindo no sentido direto (*forward*), ou seja:

Entrada $\longrightarrow$ Camada Oculta $\longrightarrow$ Camada de Saída

O sentido direto da rede começa com a apresentação de um vetor de entrada $\mathbf{x}$ na iteração t. A ativação de um neurônio da camada oculta é dada por

$$u_i^{(h)}(t) = \sum_{j=0}^{p} w_{ij}(t)x_j(t) = \mathbf{w}_i^T(t)\mathbf{x}(t), \quad i = 1,\ldots,q, \tag{5}$$

em que $(\cdot)^T$ indica a operação de transposição e q indica o número de neurônios da camada escondida. Em seguida, as saídas correspondentes são calculadas como

$$y_i^{(h)}(t) = \varphi_i\left[u_i^{(h)}(t)\right] = \varphi_i\left[\sum_{j=0}^{p} w_{ij}(t)x_j(t)\right], \quad i = 1,\ldots,q, \tag{6}$$

tal que $\varphi_i(\cdot)$ é chamada de função de ativação, geralmente assumindo uma das seguintes formas, mas não restritas a elas apenas:

i) função sigmóide logística, como

$$\varphi(u(t)) = \frac{1}{1+\exp\{-u(t)\}}, \tag{7}$$

ii) função tangente hiperbólica

$$\varphi(u(t)) = \frac{1-\exp\{-u(t)\}}{1+\exp\{-u(t)\}}. \tag{8}$$

De forma similar aos neurônios ocultos, a ativação calculada para o k-ésimo neurônio da camada de saída é dada por

$$u_k^{(o)}(t) = \sum_{i=0}^{q} m_{ki}(t)y_i^{(h)}(t). \quad k = 1,\ldots,m, \tag{9}$$

É importante notar que as saídas dos neurônios da camada oculta, $y_i^{(h)}(t)$, fazem o papel de entrada para os neurônios da camada de saída. Em seguida, as saídas dos neurônios da camada de saída são calculadas como

$$y_k^{(o)}(t) = \varphi_k\left[u_k^{(o)}(t)\right] = \varphi_k\left[\sum_{i=0}^{q} m_{ki}(t)y_i^{(h)}(t)\right], \tag{10}$$

tal que a função de ativação φ_k assume geralmente uma das duas formas das Eqs. (7) e (8).

2.2.2 Sentido Reverso

Esta etapa de funcionamento da rede MLP envolve o cálculo dos gradientes locais e o ajuste dos pesos de todos os neurônios da camada escondida e da camada de saída. Assim, o fluxo de sinais, ou seja, da informação, se dá dos neurônios de saída para os neurônios da camada oculta. Por isso, diz-se que o informação está fluindo no sentido inverso (*backward*), ou seja da camada de saída para camada oculta.

Assim, após os cálculos das ativações e saídas levados a cabo no sentido direto, o primeiro passo no sentido reverso consiste em calcular os gradientes locais dos neurônios da camada de saída:

$$\delta_k^{(o)} = e_k^{(o)}(t)\varphi'(u_k(t)), \quad k = 1,\ldots,m, \tag{11}$$

em que $e_k^{(o)}(t)$ é o erro entre a saída desejada $d_k(t)$ para o k ésimo neurônio de saída e a saída gerada por ele, $y_k^{(o)}(t)$:

$$e_k^{(o)}(t) = d_k(t) - y_k^{(o)}(t), \quad k = 1,\ldots,m, \tag{12}$$

em que $\varphi'(.)$ é a derivada da função de ativação, que pode assumir diferentes formas, dependendo da escolha da função de ativação. Para o caso da sigmóide logística, sua derivada é dada por

$$\varphi'(u_k(t)) = \frac{d[\varphi(u_k(t))]}{d[u_k(t)]} = y_k(t)[1 - y_k(t)], \tag{13}$$

e, para o caso da tangente hiperbólica, tem-se

$$\varphi'(u_k(t)) = \frac{d[\varphi(u_k(t))]}{d[u_k(t)]} = 1 - y_k^2(t), \tag{14}$$

O segundo passo da etapa reversa consiste em calcular os gradientes locais dos neurônios da camada oculta, usando a seguinte equação:

$$\delta_i^{(h)}(t) = \varphi_i'(u_i(t)) \sum_{k=1}^{m} m_{ki}\delta_k^{(o)}(t), \quad i = 1,\ldots,q, \tag{15}$$

o termo $\sum_{k=1}^{m} m_{ki}\delta_k^{(o)}(t)$ corresponde ao somatório dos erros retropropagados referentes a cada neurônio de saída.

O terceiro passo no sentido reverso corresponde ao processo de atualização ou ajuste dos parâmetros (pesos sinápticos e limiares) da rede MLP (com uma camada oculta). Assim, para a camada escondida temos que a regra de atualização dos pesos, w_{ij} , é dada por

$$w_{ij}(t+1) = w_{ij}(t) + \Delta w_{ij}(t) = w_{ij}(t) + \eta\,\delta_i^{(h)}(t)x_j(t), \tag{16}$$

em que η é a taxa de aprendizagem. Para camada de saída tem-se que a regra de atualização dos pesos, m_{ki} , é dada por

$$m_{ki}(t+1) = m_{ki}(t) + \Delta m_{ki} = m_{ki}(t) + \eta\,\delta_i^{(o)}(t) y_i^{(h)}. \tag{17}$$

Durante o processo de aprendizagem, busca-se o menor valor possível do erro quadrático médio (ε_{train}), cujo valor final é calculado ao término de cada rodada de treinamento usando os próprios dados de treinamento:

$$\varepsilon_{train} = \frac{1}{2N} \sum_{t=1}^{N} \sum_{k=1}^{m} \left[d_k(t) - y_k^{(o)}(t) \right]^2, \tag{18}$$

2.3 Regras Heurísticas

Nesta seção são apresentadas algumas regras heurísticas(*ad hoc*), que estão entre as técnicas mais amplamente são utilizadas para a definição do número de neurônios da camada oculta da rede MLP. O objetivo desta seção é mostrar que tais métodos não levam em consideração informações relacionadas ao conjunto de dados envolvido no problema. Entre os fatores que entram na conta do calculo do número de neurônios da camada oculta para os método heurísticos, 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 parâmetros ajustáveis (pesos e limiares) da rede.
4. Nível de complexidade do problema (não-linear, descontínuo, etc.).

Uma desvantagem do uso das regras heurísticas é o fato de que valor sugerido q deve ser tomado apenas como um valor inicial para realização das simulações. Nesse caso, 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 a regras heurísticas destacam-se, a seguir, três que são comumente utilizados.

Regra do valor médio - De acordo com esta fórmula o número de neurônios da camada oculta é igual ao valor médio do número de unidades de entrada (p) e o número de neurônios na camada de saídas (m) da rede, ou seja

$$q = \frac{p+m}{2}. \tag{19}$$

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 unidades de entrada pelo número de neurônios na camada de saídas da rede, ou seja

$$q = \sqrt{p \cdot m}. \tag{20}$$

Regra de Kolmogorov De acordo com esta fórmula o número de neurônios da camada oculta é igual a duas vezes o número de unidades de entrada da rede adicionado de 1, ou seja

$$q = 2p + 1. \tag{21}$$

É possível perceber que as regras só levam em consideração características da rede em si, como número de unidades de entradas e número de neurônios de saídas, desprezando informações úteis, tais como número de dados disponíveis para treinar/testar a rede e o erro de generalização máximo aceitável.

2.4 Resumo do Capítulo

Foi mencionado neste capítulo que a rede MLP é uma importante arquitetura de redes neurais e que seu desempenho é fortemente influenciado pela definição do número de neurônios da camada oculta. Mencionou-se também que existem na literatura algumas formas de se fazer esta estimação de forma heurística. No capítulo 3, serão apresentados alguns métodos de estimação do número de neurônios ocultos da rede MLP através das técnicas baseada em PCA. Para tanto, será necessário introduzir o conceito de variáveis e matrizes de estado, sobre as quais atuam as técnicas abordadas.

3 PCA PARA ESTIMAÇÃO DO NÚMERO DE NEURÔNIOS OCULTOS

Neste capítulo são introduzidas as variáveis de estados usando-se de uma abordagem matricial para a descrição matemática do funcionamento da rede MLP. Aborda-se, também, a teoria da análise de componentes principais (PCA) e, por fim, são apresentados os métodos de estimação do número de neurônios ocultos da rede MLP através das técnicas baseada em PCA, as quais foram propostas por Santos, Barreto e Medeiros (2010).

3.1 Variáveis de Estado

Seja uma rede MLP(p, q_0, m) totalmente conectada, devidamente treinada, cujas matrizes $\mathbf{X} \in \mathbb{R}^{(p+1)\times N}$, $\mathbf{W} \in \mathbb{R}^{q_0\times(p+1)}$ e $\mathbf{M} \in \mathbb{R}^{m\times(q_0+1)}$ armazenam, respectivamente, os vetores de entrada (incluindo o limiar) para as N amostras de treinamento, os pesos entre as unidades de entrada e os neurônios da camada oculta e os pesos entre os neurônios ocultos e os neurônios da camada de saída.

No contexto de engenharia de controle, Nise (2010) define variáveis de estado como o menor conjunto de variáveis do sistema a partir da qual se consegue determinar o comportamento da saída do sistema para o tempo $t \geq 0$. Nesta dissertação, o conceito de variável de estado extrapolado e usado para referenciar três matrizes, a saber, (*i*) saídas dos neurônios ocultos ($\mathbf{Y}^{(h)}$), (*ii*) erros retropropagados ($\mathbf{E}^{-(h)}$) e (*iii*) gradientes locais dos erros retropropagados ($\mathbf{\Delta}^{(h)}$).

O primeiro passo para o calculo das **variáveis de estado** (ou matrizes de estado) é o treinamento da rede MLP, que segue o procedimento do algoritmo *erro backpropagation*. Esse treinamento é feito a partir de uma arquitetura MLP com o número de neurônios ocultos em excesso. Uma consequência dessa ação é a ocorrência de *overfitting*. O objetivo neste momento é garantir que a rede capture toda a informação contida no conjunto de treinamento.

Uma vez que o treinamento tenha convergido, pode-se então calcular as variáveis de estado. Para tanto, são utilizados os dados de treinamento e os pesos sinápticos previamente calculado na etapa de treinamento. Observa-se que, quando associadas a uma rede com *overfitting*, tais matrizes contêm informação redundante e, por isso, serão alvo dos métodos de redução de dimensionalidade abordados nesta dissertação. Em outras palavras, almeja-se justamente diminuir ao máximo a redundância da informação da rede MLP que está refletida nessas três matrizes de estado.

3.1.1 Matriz de ativações dos neurônios ocultos

A matriz $\mathbf{Y}^{(h)}$ contém as saídas dos neurônios ocultos para cada elemento do conjunto de treinamento exposto à rede treinada. Ao final da apresentação de todos os N vetores de treinamento, temos cada vetor de treinamento apresentado

$$\mathbf{Y}^{(h)} = \varphi_i\left[\mathbf{WX}\right], \tag{22}$$

cuja forma expandida é

$$\mathbf{Y}^{(h)} = \begin{bmatrix} y_1^{(h)}(1) & y_1^{(h)}(2) & \cdots & y_1^{(h)}(N) \\ y_2^{(h)}(1) & y_2^{(h)}(2) & \cdots & y_2^{(h)}(N) \\ \vdots & \vdots & \vdots & \vdots \\ y_{q_0}^{(h)}(1) & y_{q_0}^{(h)}(2) & \cdots & y_{q_0}^{(h)}(N) \end{bmatrix}. \tag{23}$$

A matriz $\mathbf{Y}^{(h)}$ possui dimensões $q_0 \times N$, com $N \gg q_0$ em geral.

3.1.2 Matriz de erros retropropagados

A fase de retropropagação dos erros começa a partir da camada de saída, pela projeção dos erros $e_k^{(o)}(t) = d_k(t) - y_k^{(o)}(t)$ em direção à camada oculta, em que $d_k(t)$ e $y_k^{(o)}(t)$ são respectivamente os alvos e as respostas de saída para o k-ésimo neurônio de saída. Seja o erro retropropagado para o i-ésimo neurônio oculto:

$$e_i^{(h)}(t) = \sum_{k=1}^{m} m_{ki}(t)\delta_k^{(o)}(t), \quad i = 0, \ldots, q_0, \tag{24}$$

em que m_{ki} é o peso sináptico entre o i-ésimo neurônio oculto e o k-ésimo neurônio de saída. O termo $\delta_k^{(o)}(t) = \varphi_k'(t)e_k^{(o)}(t)$ é o gradiente local do k-ésimo neurônio de saída. O termo $\varphi_k'(t)$ é a derivada da função de ativação do k-ésimo neurônio de saída.

Considere agora a matriz $\mathbf{\Delta}^{(o)} \in \mathbb{R}^{m \times N}$ como a matriz cujas N colunas são formadas pelos gradientes locais dos m neurônios de saída, construída para os N exemplos de treinamento. Tem-se assim que

$$\mathbf{\Delta}^{(o)} = [\delta^{(o)}(1) \mid \delta^{(o)}(2) \mid \ldots \mid \delta^{(o)}(N)], \tag{25}$$

na qual cada vetor $\delta^{(o)}(t)$ é definido como

$$\delta^{(o)}(t) = \begin{bmatrix} \delta_1^{(o)}(t) \\ \delta_2^{(o)}(t) \\ \vdots \\ \delta_m^{(o)}(t) \end{bmatrix}, \quad t = 1, \ldots, N. \tag{26}$$

A Eq. (24) pode ser representada na forma matricial como

$$\mathbf{E}^{(h)} = M^T \mathbf{\Delta}^{(o)}, \tag{27}$$

em que a matriz $\mathbf{E}^{(h)} \in \mathbb{R}^{(q_0+1)\times N}$ armazena em suas colunas os erros retropropagados associados aos neurônios da camada oculta, para todo o conjunto de treinamento. A matriz $\mathbf{E}^{(h)}$ na sua forma expandida é representada por

$$\mathbf{E}^{(h)} = \begin{bmatrix} e_0^{(h)}(1) & e_0^{(h)}(2) & \cdots & e_0^{(h)}(N) \\ e_1^{(h)}(1) & e_1^{(h)}(2) & \cdots & e_1^{(h)}(N) \\ \vdots & \vdots & \vdots & \vdots \\ e_{q_0}^{(h)}(1) & e_{q_0}^{(h)}(2) & \cdots & e_{q_0}^{(h)}(N) \end{bmatrix}. \tag{28}$$

Em particular, a primeira linha de $\mathbf{E}^{(h)}$ corresponde aos erros retropropagados associados aos limiares $m_{k0} = \theta_k^{(o)}$, $k = 1, \ldots, m$. Para os fins desta dissertação, a primeira linha de $\mathbf{E}^{(h)}$ não é necessária e será removida, resultando na matriz $\mathbf{E}^{-(h)} \in \mathbb{R}^{q_0\times N}$, dada por

$$\mathbf{E}^{-(h)} = \begin{bmatrix} e_1^{(h)}(1) & e_1^{(h)}(2) & \cdots & e_1^{(h)}(N) \\ e_2^{(h)}(1) & e_2^{(h)}(2) & \cdots & e_2^{(h)}(N) \\ \vdots & \vdots & \vdots & \vdots \\ e_{q_0}^{(h)}(1) & e_{q_0}^{(h)}(2) & \cdots & e_{q_0}^{(h)}(N) \end{bmatrix}. \tag{29}$$

3.1.3 Matriz de gradientes locais dos neurônios ocultos

A ultima variável de estado a ser definida é a matriz dos gradientes locais dos erros retropropagados $\mathbf{\Delta}^{(h)}$, a qual envolve os gradientes locais dos neurônios ocultos, dados por

$$\delta_i^{(h)}(t) = \varphi_i' \left[u_i^{(h)}(t) \right] \sum_{k=1}^{m} m_{ki} \delta_k^{(o)}(t) = \varphi_i' \left[u_i^{(h)}(t) \right] e_i^{(h)}(t), \quad i = 0, \ldots, q_0. \tag{30}$$

Considere a matriz $\mathbf{\Phi}^{(h)} \in \mathbb{R}^{q_0 \times N}$ como aquela formada por todas as derivadas das funções de ativação dos neurônios ocultos, calculadas para os N exemplos do conjunto de treinamento, e representada por

$$\mathbf{\Phi}^{(h)} = \left[\varphi'(1) \mid \varphi'(2) \mid \cdots \mid \varphi'(N)\right], \tag{31}$$

na qual cada vetor $\varphi'(t)$, $t = 1, 2, ..., N$, é definido como

$$\varphi'(t) = \begin{bmatrix} \varphi'_1(t) \\ \varphi'_2(t) \\ \vdots \\ \varphi'_i(t) \\ \vdots \\ \varphi'_{q_0}(t) \end{bmatrix}. \tag{32}$$

Definida a matriz $\mathbf{\Phi}^{(h)}$, a Eq. (30) pode ser representada na forma matricial como

$$\mathbf{\Delta}^{(h)} = \mathbf{\Phi}^{(h)} \odot \mathbf{E}^{-(h)}, \tag{33}$$

em que $\odot$ define o operador multiplicação componente-a-componente (*component-wise*) das matrizes envolvidas. Portanto, a matriz $\mathbf{\Delta}^{(h)} \in \mathbb{R}^{q_0 \times N}$ armazena os gradientes locais dos neurônios da camada oculta para todo o conjunto de treinamento. Em sua forma expandida $\mathbf{\Delta}^{(h)}$ pode ser escrita como

$$\mathbf{\Delta}^{(h)} = \begin{bmatrix} \delta_1^{(h)}(1) & \delta_1^{(h)}(2) & \cdots & \delta_1^{(h)}(N) \\ \delta_2^{(h)}(1) & \delta_2^{(h)}(2) & \cdots & \delta_2^{(h)}(N) \\ \vdots & \vdots & \vdots & \vdots \\ \delta_{q_0}^{(h)}(1) & \delta_{q_0}^{(h)}(2) & \cdots & \delta_{q_0}^{(h)}(N) \end{bmatrix}. \tag{34}$$

3.1.4 Diagrama de Fluxo

Para o diagrama de fluxo apresentado na Figura 2, considere a rede MLP(p, q_0, m) previamente treinada. Considere também a matriz $\mathbf{D}$ (de dimensão $m \times N$) sendo definida como a matriz de rótulos dos exemplos de treinamento, a matriz $\mathbf{E}^{(o)}$ sendo matriz contendo os vetores de erros de saída e para todos os N dados de treinamento e a matriz $\mathbf{\Phi}^{(o)} \in \mathbb{R}^{q_0 \times N}$ como aquela

formada por todas as derivadas das funções de ativação dos neurônios de saída, calculadas para os N exemplos do conjunto de treinamento, e representada por

$$\boldsymbol{\Phi}^{(o)} = \left[\boldsymbol{\varphi}'(1) \mid \boldsymbol{\varphi}'(2) \mid \cdots \mid \boldsymbol{\varphi}'(N)\right], \tag{35}$$

na qual cada vetor $\boldsymbol{\varphi}'(t)$, $t = 1,2,...,N$, é definido como

$$\boldsymbol{\varphi}'(t) = \begin{bmatrix} \varphi'_1(t) \\ \varphi'_2(t) \\ \vdots \\ \varphi'_i(t) \\ \vdots \\ \varphi'_m(t) \end{bmatrix}. \tag{36}$$

A Figura 2 tem por objetivo mostrar uma visão geral do fluxo de informação no sentido direto e reverso da rede, exemplificando como são construídas as variáveis de estado pós-treinamento.

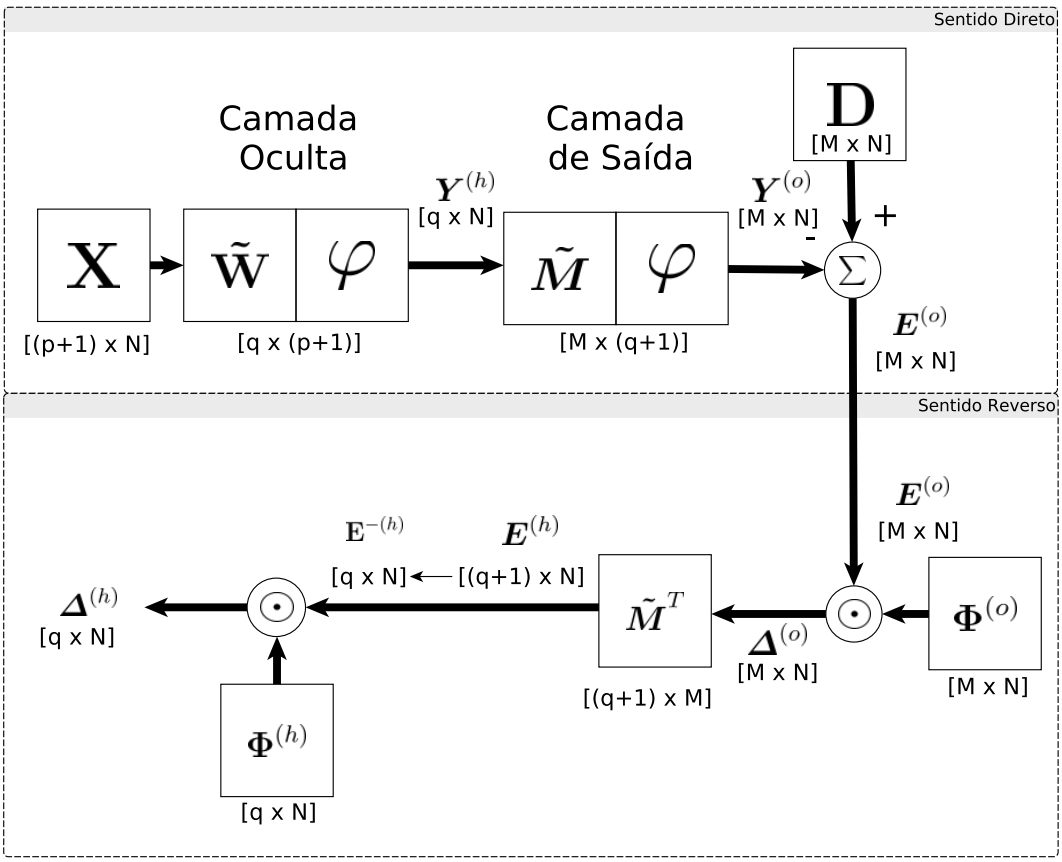


Figura 2 – Fluxo de informação nos sentidos direto e reverso da rede.

3.2 Fundamentos de PCA

Dada a importância da técnica PCA para as técnicas discutidas nesta dissertação, seus fundamentos são brevemente descritos nesta seção. PCA é uma técnica estatística multivariada de amplo uso na área de reconhecimento de padrões para fins de redução de dimensionalidade. Seja uma matriz de dados $\mathbf{X} \in \mathbb{R}^{p \times N}$ formada por N vetores coluna, cujas colunas são formadas por vetores $\mathbf{x}_i \in \mathbb{R}^p$. Assim, a dimensão da matriz $\mathbf{X}$ é $(p \times N)$ e pode ser escrita como:

$$\mathbf{X} = [\mathbf{x}_1 | \mathbf{x}_2 | \cdots | \mathbf{x}_N], \tag{37}$$

em que cada coluna $\mathbf{x}_i = [x_1, \ldots, x_p]^T$. A matriz de covariância de $\mathbf{X}$, para dados com média nula, é estimada por:

$$\mathbf{C}_x = E[\mathbf{X}\mathbf{X}^T] \approx \frac{1}{N} \sum_{i=1}^{N} \mathbf{x}_i \mathbf{x}_i^T, \tag{38}$$

em que $E[\cdot]$ é o operador valor esperado. Os autovalores da matriz de covariância são extraídos pela solução da seguinte equação característica

$$(\mathbf{C}_x - \lambda I)\mathbf{v} = 0; \quad \mathbf{v}^T \mathbf{v} = 0, \tag{39}$$

em que $\mathbf{v} \in \mathbb{R}^p$ é o autovetor e λ é seu autovalor correspondente.

A técnica PCA consiste na projeção do conjunto de dados em uma nova base vetorial, de forma que as projeções estejam nas direções de maior variância no conjunto de dados original. Em outras palavras, as coordenadas do novo sistema representam as direções de maior variabilidade dos dados. Considere a matriz de transformação $\mathbf{S} \in \mathbb{R}^{p \times q}$, em que $q \leq p$, PCA trata de uma transformação linear da forma:

$$\mathbf{Y} = \mathbf{S}^T \mathbf{X}, \tag{40}$$

em que $\mathbf{Y} \in \mathbb{R}^{q \times N}$ é matriz do conjunto de dados de saída. As colunas da matriz de transformação $\mathbf{S}$ são formadas pelas componentes principais, que são dadas pelos autovetores normalizados da matriz de covariância $\mathbf{C}_x$ e ordenadas de acordo com a ordem de grandeza dos autovalores associados, os quais devem ser organizados em ordem decrescente. A matriz de transformação pode ser escrita na forma

$$\mathbf{S} = [\mathbf{v}_1, \cdots, \mathbf{v}_q], \tag{41}$$

em que $\mathbf{v}_1, \mathbf{v}_2, \cdots \mathbf{v}_q$ são os q primeiros autovetores da matriz de covariância.

Dentre as principais aplicações de PCA encontram-se na compressão de informação (voz e imagens) (YU; KANG; KIM, 1999; OSOWSKI; MAJKOWSKI; CICHOCKI, 1997), na visualização de dados (ALAKKARI; DINGLIANA, 2018) e na redução de dimensionalidade (seleção de atributos e modelos) (GOLOVKO *et al.*, 2007; LI; XING; LUO, 2008; LUO; XIONG; WANG, 2008; GOOD; KOST; CHERRY, 2010; WEIGEND; RUMELHART, 1991).

3.3 Estimação do Número de Neurônios Ocultos via PCA

É uma técnica linear que tem por objetivo a estimação do número de neurônios da camada oculta da rede MLP. Essa técnica foi discutida e proposta por Santos, Barreto e Medeiros (2010). Segundo os propositores, a rede MLP é inicialmente treinada com um número superestimado de neurônios da camada oculta q_0, com o objetivo de calcularem-se as matrizes $\mathbf{Y}^{(h)}, \mathbf{E}^{-(h)}$ e $\mathbf{\Delta}^{(h)}$, aqui nomeadas de matrizes de estado.

A técnica consiste na determinação dos autovalores das matrizes de covariância para cada uma das três matrizes de estado, $\mathbf{Y}^{(h)}, \mathbf{E}^{-(h)}$ e $\mathbf{\Delta}^{(h)}$, ordenando-os em ordem decrescente para então se determinar o número adequado de neurônios ocultos de acordo com a seguinte critério de decisão:

$$q^* = arg\min\{VE(q) \geq \gamma\}, \tag{42}$$

em que $VE(\cdot)$ representa a variância explicada e γ representa variância explicada mínima. A variância explicada revela a quantidade de informação a ser preservada pelas q^* componentes principais selecionadas. A variância explicada é calculada como

$$VE(q) = \frac{\sum_{i=1}^{q} \lambda_i}{\sum_{i=1}^{p} \lambda_i}, \tag{43}$$

em que λ_i é o i-ésimo autovalor associado ao i-ésimo autovetor da decomposição espectral da matriz de covariância da matriz de estado em questão. Deste modo, ao aplicar PCA às matrizes de estado, busca-se encontrar o número de autovalores mais relevantes, denotado por q^*, que será associado ao número sugerido de neurônios ocultos.

É importante notar que a técnica PCA é aplicada às matrizes de covariância resultantes das matrizes de estado. As linhas das três matrizes de estado correspondem ao número

inicial de neurônios ocultos, que é propositalmente definido como uma valor superestimado. Ao se construir as três matrizes de covariância correspondentes, todas elas terão dimensões $q_0 \times q_0$.

Este processo é repetido por K vezes, sendo o número de neurônios ocultos dado pelo valor de q^* com maior frequência de ocorrência ao longo das K repetições (SANTOS, 2010). O algoritmo 1 mostra o pseudocódigo para a técnica baseada em PCA. Tendo o valor de q^* definido, a rede MLP deve ser treinada e testada novamente, objetivando a validação da nova topologia.

Algoritmo 1: Pseudocódigo para estimar o número de neurônios ocultos via PCA

Input: Dados de treinamento, número inicial de neurônios ocultos (q_0) e variância explicada mínima (γ)

Output: Número de neurônios ocultos estimado: q_1^*, q_2^* e q_3^*.

begin

Algorithm:

for $n = 1$ ***to*** K **do**

1. Treinar a rede MLP, com número de neurônios ocultos superestimado

2. Calcular as variáveis de estado

$\mathbf{Y}^{(h)}$, $\mathbf{E}^{-(h)}$ e $\mathbf{\Delta}^{(h)}$

3. Estimar o número de neurônios ocultos

(3.1) Calcular as matrizes de covariância para cada variável de estado.

(3.2) Dispor os autovalores, em ordem decrescente, para cada matriz de covariância.

(3.3) Aplicar a Eq. (42) para cada conjunto de autovalores.

end

4. Escolher os resultados q_1^*, q_2^* e q_3^* mais frequentes para as K rodadas.

end

3.4 Resumo do capítulo

Neste capítulo foi apresentado um método de estimação do número de neurônios da camada oculta baseados em PCA que associa a quantidade de neurônios ocultos com a quantidade de autovalores mais relevantes das matrizes de covariância das matrizes de estado. No próximo capítulo, será introduzida uma extensão não-linear para esta técnica, usando-se como base métodos de kernel. Para tanto, inicialmente será feito um desenvolvimento teórico

a respeito de uma versão kernelizada de PCA e por fim será apresentado a metodologia de estimação.

4 KERNEL PCA PARA ESTIMAÇÃO DE NEURÔNIOS OCULTOS

Este capítulo trata da fundamentação, propriedades e aplicações da Análise de Componentes Principais Kernalizada (KPCA). São apresentados o conceito de função de kernel e matriz de kernel. Finalmente, são mostradas as técnicas não-lineares de estimação do número de neurônios da camada oculta da rede MLP.

4.1 Detalhamento Técnico do KPCA

Bishop (2006) afirma que se um algoritmo puder ser formulado de tal forma que os vetores de entrada sejam dispostos na forma de produtos escalares então pode-se substituir o produto escalar por uma função escolhida, denominada de função kernel. Dessa forma, a técnica KPCA pode ser entendida como uma generalização do PCA, sendo aplicada a casos em que se está interessado nas componentes principais no espaço de características, que é não-linearmente relacionado com as variáveis de entrada originais (SCHÖLKOPF; SMOLA; MÜLLER, 1998).

A técnica KPCA tem sido aplicado em muitos problemas práticos tais como em extração de características (WU; WANG; LIU, 2007; DATTA; GHOSH; GHOSH, 2018), em reconhecimento de face (WANG, 2012; YANG, 2002) e em redução de ruídos (MIKA *et al.*, 1999). O objetivo do restante desta seção é mostrar o desenvolvimento matemático do KPCA a partir da teoria do PCA linear.

Considere a matriz de dados $\mathbf{X} \in \mathbb{R}^{p \times N}$ formada por N vetores coluna, cujas colunas são formadas por vetores $\mathbf{x}_i \in \mathbb{R}^p$. Considere, também, o seguinte mapeamento não-linear de cada vetor coluna $\mathbf{x}_i$ do conjunto de dados $\mathbf{X}$:

$$\begin{aligned} \phi : \mathbb{R}^p &\longrightarrow \mathbb{F} \\ \mathbf{x}_i &\mapsto \phi(\mathbf{x}_i), \end{aligned} \tag{44}$$

em que $\mathbb{F}$ é referenciado como espaço de características, geralmente um espaço de alta dimensão, tal como um espaço de Hilbert reproduzido por kernel (*reproducing kernel Hilbert space*, RKHS). O desenvolvimento teórico da técnica de KPCA foi proposto por (SCHÖLKOPF; SMOLA; MÜLLER, 1998). Assumindo que os vetores $\{\phi(\mathbf{x}_i)\}_{i=1}^N$ estão centralizados, i.e., $\sum_{i=1}^N \phi(\mathbf{x}_i) = \mathbf{0}$, a matriz de covariância $\bar{\mathbf{C}}$ pode ser escrita no espaço de características na forma

$$\bar{\mathbf{C}} = \frac{1}{N} \sum_{i=1}^{N} \phi(\mathbf{x}_i) \phi(\mathbf{x}_i)^T. \tag{45}$$

O cálculo das componentes principais no espaço de características é feito a partir da decomposição espectral da matriz de covariância $\bar{\mathbf{C}}$. Dado um vetor $\bar{\mathbf{v}} \in \mathbb{F}$, pode-se escrever a decomposição espectral

$$\tilde{\lambda}\,\bar{\mathbf{v}} = \bar{\mathbf{C}}\bar{\mathbf{v}}. \tag{46}$$

Lembrando que todas as soluções de $\bar{\mathbf{v}}$ são combinações lineares de $\{\phi(\mathbf{x}_1), \phi(\mathbf{x}_2), \ldots, \phi(\mathbf{x}_N)\}$. Portanto, existem coeficientes α_i $(i = 1, 2, \ldots, N)$ tal que

$$\bar{\mathbf{v}} = \sum_{i=1}^{N} \alpha_i \phi(\mathbf{x}_i). \tag{47}$$

Considere agora a expressão abaixo como sendo equivalente àquela da Eq. (46):

$$\tilde{\lambda}\,[\phi(\mathbf{x}_k) \cdot \bar{\mathbf{v}}] = \left[\phi(\mathbf{x}_k) \cdot \bar{\mathbf{C}}\bar{\mathbf{v}}\right], \tag{48}$$

para todo $k = 1, \ldots, N$. Assim, substituindo $\bar{\mathbf{v}}$ e $\bar{\mathbf{C}}$ das Eqs. (45) e (47) na Eq. (48), tem-se

$$\tilde{\lambda} \sum_{i=1}^{N} \alpha_i (\phi(\mathbf{x}_k)\phi(\mathbf{x}_i)) = \frac{1}{N} \sum_{i=1}^{N} \alpha_i (\phi(\mathbf{x}_k) \sum_{j=1}^{N} \phi(\mathbf{x}_j))(\phi(\mathbf{x}_j)\phi(\mathbf{x}_i)), \tag{49}$$

para todo $k = 1, \ldots, N$. Definindo uma matriz $\mathbf{K}$ $(N \times N)$, a qual será chamada de matriz de kernel, como

$$\mathbf{K}_{ij} := \phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_j) \equiv \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j), \tag{50}$$

temos então

$$N\tilde{\lambda}\mathbf{K}\boldsymbol{\alpha} = \mathbf{K}^2\boldsymbol{\alpha}, \tag{51}$$

em que $\boldsymbol{\alpha} = [\alpha_1 \ \cdots \ \alpha_N]^T$ denota um vetor coluna. De forma equivalente, tem-se

$$N\tilde{\lambda}\,\boldsymbol{\alpha} = \mathbf{K}\boldsymbol{\alpha}. \tag{52}$$

Considere que $(\lambda_1 > \ldots > \lambda_N)$ represente os autovalores da matriz de kernel $\mathbf{K}$; ou seja

$$\lambda_j = N\tilde{\lambda}_j, \tag{53}$$

com $j = 1, \ldots, N$. Em que $\tilde{\lambda}_j$ é o j-ésimo autovalor da matriz de covariância $\bar{\mathbf{C}} \in \mathbb{F}$. Então a Eq. (52) toma a forma padrão dada por

$$\lambda\,\boldsymbol{\alpha} = \mathbf{K}\boldsymbol{\alpha}, \tag{54}$$

em que $\boldsymbol{\alpha}$ desempenha papel de autovetor associado ao autovalor λ da matriz de kernel $\mathbf{K}$.

De acordo com (SCHÖLKOPF; SMOLA; MÜLLER, 1998), a exigência de que os autovetores da matriz de covariância $\bar{\mathbf{C}}$ devem ser normalizados é colocada como

$$(\bar{\mathbf{v}}^k \cdot \bar{\mathbf{v}}^k) = 1, \quad \forall k = p, \ldots, N, \tag{55}$$

em que p corresponde ao primeiro autovalor não nulo de $\bar{\mathbf{C}}$, os quais previamente são organizados em ordem crescente. A condição dada pela Eq. (55) se traduz na seguinte normalização para os autovetores $\boldsymbol{\alpha}^k$:

$$(\boldsymbol{\alpha}^k \cdot \boldsymbol{\alpha}^k) = \frac{1}{\lambda_k} \tag{56}$$

com $k = 1, \ldots, N$. Para mais detalhes consulte (SCHÖLKOPF; SMOLA; MÜLLER, 1998).

De acordo com (YANG, 2002), pode-se extrair as primeiras q $(1 \leq q \leq N)$ componentes principais não-lineares (i.e., autovetores da matriz $\mathbf{K}$) sem a operação custosa de explicitamente projetar as amostras no espaço $\mathbb{F}$. De acordo com (SCHÖLKOPF; SMOLA; MÜLLER, 1998), os autovalores de $\mathbf{K}$ darão exatamente a solução de $N\tilde{\lambda}$ da Eq. (52). Portanto, de modo similar à versão linear, a quantidade de componentes principais não-lineares q está associada com a ordem de grandeza dos autovalores e a uma tolerância γ, como mostrado mais adiante na Eq. (62).

A técnica KPCA compartilha das mesmas propriedades da PCA, mas em espaços diferentes. Ambas as técnicas requerem a solução do problema do autovalor, mas em diferentes dimensões. (DATTA; GHOSH; GHOSH, 2018). Um ponto importante a ser ressaltado é que, no caso da técnica PCA linear, o número total de autovetores (i.e., componentes principais) é igual à dimensão do vetor de atributos (p), enquanto no caso da versão não-linear, o número de autovetores é igual ao tamanho do conjunto de dados de treinamento (N). Isso pode ser visto como uma desvantagem do KPCA, pelo fato de a técnica depender do tamanho do conjunto de dados de treinamento. Isto é particularmente verdade, em problemas em que $N \gg p$.

4.2 Matriz de Kernel

Ao considerarmos a matriz de entrada $\mathbf{X} \in \mathbb{R}^{p \times N}$, a matriz Gram associada é definida como uma matriz $(N \times N)$ em que entradas são produtos internos entre os vetores colunas que compõem a matriz $\mathbf{X}$, na forma $\mathbf{G}_{ij} = \boldsymbol{\phi}(\mathbf{x}_i)^T \boldsymbol{\phi}(\mathbf{x}_j) = \boldsymbol{\phi}(\mathbf{x}_j)^T \boldsymbol{\phi}(\mathbf{x}_i)$. Graças ao chamado *truque*

do kernel (MERCER, 1909; HOFMANN; SCHÖLKOPF; SMOLA, 2008), é possível calcular tais produtos internos mesmo sem conhecer a função não-linear ϕ. Para isso, escolhe-se uma função de kernel $k(\cdot,\cdot)$, de tal forma que a matriz de kernel $\mathbf{K} = [\mathbf{K}_{ij}]_{N\times N}$ resultante seja positiva definida. Os elementos $\mathbf{K}_{ij}$, $i,j = 1,\ldots,N$, dessa matriz são definidos de tal modo a garantir a seguinte propriedade:

$$\mathbf{K}_{ij} = k(\mathbf{x}_i,\mathbf{x}_j) = \phi(\mathbf{x}_i)^T\phi(\mathbf{x}_j) = \mathbf{G}_{ij}, \tag{57}$$

ou seja, a matriz de kernel reproduz as propriedades da matriz Gram no espaço de características. A próxima seção apresenta algumas funções do kernel comumente usadas em problemas de classificação de padrões.

4.3 Função de kernel

Uma função de kernel é usada para calcular o produto interno no espaço de características. Nas subseções seguintes serão apresentadas algumas funções de kernel que foram testadas ao longo do desenvolvimento desta dissertação. Considere dois vetores, $\mathbf{x} \in \mathbb{R}^p$ e $\mathbf{y} \in \mathbb{R}^p$, nas funções de kernel apresentadas a seguir.

4.3.1 Kernel Linear

O kernel linear é o mais simples, em que a função de saída é igual ao produto interno entre dois vetores. Esse kernel pode ser formalizado como:

$$k(\mathbf{x},\mathbf{y}) = \mathbf{x}^T\mathbf{y} \tag{58}$$

4.3.2 Kernel Gaussiano

A função de kernel Gaussiano tem a seguinte formula geral:

$$k(\mathbf{x},\mathbf{y}) = \exp\left(-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{2\sigma^2}\right), \tag{59}$$

em que $\sigma > 0$ é um parâmetro escalar real, $\exp(\cdot)$ é a função exponencial e $\|\cdot\|$ denota a norma euclidiana.

4.3.3 *Kernel Cauchy*

A função de kernel Cauchy tem a seguinte formula geral:

$$k(\mathbf{x},\mathbf{y}) = \left(1 + \frac{\|\mathbf{x}-\mathbf{y}\|^2}{\sigma^2}\right)^{-1}, \tag{60}$$

em que $\sigma > 0$ é um número real. Essa função de kernel é um kernel *long-tailed*, um termo emprestado da teoria das probabilidades para denotar distribuições em que valores muito pequenos ou muito grandes têm uma grande probabilidade de ocorrer, em contraste com a distribuição gaussiana em que raramente ocorrem valores longe da média. Por esta razão, o kernel Cauchy pode ser usado para fornecer um grande intervalo de influência e sensibilidade sobre espaços de alta dimensionalidade (SOUZA, 2010).

4.3.4 *Kernel log*

Está função kernel log foi proposta por Boughorbel, Tarel e Boujemaa (2005) e sua expressão é dada pela seguinte equação:

$$k(\mathbf{x}_i,\mathbf{x}_j) = -\log\left(1 + \frac{\| \mathbf{x}_i - \mathbf{x}_j \|}{\sigma^2}\right) \tag{61}$$

em que $\log(\cdot)$ é a função logaritmo natural e e σ é um número real positivo.

A função de kernel log pertence a classe de funções de kernel não estritamente positiva definida, nomeada função de kernel *condicionalmente positiva definida* [1], a qual tem demonstrado desempenho superior em diversas aplicações.

4.4 Técnica Proposta

Esta seção descreve a metodologia proposta para estimar o número de neurônios ocultos em uma rede MLP com uma camada oculta e totalmente conectada, com base nas matrizes de estado. O procedimento começa após a etapa de aprendizagem. Neste momento, os vetores de treinamento serão apresentados mais uma vez à rede, sem alterações nas conexões sinápticas. Assim como no método descrito no capítulo anterior, o estágio inicial da metodologia consiste em construir as matrizes de estado.

[1] Seja $\mathscr{X}$ um conjunto não vazio. Uma função kernel $k(\cdot,\cdot)$ é *condicionalmente positiva definida* se e somente se ela é simétrica e $\sum_{j,k}^{n} c_j c_k k(\mathbf{x}_j,\mathbf{x}_k) \geq 0$, para $n \geq 1$, $c_1,\ldots,c_n \in \mathbb{R}$ com $\sum_{j=1}^{n} c_j = 0$ e $\mathbf{x}_1,\ldots,\mathbf{x}_n \in \mathscr{X}$.

A proposta desta dissertação consiste em usar KPCA em vez de PCA linear para estimar o número adequado de neurônios ocultos da rede MLP. A justificativa para tal abordagem reside na percepção de que as matrizes $\mathbf{Y}^{(h)}, \mathbf{E}^{-(h)}$ e $\mathbf{\Delta}^{(h)}$ são em essência resultantes de operações não-lineares via funções de ativação sigmoidais.

As razões por trás da ideia de propor uma extensão não-linear do método descrito no capítulo anterior residem no fato de que PCA é uma técnica linear, de modo que correlações não-lineares entre as variáveis de estado de interesse, ou seja, (*i*) saídas dos neurônios ocultos, (*ii*) erros retropropagados, e (*iii*) gradientes locais dos erros retropropagados, podem não ser capturadas em sua plenitude. Em suma, pode haver redundância oriunda de relações não-lineares entre as variáveis de estado. A hipótese de trabalho aqui desenvolvida é que a técnica KPCA é capaz de capturar tais relações, melhorando ainda mais a estimativa do número de neurônios ocultos.

Uma vez extraídas as matrizes de estado, o próximo passo consiste em encontrar as matrizes de kernel de cada uma delas, pois KPCA requer essas matrizes. Em seguida, os autovalores associados a cada uma das três matrizes de kernel são calculados, ordenados em ordem decrescente a fim de determinar os mais relevantes. As magnitudes desses autovalores estão diretamente relacionadas ao grau de redundância das ativações dos erros retropropagados e dos gradientes locais dos neurônios ocultos.

A metodologia proposta é muito similar àquela proposta por Santos, Barreto e Medeiros (2010), com a troca das matrizes de covariância de $\mathbf{Y}^{(h)}, \mathbf{E}^{-(h)}$ e $\mathbf{\Delta}^{(h)}$, pelas matrizes de kernel correspondentes.

Em um primeiro momento, pode parecer que os hiperparâmetros das funções de kernel sejam entendidos como uma desvantagem do método, mas na verdade não o são. Deve-se entender esta escolha do kernel e do seu hiperparâmetro como graus de liberdade a serem trabalhados a fim de encontrar uma configuração/situação em que as relações não-lineares entre as variáveis de estado fiquem evidenciadas. Este grau de liberdade confere uma maior liberdade ao método baseado em KPCA, muito embora o usuário tenha que gastar algum tempo na busca de um valor adequado.

O passo principal do método proposto consiste então em selecionar os q^* autovalores mais relevantes, ou seja, aqueles que preservam uma porcentagem pré-definida γ $(0 < \gamma < 1)$ da informação contida nas matrizes de estado. Esta informação é calculada a partir de uma medida chamada de *variância explicada* $VE(q)$ computada a partir dos autovalores da matriz de kernel.

Assim como no método baseado em PCA, o critério de decisão usado para escolher o número de neurônios ocultos usando KPCA é aquele mostrado na Eq. (42), repetida aqui para fins de completude:

$$q^* = arg\min\{VE(q^*) \geq \gamma\} \tag{62}$$

em que $VE(\cdot)$ representa a variância explicada e γ representa variância explicada mínima. A variância explicada revela a quantidade de informação a ser preservada pelas q^* componentes principais selecionadas. A variância explicada é calculada como

$$VE(q^*) = \frac{\sum_{i=1}^{q^*} \lambda_i}{\sum_{i=1}^{N} \lambda_i} \tag{63}$$

em que λ_i é o i-ésimo autovalor associado ao i-ésimo autovetor da decomposição espectral da matriz de kernel da matriz de estado em questão. Por fim, tendo sido determinado o valor mínimo de q^* , de tal modo que seja suficiente para preservar o percentual γ de informação, a nova arquitetura neural deverá passar pelo processo de treinamento e teste, agora com q^* neurônios na camada oculta, o que tem como objetivo validar a arquitetura sugerida.

Em suma, a técnica proposta baseada em KPCA funciona de modo semelhante à sua versão linear, trocando-se a matriz de covariância das variáveis de estado pela matriz de kernel das variáveis de estado.

Vale ressaltar que o processo de determinação de q^* deve ser repetido por K ($K \gg 1$) rodadas independentes, para um mesmo valor inicial (q_0) de neurônios ocultos. Essa observação é valida para as técnicas baseadas em PCA e em KPCA. Isto deve ser feito para diminuir a influência da randomização dos pesos iniciais no desempenho da rede MLP. Devido à inicialização dos pesos ser diferente a cada treinamento, a aplicação da metodologia proposta pode resultar em diferentes valores para q^* a cada execução rodada. Assim, o número de neurônios ocultos a ser escolhido ao final das K execuções é aquele com a maior frequência de ocorrência, ou seja, a moda dentre os valores sugeridos de q^* ao longo das K execuções do método.

O algoritmo 2 mostra o pseudocódigo para a técnica baseada em KPCA. Tendo o valor de q^* definido, a rede MLP deve ser treinada e testada novamente, objetivando a validação da nova topologia.

Algoritmo 2: Pseudocódigo para estimar o número de neurônios ocultos via KPCA

Input: Dados de treinamento, número inicial de neurônios ocultos (q_0), variância explicada mínima (γ), definição da função kernel de seu hyperparâmetro (σ).

Output: Número de neurônios ocultos estimado: q_1^*, q_2^* e q_3^*.

begin

Algorithm:

for $n = 1$ ***to*** K **do**

1. Treinar a rede MLP, com número de neurônios ocultos superestimado

2. Calcular as matrizes de estado

s $\mathbf{Y}^{(h)}$, $\mathbf{E}^{-(h)}$ e $\mathbf{\Delta}^{(h)}$

3. Estimação do número de neurônios ocultos

(3.1) Calcular as matrizes de kernel de cada variável de estado.

(3.2) Dispor os autovalores, em ordem decrescente, para cada matriz de kernel.

(3.3) Aplicar a eq. (62) para cada conjunto de autovalores.

end

4. Escolher os resultados q_1^*, q_2^* e q_3^* mais frequentes para as K rodadas.

end

4.5 Resumo do Capítulo

Neste capítulo, foi proposto um método não-linear para a estimação do número de neurônios da camada oculta, baseado em KPCA, que relaciona a quantidade de neurônios ocultos com a quantidade de autovalores mais relevantas das matrizes de kernel associadas às matrizes de estado. No capítulo 5, são apresentadas os parâmetros de treinamento da rede MLP, o parâmetro da função kernel e a função kernel adotada para as simulações realizadas durante o desenvolvimento desta dissertação . Também é tópico do capítulo 5 a apresentação dos bancos de dados que foram usados nos testes realizados durante do desenvolvimento desta dissertação.

5 CONJUNTOS DE DADOS E PARÂMETROS DE SIMULAÇÃO

Neste capítulo são apresentadas os parâmetros de treinamento da rede MLP, o parâmetro da função kernel e a função kernel adotada para as simulações realizadas durante o desenvolvimento desta dissertação . Também é tópico deste capítulo a apresentação dos bancos de dados que foram usados para testar os métodos propostos nesta dissertação.

5.1 Conjunto de Dados

Nesta seção, são descritas brevemente as características dos conjuntos de dados abordados nesta dissertação. São eles nomeados como: (i) Dados Artificiais, (ii) Coluna Vertebral, (iii) Câncer de Mama, (iv) Ionosfera e (v) *Wall-following*. O conjunto Wall-following possui 4 variantes, sendo duas voltadas para classificação e duas voltadas para aproximação de funções.

Os conjuntos Coluna Vertebral, Câncer de Mama e Ionosfera são disponibilizados gratuitamente para fins de pesquisa em repositório de dados da Universidade da Califórnia, campus de Irvine, podendo ser facilmente baixados a partir do sítio correspondente da internet[2] .

5.1.1 Dados Artificiais

Como o nome sugere, este conjunto foi gerado artificialmente por Medeiros e Barreto (2007). Tal conjunto é composto por dados bidimensionais divididos em duas classes não-linearmente separáveis. A Figura 3 ilustra como os dados artificiais estão distribuídos espacialmente. Foram gerados um total de 200 pontos, os quais foram igualmente distribuídos em duas classes, uma representada pelos asteriscos, a outra pelos círculos.

[2] http://archive.ics.uci.edu/ml/index.php

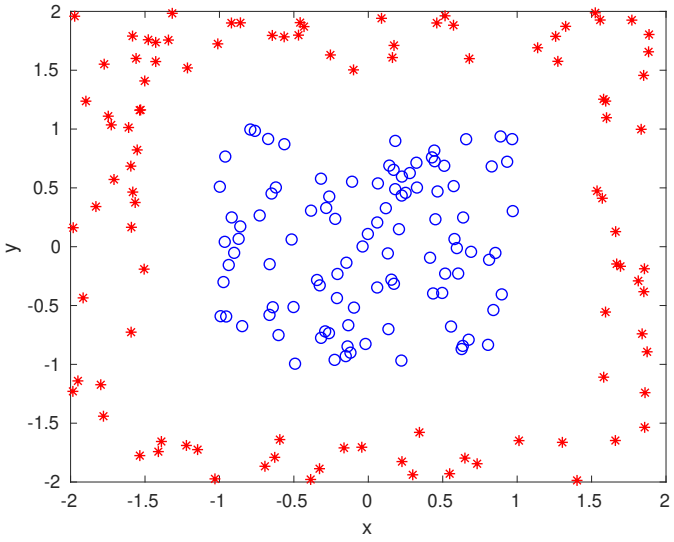


Figura 3 – Distribuição espacial dos pontos que constituem as duas classes do conjunto de dados artificiais.

5.1.2 Conjunto Coluna Vertebral

Esta base contém dados extraídos de 310 pacientes, a partir de radiografias panorâmicas sagitais em formato de 30×90 cm. Destes, 100 indivíduos são voluntários que não possuem patologias na coluna, doravante chamados de normais. Os dados restantes são obtidos a partir de radiografias de pacientes operados de hérnias de disco (60 indivíduos) ou espondilolistese (150 indivíduos).

Cada um dos 310 pacientes é descrito por seis atributos biomecânicos: ângulo de incidência pélvica, ângulo de versão pélvica, declive sacral, ângulo de lordose, raio pélvico e grau de deslizamento. A relação destes atributos com patologias comuns da coluna vertebral (e.g. hérnia de disco e espondilolistese) foi originalmente proposta na referência (BERTHONNAUD *et al.*, 2005).

Este conjunto de dados encontra-se disponível para uso público no repositório de aprendizado de máquinas da Universidade da Califórnia em Irvine[3]. Mais detalhes sobre este conjunto de dados podem ser encontrados em (ROCHA-NETO; BARRETO, 2009).

5.1.3 Conjunto Câncer de Mama

O conjunto câncer de mama *(Breast Cancer Database)* é proveniente do hospital da Universidade de Wisconsin. Este conjunto contém vetores de atributos biomédicos de pacientes,

[3] https://archive.ics.uci.edu/ml/datasets/Vertebral+Column

em que cada instância pertence a duas possíveis classes: benigno (458 pacientes) ou maligno (241 pacientes). Este conjunto de dados encontra-se disponível para uso público no repositório de aprendizado de máquinas da Universidade da Califórnia em Irvine[4].

5.1.4 Conjunto Ionosfera

O conjunto Ionosfera contém informações de sinais da ionosfera obtidos através de radares (coletados por um sistema instalado em Goose Bay, Labrador, Canadá), diferenciando aqueles que presenciam algum tipo de estrutura na ionosfera (classe 1: 225 exemplos), daqueles que nada evidenciam (classe 2: 126 exemplos). Trata-se portanto de um problema de classificação binária ($m = 2$), com 351 exemplos ao todo e $p = 34$ atributos. Entretanto, pelo fato de um dos atributos neste conjunto ter todas as componentes nulas, serão considerados $p = 33$ atributos apenas. Para o leitor interessado em mais detalhes sobre este conjunto de dados recomenda-se a leitura de Sigillito *et al.* (1989). Este conjunto de dados encontra-se disponível para uso público no repositório de aprendizado de máquinas da Universidade da Califórnia em Irvine[5].

5.1.5 Conjunto Wall-Following

O conjunto de dados *Wall-Following*, desenvolvido por Freire (2009), é composto por medidas de sensores de ultrassom do robô SCITOS G5, coletadas enquanto este executava uma algoritmo baseado em regras heurísticas, projetado para realizar da tarefa de navegação robótica, mais especificamente a tarefa de seguir paredes (*Wall-Following*).

O robô SCITOS G5 é uma plataforma mecatrônica móvel profissional para pesquisa e desenvolvimento que combina as vantagens de robôs industriais, tais como robustez e longevidade, com a mobilidade e flexibilidade de um robô de pesquisa. De acordo com (FREIRE, 2009, p. 51):

> O vetor de leituras sensoriais no instante n, $x_s(n)$ é representado por $\mathbf{x}_s(n) = [x_f(n)\ x_e(n)\ x_d(n)\ x_t(n)]^T$, em que os índices f , e, d e t indicam, respectivamente, a leitura correspondente aos sensores localizados na parte da frente, à esquerda, à direita e atrás. Cada uma dessas distâncias é extraída de um conjunto de sensores que formam juntos um feixe de 60º, e a distância utilizada é a menor encontrada pelos sensores que fazem parte do conjunto.

A coleta de dados foi realizada a uma taxa de 9 amostras por segundo, gerando, assim, um banco de dados com 5456 exemplos. Embora estejam disponíveis as 4 leituras dos

[4] https://archive.ics.uci.edu/ml/datasets/breast+cancer+wisconsin+(original)
[5] https://archive.ics.uci.edu/ml/datasets/ionosphere

sensores, o conjunto será abordado de duas formas diferentes. A primeira, serão consideradas as 4 leituras e, na segunda, apenas duas leituras.

5.1.5.1 Conjunto Wall-Following em Classificação de Padrões

O conjunto *Wall-Following* pode ser tratado como um problema de classificação de padrões, de acordo com (FREIRE, 2009):

> Neste caso, as leituras sensoriais dos ultrassons são utilizadas como entradas e as classes do problema são os quatro movimentos que podem ser realizados: Seguir-em-Frente (classe 1), Curva-Aberta-à-Direita (classe 2), Curva-Fechada-à-Direita (classe 3) e Curva-à-Esquerda (classe 4). Das 5456 amostras coletadas, 2205 pertencem à classe 1, 826 pertencem à classe 2, 2097 pertencem à classe 3 e 328 pertencem à classe 4.

5.1.5.2 Conjunto Wall-Following em Aproximação de Função

Outra forma de tratar o problema de navegação é formulando-o como um problema de regressão, em que as entradas são as leituras sensoriais (duas ou quatro entradas) e as saídas/alvos são as velocidades de translação (v_t) e de rotação (v_r) dos motores direito e esquerdo do robô. Para mais informações consulte (FREIRE *et al.*, 2009; FREIRE, 2009).

5.2 Parâmetros de Treinamento da Rede MLP

Os métodos desenvolvidos nesta dissertação foram aplicados em problemas de classificação de padrões. Além disso, diferentemente do trabalho de Santos, Barreto e Medeiros (2010), também foi avaliado o desempenho da metodologia proposta em um problema de aproximação de função. A rede MLP foi treinada usando o algoritmo *backpropagation* com a função de ativação sendo a tangente hiperbólica. Os procedimentos de treinamento e teste foram rodados 100 vezes para cada conjunto de dados.

Foi usada a estratégia *holdout* em cada rodada de treino/teste, com os exemplos sendo aleatoriamente selecionados na proporção de 80% para treinamento e 20% para testes. Os atributos foram normalizados pelo procedimento Z-score, de modo a apresentarem média zero e variância unitária. Os pesos sinápticos foram inicializados aleatoriamente no intervalo $(-0.1, +0.1)$. O passo de aprendizagem inicial foi definido como $\eta_0 = 0,75$, decaindo linearmente até $\eta_f = 0,01$. O limiar de informação mínima preservada (γ) das Eqs. (42) e (62) foi definido como γ=0,95.

O número de épocas de treinamento (N_e) foi definido no intervalo $[100, 1500]$ de

acordo com a especificidade de cada conjunto de dados abordado nesta dissertação. A definição desse hiperparâmetro é feita de forma que se garanta a convergência da minimização do erro médio quadrático. Para três conjuntos de dados (Artificial, Câncer de Mama e Ionosfera), o número de épocas de treinamento foi definido como 1500. O conjunto *Wall-following* teve o valor N_e definido como 100 e o conjunto Coluna Vertebral foi usado N_e igual a 1000.

5.3 Função Kernel e seus hiperparâmetro

Para as simulações da técnica KPCA, as função de kernel adotada foi escolhida com base em uma estudo comparativo, baseando-se nas acurácias dos classificadores sugeridos para cada kernel testado. A função kernel escolhida foi o kernel log com o parâmetro $\sigma = 10$, pois foi a combinação que apresentou melhores resultados. Outro ponto positivo a ser observado é que o método KPCA, associada a função kernel log, é pouco sensível à variações do valor do hiperparâmetro σ.

As funções de kernel testadas foram: (i) linear, (ii) RBF e (iii) log. O hiperparâmetro da função do kernel RBF e da função do kernel log foram definidas através da técnica de busca em grade, respectivamente $\sigma = 10$ e $\sigma = 5$. O método KPCA associado ao kernel linear torna o método KPCA equivalente ao método PCA.

5.4 Critérios de Avaliação

Um vez que é obtido o número de neurônios sugerido (q^*) , então é usada novamente a estratégia *holdout* em cada rodada de trenamento/teste que é levada a cabo utilizando-se a nova arquitetura sugerida pelo método. Nos casos envolvendo problemas de classificação, são calculadas as médias de acerto para cada uma das 100 rodadas. Como critério de avaliação, usou-se o número de hiperparâmetros (N_c) da arquitetura, a taxa de acerto médio ($T_m(\%)$), a média da soma dos erros quadráticos médios (ε_m) e desvio padrão da taxa de acerto médio. Já para o problema de regressão, utilizou-se como critério de avaliação o número de hiperparâmetros (N_c) da arquitetura e a média da soma dos erros quadráticos médios (ε_m).

5.5 Resumo do Capítulo

Neste capítulo foram apresentadas as configurações dos hiperparâmetros da rede MLP, a metodologia de treinamento, a definição da função kernel utilizada e de seu hiperparâme-

tro. Também foi tópico deste capítulo a apresentação dos bancos de dados que foram usados para testar os métodos propostos. No próximo capítulo, serão reportados os resultados numéricos obtidos nas simulações descritas.

6 RESULTADOS E DISCUSSÕES

Neste capítulo são apresentados os resultados numéricos resultantes de uma comparação de desempenho das técnicas baseadas em PCA e KPCA para estimação do número de neurônios ocultos a partir das matrizes de estado. Também são apresentadas as discussões dos resultados para cada conjunto de dados, descritos no Capítulo 5.

Os resultados numéricos para as simulações são apresentados em tabelas divididas para cada conjunto de dados em que os métodos foram aplicados. Objetivando um estudo comparativo entre as técnicas lineares e as não-lineares, as tabelas apresentam as acurácias para as duas abordagens. A quantidade q_0 de neurônios na camada oculta da arquitetural inicial é definida previamente para cada exemplo; essa informação consta nas tabelas de resultados.

Como dito no Capítulo 5, os procedimentos de treinamento e teste foram rodados 100 vezes para cada conjunto de dados, sendo N_c a quantidade total de pesos da arquitetura, $T_m(\%)$ a taxa de acerto médio e ε_m a média das somas dos erros quadráticos médios medidos para os dados de teste ao longo das 100 rodadas de treinamento/teste.

6.1 Prova de Conceito

São apresentados na Tabela 1 os resultados obtidos nas simulações envolvendo o conjunto de dados artificial. Para o conjunto artificial, a arquitetura inicial foi definida como MLP$(2,10,2)$.

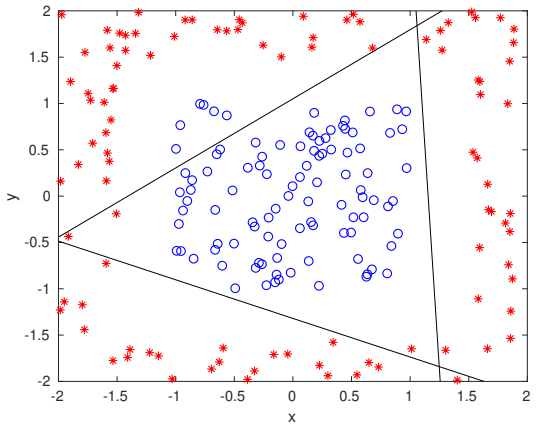
Topologia	**Matriz de Estado**	q_0	q^*	N_c	$T_m(\%) \pm$ desvio padrão	ε_m
MLP	-	10	-	52	99,05 $\pm$ 2,99	0
MLP-PCA	$\mathbf{Y}^{(h)}$	10	3	17	91,90 $\pm$ 3,94	0,33
	$\mathbf{E}^{-(h)}$	10	1	7	69,92 $\pm$ 8,26	0,86
	$\boldsymbol{\Delta}^{(h)}$	10	3	17	91,22 $\pm$ 5,54	0,58
MLP-KPCA	$\mathbf{Y}^{(h)}$	10	4	22	96,38 $\pm$ 4,68	0
kernel log	$\mathbf{E}^{-(h)}$	10	2	12	81,03 $\pm$ 6,01	0,61
$\sigma = 10$	$\boldsymbol{\Delta}^{(h)}$	10	2	12	80,42 $\pm$ 6,79	0,66

Tabela 1 – Resultados para o conjunto de dados artificiais.

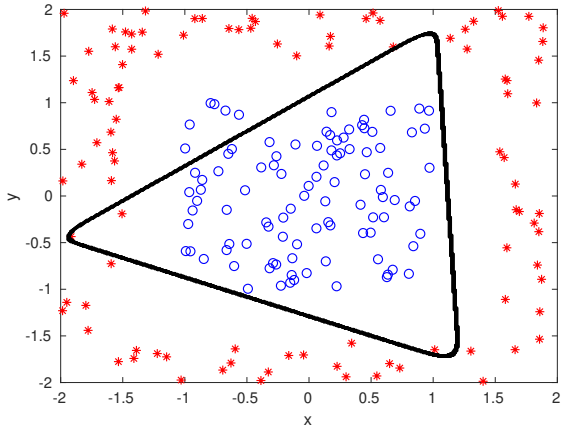
A aplicação da técnica baseada em PCA à matriz $\mathbf{E}^{(h)}$ resultou na sugestão de um único neurônio na camada oculta; porém, sabe-se previamente pela natureza não-linear dos dados, que este número de neurônios na camada oculta não resolve o problema. Já a aplicação da

técnica KPCA à matriz $\mathbf{Y}^{(h)}$ resultou em uma sugestão de que o problema poderia ser resolvido com elevada taxa de acerto com 4 neurônios na camada oculta. Esse resultado é bastante coerente, pois as duas classes podem ser facilmente separadas pela combinação de 4 retas associadas aos 4 neurônios ocultos, conforme Figura 4.

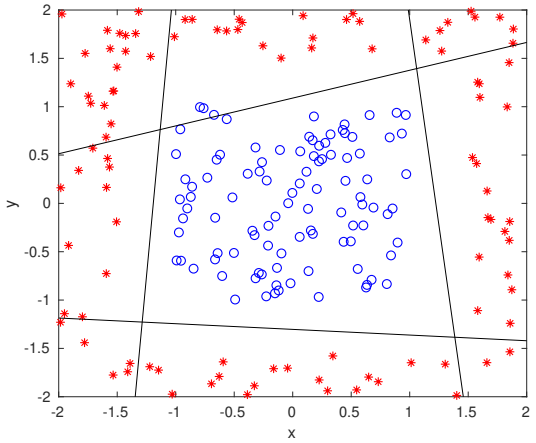
A aplicação de PCA a duas das matrizes de estado da rede MLP(2, 10, 2) resultou na sugestão de 3 neurônios ocultos. Embora a arquitetura sugerida tenha menor número de parâmetros que a topologia final sugerida pela aplicação de KPCA, a taxa de acerto final é bem inferior. Deste modo, a melhor topologia final é aquela sugerida pelo método KPCA, com 4 neurônios ocultos.



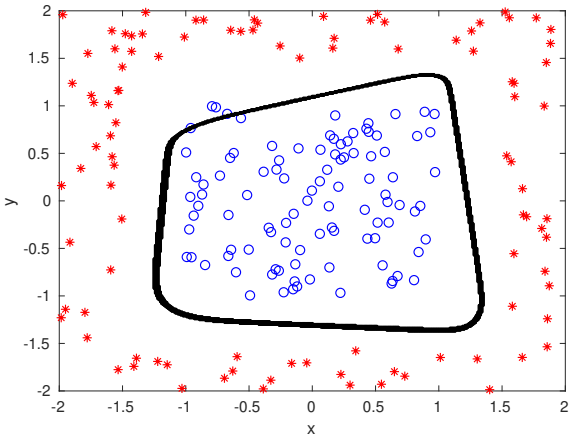
(a) Retas que compõem a superfície de decisão ($q^* = 3$).



(b) Superfície de decisão ($q^* = 3$).



(c) Retas que compõem a superfície de decisão ($q^* = 4$).



(d) Superfície de decisão ($q^* = 4$).

Figura 4 – Visão em duas dimensões das retas que compõem a superfície de decisão, bem como a própria superfície de decisão, gerada pela rede MLP com $q^* = 3$ e $q^* = 4$ neurônios na camada oculta, respectivamente.

6.2 Resultados - Classificação de Padrões

As subseções seguintes apresentam os resultados obtidos para rede MLP abordando os problemas de classificação de padrões referentes aos conjuntos de dados coluna vertebral, câncer de mama, ionosfera e duas variantes do conjunto *wall-following*. As duas variantes do conjunto *wall-following* voltadas para classificação foram chamadas de *Robot2Sensors*-classificação e *Robot4Sensors*-classificação.

6.2.1 Resultados - Coluna Vertebral

Para o conjunto coluna vertebral, foi definida uma arquitetura inicial MLP$(6,24,3)$. A Tabela 2 apresenta também as estimativas para as heurísticas apresentadas no capítulo 2, objetivando um estudo comparativo entre os principais métodos de estimação do número de neurônios ocultos. Além disso, são apresentados resultados para três variantes do método KPCA, em que variou-se a função kernel adotada.

Pode-se ver pela Tabela 2 que os resultados gerados pela aplicação da topologia MLP-KPCA, usando kernel log, são superiores (em termos de taxas médias) em comparação as outras topologias apresentadas nesta tabela.

Topologia	**Matriz de Estado**	q_0	q^*	N_c	$T_m(\%) \pm$ desvio padrão	ε_m
-	-	-	24	243	82.97 ± 4.41	0.65
MLP-PCA	$\mathbf{Y}^{(h)}$	24	9	93	82.34 ± 4.27	0.65
	$\mathbf{E}^{(h)}$	24	1	13	76.69 ± 5.08	0.62
	$\boldsymbol{\Delta}^{(h)}$	24	1	13	77.98 ± 5.13	0.60
MLP-KPCA	$\mathbf{Y}^{(h)}$	24	9	93	82.23 ± 4.02	0.67
Kernel Linear	$\mathbf{E}^{(h)}$	24	1	13	77.21 ± 4.17	0.60
	$\boldsymbol{\Delta}^{(h)}$	24	1	13	77.81 ± 4.76	0.59
MLP-KPCA	$\mathbf{Y}^{(h)}$	24	11	113	82.53 ± 4.70	0.67
Kernel Log	$\mathbf{E}^{(h)}$	24	2	23	84.29 ± 3.86	0.49
$\sigma = 10$	$\boldsymbol{\Delta}^{(h)}$	24	2	23	84.48 ± 4.63	0.49
MLP-KPCA	$\mathbf{Y}^{(h)}$	24	14	143	82.45 ± 4.10	0.67
Kernel RBF	$\mathbf{E}^{(h)}$	24	1	13	77.60 ± 5.48	0.60
$\sigma = 5$	$\boldsymbol{\Delta}^{(h)}$	24	1	13	77.81 ± 5.31	0.60
MLP - Valor Médio	-	-	5	53	83.79 ± 4.42	0.53
MLP - Raiz Quadrada	-	-	5	53	83.10 ± 4.60	0.56
MLP - Kolmogorov	-	-	13	133	81.11 ± 4.27	0.72

Tabela 2 – Resultados para o conjunto coluna vertebral.

Vale destacar aqui o resultado obtido com a arquitetura estimada pelo método KPCA, usando kernel *log* com $\sigma = 10$, aplicado à matriz de estado $\mathbf{\Delta}^{(h)}$, o qual produziu o classificador com arquitetura final MLP$(6,2,3)$ com acurácia de $84,48\% \pm 4,63$.

O melhor resultado pela aplicação do método baseado em PCA levou a uma arquitetura com $q^* = 9$ neurônios ocultos, usando a matriz de estado $\mathbf{Y}^{(h)}$. O mesmo resultado foi obtido com a aplicação do método KPCA usando o kernel linear.

Dentre as heurísticas testadas, a que apresentou melhor resultado foi o método do valor médio. O calculo do número de neurônios na camada oculta por este método deu $q^* = 5$ neurônios ocultos.

6.2.2 Resultados - Câncer de Mama

Para o conjunto câncer de mama, foi definido uma arquitetura neural inicial como MLP$(31,50,2)$. O número de neurônios na camada oculta sugeridos pelas técnicas abordadas aplicadas ao conjunto câncer de mama podem ser conferidos na Tabela 3. Em duas das três aplicações possíveis de KPCA a esse conjunto houve o indicativo de que o problema poderia ser solucionado apenas $q^* = 1$ neurônio na camada oculta e, mesmo assim, manter a taxa de acerto em níveis bem elevados. Uma interpretação plausível para este resultado é a de que há evidências para se afirmar que os dados do conjunto Câncer de Mama são linearmente separáveis. Este fato pode ser verificado pelas altas taxas de acerto obtidas, mesmo com apenas um neurônio oculto.

Topologia	**Matriz de Estado**	q_0	q^*	N_c	$T_m(\%) \pm$ desvio padrão	ε_m
MLP	-	50	-	1702	$97,21 \pm 1,47$	0,17
MLP-PCA	$\mathbf{Y}^{(h)}$	50	7	240	$96,75 \pm 1,47$	0,12
	$\mathbf{E}^{-(h)}$	50	1	36	$92,88 \pm 15,18$	0,03
	$\boldsymbol{\Delta}^{(h)}$	50	2	70	$96,89 \pm 1,43$	0,16
MLP-KPCA	$\mathbf{Y}^{(h)}$	50	9	308	$96,60 \pm 1,77$	0,23
Kernel Log	$\mathbf{E}^{-(h)}$	50	1	36	$95,65 \pm 10,02$	0,08
$\sigma = 10$	$\boldsymbol{\Delta}^{(h)}$	50	1	36	$97,00 \pm 1,47$	0,07

Tabela 3 – Resultados para o conjunto câncer de mama.

Outro ponto digno de nota é que as taxas de acerto médio foram equivalentes para as abordagens baseadas em PCA e KPCA. Um classificador perceptron simples aplicado a este conjunto de dados obteve taxas de acerto da mesma ordem de grandeza, confirmando a sugestão de que o problema pode ser tratado adequadamente por um classificador linear.

6.2.3 Resultados - Conjunto Ionosfera

Para o conjunto ionosfera, a arquitetura inicial foi definida como MLP$(33,28,2)$. Pode-se ver, pela análise da Tabela 4, que arquitetura resultante da aplicação do método baseado em PCA à matriz de estado $\mathbf{Y}^{(h)}$ possui $q^* = 13$ neurônios ocultos, gerando uma a melhor acurácia para o método baseado em PCA. Por outro lado, a aplicação da técnica proposta baseada em KPCA resultou em arquiteturas com mesmo desempenho, mas com um número bem inferior de neurônios ocultos $q^* = 2$. Este resultado ilustra bem a habilidade da técnica proposta em capturar correlações não-lineares. Para o exemplo em questão, o método baseado em PCA não foi capaz de capturar tais relações. Pode-se concluir que o método baseado em KPCA otimiza a acurácia ao mesmo tempo que mantêm a arquitetura compacta.

Topologia	**Matriz de Estado**	q_0	q^*	N_c	$T_m(\%) \pm$ desvio padrão	ε_m
MLP	-	28	-	1010	$89,87 \pm 3,89$	0,13
MLP-PCA	$\mathbf{Y}^{(h)}$	28	13	470	$89,76 \pm 3,96$	0,14
	$\mathbf{E}^{-(h)}$	28	1	38	$86,76 \pm 4,18$	0,06
	$\boldsymbol{\Delta}^{(h)}$	28	1	38	$85,85 \pm 3,69$	0,12
MLP-KPCA	$\mathbf{Y}^{(h)}$	28	14	506	$90,18 \pm 3,48$	0,11
Kernel Log	$\mathbf{E}^{-(h)}$	28	2	74	$89,45 \pm 3,59$	0,10
$\sigma = 10$	$\boldsymbol{\Delta}^{(h)}$	28	2	74	$89,51 \pm 3,45$	0,10

Tabela 4 – Resultados para o conjunto ionosfera.

6.2.4 Resultados - Conjunto Wall-Following

Para esse conjunto foram realizados testes usando-se as duas variantes do conjunto de dados; a primeira usando apenas dados para duas regiões de sensores e a segunda usando-se as quatro regiões de sensores. As duas variantes do conjunto *wall-following* voltadas para classificação foram chamadas de *Robot2Sensors*-classificação e *Robot*4*Sensors*-classificação, cujos os resultados estão expostos nas Tabelas 5 e 6, respectivamente.

Para o conjunto *Robot2Sensors*-classificação, foi definido uma arquitetura neural inicial como MLP$(2,30,4)$, já para a variantes *Robot*4*Sensors*-classificação a arquitetura inicial foi definida como MLP$(4,30,4)$.

Topologia	**Matriz de Estado**	q_0	q^*	N_c	$T_m(\%) \pm$ desvio padrão	ε_m
MLP	-	30	-	214	$97,51 \pm 0,65$	0,04
MLP-PCA	$\mathbf{Y}^{(h)}$	30	3	25	$93,75 \pm 3,94$	0,07
	$\mathbf{E}^{-(h)}$	30	1	11	$78,00 \pm 1,33$	0,16
	$\boldsymbol{\Delta}^{(h)}$	30	4	32	$95,80 \pm 2,04$	0,06
MLP-KPCA	$\mathbf{Y}^{(h)}$	30	4	32	$95,72 \pm 2,00$	0,06
Kernel Log	$\mathbf{E}^{-(h)}$	30	2	18	$84,36 \pm 3,25$	0,12
$\sigma = 10$	$\boldsymbol{\Delta}^{(h)}$	30	5	39	$96,75 \pm 1,39$	0,05

Tabela 5 – Resultados para o conjunto *Robot2Sensors*-classificação.

Topologia	**Matriz de Estado**	q_0	q^*	N_c	$T_m(\%) \pm$ desvio padrão	ε_m
MLP	-	30	-	274	$97,32 \pm 0,99$	0,04
MLP-PCA	$\mathbf{Y}^{(h)}$	30	4	40	$95,09 \pm 2,20$	0,06
	$\mathbf{E}^{-(h)}$	30	1	13	$78,13 \pm 1,08$	0,16
	$\boldsymbol{\Delta}^{(h)}$	30	4	40	$94,86 \pm 2,00$	0,06
MLP-KPCA	$\mathbf{Y}^{(h)}$	30	4	40	$94,66 \pm 1,99$	0,07
Kernel Log	$\mathbf{E}^{-(h)}$	30	2	22	$83,61 \pm 2,82$	0,12
$\sigma = 10$	$\boldsymbol{\Delta}^{(h)}$	30	5	49	$96,53 \pm 1,37$	0,05

Tabela 6 – Resultados para o conjunto *Robot4Sensors*-classificação.

A sugestão que apresentou maior acurácia, nas duas variantes, foi aquela devido à aplicação de KPCA sobre a matriz de estado $\boldsymbol{\Delta}^{(h)}$, a qual sugeriu $q^* = 5$ neurônios na camada oculta. Esse resultado pode ser comparado com aquele reportado em Freire (2009), que utilizando a técnica de *busca exaustiva* obteve um valor muito próximo de 5 neurônios ocultos como melhor resultado para as duas variantes abordadas aqui, a saber MLP$(2,6,4)$ com acurácia de 96,82%, para a variantes com 2 sensores e a MLP$(4,6,4)$ para a variante com 4 sensores, o qual teve acurácia de 97,04%.

6.3 Aproximação de Função

A seguir são apresentados os resultados obtidos para a rede MLP, quando aplicadas em um problema de aproximação de função com os dados do robô móvel chamado de Scitos G5. Para este conjunto, o vetor de atributos contém as leituras dos sensores de ultrassom (duas ou quatro entradas) e as saídas/alvos são as velocidades de translação (v_t) e de rotação (v_r) do robô. As duas variantes do conjunto *wall-following* voltadas para regressão foram chamadas

de *Robot2Sensors*-regressão e *Robot4Sensors*-regressão, cujos os resultados são reportados nas Tabelas 7 e 8, respectivamente.

Para o conjunto *Robot2Sensors*-regressão, foi definido uma arquitetura neural inicial como MLP$(2,30,2)$, já para a variantes *Robot4Sensors*-regressão a arquitetura inicial foi definida como MLP$(4,30,2)$.

Topologia	**Matriz de Estado**	q_0	q^*	N_c	$\varepsilon_m \pm$ desvio padrão
MLP	-	30	-	152	$0,30 \pm 0,01$
MLP-PCA	$\mathbf{Y}^{(h)}$	30	3	17	$0,30 \pm 0,01$
	$\mathbf{E}^{(h)}$	30	1	7	$0,32 \pm 0,01$
	$\boldsymbol{\Delta}^{(h)}$	30	3	17	$0,30 \pm 0,01$
MLP-KPCA	$\mathbf{Y}^{(h)}$	30	3	17	$0,30 \pm 0,01$
Kernel Log	$\mathbf{E}^{(h)}$	30	2	12	$0,30 \pm 0,01$
$\sigma = 10$	$\boldsymbol{\Delta}^{(h)}$	30	3	17	$0,30 \pm 0,01$

Tabela 7 – Resultados para o conjunto *robot2Sensors*-regressão.

Topologia	**Matriz de Estado**	q_0	q^*	N_c	$\varepsilon_m \pm$ desvio padrão
MLP	-	30	-	212	$0,28 \pm 0,01$
MLP-PCA	$\mathbf{Y}^{(h)}$	30	3	23	$0,30 \pm 0,01$
	$\mathbf{E}^{(h)}$	30	1	9	$0,32 \pm 0,01$
	$\boldsymbol{\Delta}^{(h)}$	30	3	23	$0,30 \pm 0,01$
MLP-KPCA	$\mathbf{Y}^{(h)}$	30	3	23	$0,30 \pm 0,01$
Kernel Log	$\mathbf{E}^{(h)}$	30	2	16	$0,30 \pm 0,01$
$\sigma = 10$	$\boldsymbol{\Delta}^{(h)}$	30	3	23	$0,30 \pm 0,01$

Tabela 8 – Resultados para o conjunto *robot4Sensors*-regressão.

Em problemas relacionados a aproximação de funções, o interesse está na redução da média do MSE[6] (ε_m) ao longo das 100 rodadas de treinamento/teste e, assim como em problemas de classificação, deseja-se reduzir o número de neurônios da camada oculta a uma quantidade mínima possível, mantendo o mesmo desempenho em termos de MSE. Isto posto, a arquitetura com $q^* = 2$ neurônios na camada oculta foi a arquitetura com menor número de neurônios na camada oculta, tento mentido o menor ε_m entre as arquiteturas sugeridas.

Vale lembrar que o problema de aproximação de funções está estimando 2 saídas ao mesmo tempo, que são as velocidades rotacional e translacional. Este procedimento torna

[6] *Mean squared error*.

o processo de aproximação mais complexo, visto que as duas velocidades estão acopladas de forma não-linear em termos mecânicos.

6.4 Resumo do Capítulo

Neste capítulo, foram apresentados os resultados numéricos das simulações provenientes da aplicação do método proposto para estimação do número de neurônios ocultos da rede MLP usando KPCA. A título de comparação, os resultados gerados pela aplicação de uma versão linear do método proposto são também reportados. Problemas de classificação de padrões e de aproximação de funções são estudados e os resultados discutidos. Também foram apresentadas discussões dos resultados para cada conjunto de dados. O próximo capítulo apresenta as conclusões que podem ser tiradas a partir dos resultados discutidos no capítulo atual, além de trazer um tópico sobre sugestões de trabalhos futuros.

7 CONCLUSÕES E TRABALHOS FUTUROS

Neste capítulo, são apresentadas as conclusões que podem ser tiradas a partir dos resultados apresentados no Capítulo 6, além de trazer um tópico sobre sugestões de trabalhos futuros.

7.1 Principais Conclusões

Nesta dissertação, foi desenvolvida uma proposta de estimação do número de neurônios ocultos da rede MLP, de uma camada oculta, que estende a técnica proposta por Santos, Barreto e Medeiros (2010). Esta é baseada na aplicação de PCA às matrizes de estado da rede MLP de uma camada oculta, enquanto a técnica proposta usa uma versão não-linear de PCA, ou seja, kernel PCA.

7.2 Trabalhos Futuros

Apesar do amplo estudo comparativo levado a cabo ao longo do desenvolvimento desta dissertação, restam ainda alguns pontos importantes a serem trabalhados visando o pleno entendimento da técnica proposta. A seguir são listadas algumas linhas possíveis de desenvolvimento.

- Aplicar a técnica não-linear para estimação da quantidade de neurônios em múltiplas camadas ocultas.
- Avaliar a possibilidade de usar outras técnicas de compressão da informação ou redução de dimensionalidade para estimar o número de neurônios ocultos da rede MLP, tais como PCA local (KAMBHATLA; LEEN, 1997) e Curvas Principais (HASTIE; STUETZLE, 1989).
- Desenvolver técnicas baseadas em outros métodos de remoção de informação redundante aplicadas às variáveis de estado, a exemplo do Local-PCA.
- Desenvolver estratégias que permitam a aplicação eficiente da técnica proposta em problemas de *big data*.
- Desenvolver estratégias incrementais que permitam a aplicação eficiente da técnica proposta em problemas de aprendizado online ou sequencial.

REFERÊNCIAS

ALAKKARI, S.; DINGLIANA, J. Principal component analysis techniques for visualization of volumetric data. In: NAIK, G. R. (Ed.). **Advances in Principal Component Analysis: Research and Development**. Singapore: Springer, 2018. p. 99–120. ISBN 978-981-10-6704-4. Disponível em: <https://doi.org/10.1007/978-981-10-6704-4_5>. Acesso em: 20 dez. 2019.

BERTHONNAUD, E.; DIMNET, J.; ROUSSOULY, P.; LABELLE, H. Analysis of the sagittal balance of the spine and pelvis using shape and orientation parameters. **Journal of Spinal Disorders & Techniques**, v. 18, n. 1, p. 40–47, 2005.

BISHOP, C. M. **Neural Networks for Pattern Recognition**. Cambridge: Oxford University Press, 2005.

BISHOP, C. M. **Pattern recognition and machine learning**. New York: Springer, 2006.

BOUGHORBEL, S.; TAREL, J.-P.; BOUJEMAA, N. Conditionally positive definite kernels for SVM based image recognition. In: IEEE INTERNATIONAL CONFERENCE ON MULTIMEDIA AND EXPO (ICME), 2005, Amsterdam. **Proceedings [...]**. Amsterdam: IEEE, 2005. p. 113–116.

DATTA, A.; GHOSH, S.; GHOSH, A. Pca, kernel pca and dimensionality reduction in hyperspectral images. In: NAIK, G. R. (Ed.). **Advances in Principal Component Analysis: Research and Development**. Singapore: Springer Singapore, 2018. ISBN 978-981-10-6704-4. Disponível em: <https://doi.org/10.1007/978-981-10-6704-4_2>. Acesso em: 20 dez. 2019.

FREIRE, A. L. **A dimensão temporal no projeto de classificadores de padrões para navegação de robôs móveis: um estudo de caso**. Dissertação (Mestrado em Engenharia de Teleinformática) — Centro de Tecnologia, Universidade Federal do Ceará - UFC, Ceará, 2009.

FREIRE, A. L.; BARRETO, G. A.; VELOSO, M.; VARELA, A. T. Short-term memory mechanisms in neural network learning of robot navigation tasks: A case study. In: LATIN AMERICAN ROBOTICS SYMPOSIUM (LARS 2009), 6th., New York. **Proceedings[...]**. New York: IEEE, 2009. p. 1–6.

GOLOVKO, V. A.; VAITSEKHOVICH, L. U.; KOCHURKO, P. A.; RUBANAU, U. S. Dimensionality reduction and attack recognition using neural networks approaches. In: INTERNATIONAL JOINT CONFERENCE ON NEURAL NETWORKS (IJCNN'2007), Orlando. **Proceedings[...]**. Orlando: IEEE, 2007. p. 2734–2739.

GOOD, R. P.; KOST, D.; CHERRY, G. A. Introducing a unified pca algorithm for model size reduction. **IEEE Transactions on Semiconductor Manufacturing**, Raleigh, IEEE, v. 23, n. 2, p. 201–209, 2010.

HASTIE, T.; STUETZLE, W. Principal curves. **Journal of the American Statistical Association**, Philadelphia, Taylor & Francis, v. 84, n. 406, p. 502–516, 1989.

HAYKIN, S. **Redes neurais: princípios e prática**. 2nd. ed. Porto Alegre: Pearson Prentice Hall, 2001.

HOFMANN, T.; SCHÖLKOPF, B.; SMOLA, A. J. Kernel methods in machine learning. **The annals of statistics**, New York, JSTOR, p. 1171–1220, 2008.

HORNIK, K. Approximation capabilities of multilayer feedforward networks. **Neural networks**, Columbus, Elsevier, v. 4, n. 2, p. 251–257, 1991.

KAMBHATLA, N.; LEEN, T. K. Dimension reduction by local principal component analysis. **Neural computation**, Cambridge, MIT Press, v. 9, n. 7, p. 1493–1516, 1997.

LI, M.; XING, Y.; LUO, R. A novel feature extraction approach based on pca and improved fisher score applied in speaker verification. In: INTERNATIONAL CONFERENCE ON AUDIO, LANGUAGE AND IMAGE PROCESSING (ICALIP'2008), 2008, Shanghai. **Proceedings[...]**. Shanghai: IEEE, 2008. p. 56–60.

LUO, Y.; XIONG, S.; WANG, S. A pca based unsupervised feature selection algorithm. In: SECOND INTERNATIONAL CONFERENCE ON GENETIC AND EVOLUTIONARY COMPUTING, 2., 2008, Hubei. **Proceedings[...]**. Hubei: IEEE, 2008. p. 299–302.

MEDEIROS, C. M. S.; BARRETO, G. A. Pruning the multilayer perceptron through the correlation of backpropagated errors. In: SEVENTH INTERNATIONAL CONFERENCE ON INTELLIGENT SYSTEMS DESIGN AND APPLICATIONS (ISDA 2007), 7., 2007, Rio de Janeiro. **Proceedings[...]**. Rio de Janeiro: IEEE, 2007. p. 64–69.

MEDEIROS, C. M. S.; BARRETO, G. A. A novel weight pruning method for mlp classifiers based on the maxcore principle. **Neural Computing and Applications**, London, Springer, v. 22, n. 1, p. 71–84, 2013. Disponível em: <https://doi.org/10.1007/s00521-011-0748-6>. Acesso em: 5 jun. 2020.

MERCER, J. Xvi. functions of positive and negative type, and their connection the theory of integral equations. **Philosophical transactions of the royal society of London. Series A, containing papers of a mathematical or physical character**, London, The Royal Society London, v. 209, n. 441-458, p. 415–446, 1909.

MIKA, S.; SCHÖLKOPF, B.; SMOLA, A. J.; MÜLLER, K.-R.; SCHOLZ, M.; RÄTSCH, G. Kernel pca and de-noising in feature spaces. In: ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS, 11., 1999, [S.l]. **Proceedings[...]**. [S.l]: MIT Press, 1999. p. 536–542.

NISE, N. S. **Control Systems Engineering, International Student Version**. New Jersey: John Wiley & Sons, Inc, 2010.

OSOWSKI, S.; MAJKOWSKI, A.; CICHOCKI, A. Robust pca neural networks for random noise reduction of the data. In: INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH AND SIGNAL PROCESSING (ICASSP'1997), 1997, Munich. **Proceedings[...]**. Munich: IEEE, 1997. v. 4, p. 3397–3400.

PRINCIPE, J. C.; EULIANO, N. R.; LEFEBVRE, W. C. **Neural and Adaptative Systems: Fundamentals Through Simulations**. 1st. ed. New York: John Wiley & Sons, Inc., 2000.

ROCHA-NETO, A. R.; BARRETO, G. A. On the application of ensembles of classifiers to the diagnosis of pathologies of the vertebral column: A comparative analysis. **IEEE Latin America Transactions**, New York, IEEE, v. 7, n. 7, p. 487–496, 2009.

RUMELHART, D.; HINTON, G.; WILLIAMS, R. Learning representations by back-propagating errors. **Nature**, London, Nature Publishing Group, v. 323, n. 6088, p. 533–566, 1986.

SANTOS, J. D. A. **Sobre a estimação do número de neurônios ocultos da rede mlp: uma nova técnica baseada em pca e svd**. Dissertação (Mestrado em Engenharia de Teleinformática) — Centro de Tecnologia, Universidade Federal do Ceará - UFC, Ceará, 2010.

SANTOS, J. D. A.; BARRETO, G. A.; MEDEIROS, C. M. S. Estimating the number of hidden neurons of the mlp using singular value decomposition and principal components analysis: a novel approach. In: IEEE ELEVENTH BRAZILIAN SYMPOSIUM ON NEURAL NETWORKS, 11., 2010, São Paulo. **Proceedings[...]**. São Paulo: IEEE, 2010. p. 19–24.

SCHMIDHUBER, J. Deep learning in neural networks: An overview. **Neural networks**, Columbus, Elsevier, v. 61, p. 85–117, 2015.

SCHÖLKOPF, B.; SMOLA, A.; MÜLLER, K.-R. Nonlinear component analysis as a kernel eigenvalue problem. **Neural computation**, Cambridge, MIT Press, v. 10, n. 5, p. 1299–1319, 1998.

SIGILLITO, V. G.; WING, S. P.; HUTTON, L. V.; BAKER, K. B. Classification of radar returns from the ionosphere using neural networks. **Johns Hopkins APL Technical Digest**, Laurel, Johns Hopkins, v. 10, n. 3, p. 262–266, 1989.

SOUZA, C. R. de. Kernel functions for machine learning applications. **crsouza**, 2010. Disponível em: <http://crsouza.com/2010/03/17/kernel-functions-for-machine-learning-applications/>. Acesso em: 05 jun. 2020.

TEOH, E. J.; TAN, K. C.; XIANG, C. Estimating the number of hidden neurons in a feedforward network using the singular value decomposition. **IEEE Transactions on Neural Networks**, New York, IEEE, v. 17, n. 6, p. 1623–1629, 2006.

WANG, Q. Kernel principal component analysis and its applications in face recognition and active shape models. **arXiv preprint arXiv:1207.3538**, New York, Cornell University, 2012.

WEIGEND, A. S.; RUMELHART, D. E. The effective dimension of the space of hidden units. In: 1991 IEEE INTERNATIONAL JOINT CONFERENCE ON NEURAL NETWORKS, 1991, Singapore. **Proceedings[...]**. Singapore: IEEE, 1991. p. 2069–2074.

WU, J.; WANG, J.; LIU, L. Feature extraction via kpca for classification of gait patterns. **Human movement science**, Columbus, Elsevier, v. 26, n. 3, p. 393–411, 2007.

YANG, M.-H. Kernel eigenfaces vs. kernel fisherfaces: Face recognition using kernel methods. In: FIFTH INTERNATIONAL CONFERENCE ON AUTOMATIC FACE AND GESTURE RECOGNITION, 5., 2002, Washington. **Proceedings[...]**. Washington: IEEE, 2002. v. 2, p. 215–220.

YU, Y.-D.; KANG, D.-S.; KIM, D. Color image compression based on vector quantization using pca and lebld. In: TENCON, IEEE REGION 10 INTERNATIONAL CONFERENCE, 1999, Cheju Island. **Proceedings[...]**. Cheju Island: IEEE, 1999. v. 2, p. 1259–1262.