



**UNIVERSIDADE FEDERAL DO CEARÁ**  
**CENTRO DE CIÊNCIAS**  
**DEPARTAMENTO DE MATEMÁTICA E ESTATÍSTICA APLICADA**  
**PROGRAMA DE PÓS-GRADUAÇÃO EM MODELAGEM E MÉTODOS**  
**QUANTITATIVOS**

**TARSIS LIMA DA SILVA**

**REDES NEURAS RANDOMIZADAS PARA DETERMINAÇÃO DA CURVA DE**  
**POTÊNCIA DE AEROGERADORES**

**FORTALEZA**

**2022**

TARSIS LIMA DA SILVA

REDES NEURAIAS RANDOMIZADAS PARA DETERMINAÇÃO DA CURVA DE  
POTÊNCIA DE AEROGERADORES

Dissertação apresentada ao Programa de Pós-Graduação em Modelagem e Métodos Quantitativos da Universidade Federal do Ceará, como requisito parcial à obtenção do título de Mestre em Modelagem e Métodos Quantitativos. Área de concentração: Inteligência Computacional e Otimização.

Orientador: Prof. Dr. Guilherme de Alencar Barreto.

FORTALEZA

2022

Dados Internacionais de Catalogação na Publicação  
Universidade Federal do Ceará  
Sistema de Bibliotecas

Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

---

S584r Silva, Tarsis Lima da.

Redes neurais randomizadas para determinação da curva de potência de aerogeradores / Tarsis Lima da Silva. – 2022.

60 f. : il. color.

Dissertação (mestrado) – Universidade Federal do Ceará, Centro de Ciências, Programa de Pós-Graduação em Modelagem e Métodos Quantitativos, Fortaleza, 2022.

Orientação: Prof. Dr. Guilherme de Alencar Barreto.

1. Curva de Potência. 2. Ajuste de curvas. 3. Geração eólica. 4. Redes neurais randomizadas. 5. Regressão não-linear. I. Título.

CDD 510

---

TARSIS LIMA DA SILVA

REDE NEURAIIS RANDOMIZADAS PARA A DETERMINAÇÃO DA CURVA DE  
POTÊNCIA DE AEROGERADORES

Dissertação apresentada ao Programa de Pós-Graduação em Modelagem e Métodos Quantitativos da Universidade Federal do Ceará, como requisito parcial à obtenção do título de Mestre em Modelagem e Métodos Quantitativos. Área de concentração: Inteligência Computacional e Otimização.

Aprovada em 29/07/2022.

BANCA EXAMINADORA

---

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

---

Prof. Dr. Ricardo Coelho Silva  
Universidade Federal do Ceará (UFC)

---

Prof. Dr. Marinho Gomes de Andrade Filho  
Universidade de São Paulo (USP)

Este trabalho é dedicado a meus pais, Francisco Gilson Almeida da Silva e Elizabeth Lima da Silva.

A meus familiares, amigos, professores e professoras que me acompanharam e acompanham na jornada da vida.

## AGRADECIMENTOS

Agradeço minha família e namorada, que tanto prezam por mim, e que sem apoio não teria chegado tão longe. Meus sinceros agradecimentos, pois o amor de vocês só me deixa mais forte, e deixa meus dias mais felizes. Obrigado mãe Elizabeth, meu pai Gilson, minha tia Luzimar e minha namorada Wennya, obrigado por tudo.

Ao Prof. Dr. Guilherme de Alencar Barreto, pela excelente orientação, que dedicou seu tempo à difícil tarefa de orientar um jovem pesquisador. Sua expertise no trato pessoal, do trabalho e sua capacidade colaborativa, ensinamentos e acessibilidade foram fundamentais para o meu bom desenvolvimento e do trabalho.

Aos professores participantes da banca examinadora Prof. Dr. Ricardo Coelho Silva, Prof. Dr. Marinho Gomes de Andrade Filho, Prof. Dr. Charles Casimiro Cavalcante e Prof. Dr. João Paulo Pordeus Gomes pelo tempo, pelas valiosas colaborações e sugestões.

Aos professores(as), pessoal do administrativo do departamento, que sempre zelaram pelo boa relação e bom trabalho. Em especial, a todos meus professores e professoras que contribuíram fortemente para minha formação, pois foi nesse mestrado que fui apresentado “à minha praia” enquanto pesquisador.

Agradeço aos e as colegas do curso que tanto contribuíram para a vivência acadêmica.

Agradeço as amigas e aos amigos mais próximos que me fazem um bem danado e deixam meus dias mais felizes.

Por fim, agradeço a Universidade Federal do Ceará (UFC) e a Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) que me proporcionaram essa grande experiência de formação profissional e também pessoal.

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Código de Financiamento 001.

“Ensinar não é transferir conhecimento, mas criar as possibilidades para a sua própria produção ou a sua construção.” (FREIRE, 1996, p.47).

## RESUMO

A modelagem da curva de potência de turbina eólicas (WTPC, na sigla em inglês) a partir de dados medidos é fundamental para prever a energia elétrica efetivamente gerada pelos parques eólicos. Regressão polinomial é geralmente a primeira escolha para modelagem da curva de potência devido à sua ampla disponibilidade em pacotes de software conhecidos, como MS-Excel (especificamente com o pacote Solver), mas existem várias alternativas poderosas baseadas, por exemplo, em redes neurais e algoritmos fuzzy. No que diz respeito ao paradigma neural, a redescoberta de uma classe de algoritmos supervisionados, genericamente chamados de redes neurais randomizadas, tem despertado o interesse de seu uso em aplicações práticas, devido principalmente ao rápido processo de treinamento. No entanto, a aplicação de redes randomizadas na modelagem de curvas de potência não é imediata e comumente apresenta soluções incorretas, aqui chamadas de soluções patológicas, apesar dos bons valores alcançados pelos índices de desempenho numéricos tradicionais, como o coeficiente de determinação  $R^2$ . Nesta dissertação, um estudo crítico abrangente é realizado a fim de avaliar a aplicação de redes randomizadas na modelagem de curvas de potência de aerogeradores. Serão avaliadas três arquiteturas, a saber, rede *random vector functional link* (RVFL), a rede *extreme learning machine* (ELM) e a rede *No-Propagation* (NoProp) em dois conjuntos de dados reais. Adaptações destas são propostas e avaliadas quanto à ocorrência de soluções patológicas. Os desempenhos das redes randomizadas acima mencionadas são comparados com o do modelo de regressão polinomial e com o da rede perceptron multicamadas (MLP, na sigla em inglês). O estudo realizado conclui que as redes randomizadas são viáveis no trato da tarefa de interesse e apresentam desempenho superior aos modelos supracitados, tanto em termos numéricos, quanto em rapidez na construção do modelo de regressão, sem geração de soluções patológicas.

**Palavras-chave:** curva de potência; ajuste de curva; geração eólica; redes neurais randomizadas; regressão não-linear; estimação de parâmetros.

## ABSTRACT

Modeling the wind turbine power curve (WTPC) from measured data is critical to predict the electricity actually generated by wind farms. Polynomial regression is usually the first choice for WTPC modeling due to its wide availability in well-known software packages, but there are several powerful alternatives based, for example, on neural networks and fuzzy algorithms. With regarding the neural paradigm, the rediscovery of a class of supervised algorithms, generically called randomized neural networks, has aroused the interest of its use in practical applications, mainly due to the fast training process. However, the application of random networks in the modeling of power curves is not immediate and commonly presents incorrect solutions, here called pathological solutions, despite the good values achieved by the standard performance numerical indices, such as determining the  $R^2$  coefficient. In this thesis, a comprehensive critical study is carried out in order to evaluate the feasibility of using randomized networks in power curve modeling. Three architectures will be evaluated, namely, Random Vector Functional Link (RVFL), Extreme Learning Machine (ELM) and No-Propagation Network (No-Prop) on two real data sets. The adaptations are proposed and evaluated for the occurrence of pathological solutions. The performances of the aforementioned randomized networks are compared with the polynomial regression model and with the MLP (Multilayer Perceptron) network. The study carried out concludes that randomized networks are viable in the treatment of task of interest and present superior performance to the aforementioned models, both in numerical performance and in speed in the construction of the regression model, without generating pathological solutions.

**Keywords:** power curve; curve fitting; wind generation; randomized neural networks; non-linear regression; estimation of parameters.

## LISTA DE ILUSTRAÇÕES

Figura 1	–	Esquema da passagem do fluxo do ar pela área $A$ varrida do aerogerador ...	21
Figura 2	–	Gráfico com típica forma da curva de potência de um aerogerador com controle de pitch .....	22
Figura 3	–	Gráfico da função sigmoide logística $P(v)$ e de sua derivada $P'(v)$ .....	24
Figura 4	–	Arquitetura típica de uma rede neural <i>feedforward</i> com uma única camada oculta .....	28
Figura 5	–	Gráficos de dispersão dos conjuntos de dados amostrais .....	40
Figura 6	–	Soluções gráficas do modelo polinomial .....	42
Figura 7	–	Soluções gráficas do modelo MLP .....	44
Figura 8	–	Típica solução patológica produzida pelo modelo $ELM(1,10,1)$ .....	46
Figura 9	–	Típica solução produzida pelo modelo NoProp .....	47
Figura 10	–	Típica solução patológica produzida pelo modelo RVFL com saída linear	49
Figura 11	–	Típica curva de potência produzida pelos seguintes modelos para os dados Delfos .....	54

## LISTA DE TABELAS

Tabela 1	–	Especificações do parque eólico dos WT's .....	39
Tabela 2	–	Índices de desempenho numérico para o modelo de regressão polinomial de grau $q$ – conjunto WT1 .....	41
Tabela 3	–	Índices de desempenho numérico para o modelo $MLP(1, q, 1)$ – conjunto WT1 .....	43
Tabela 4	–	Índices de desempenho numérico para o modelo $ELM(1, q, 1)$ – conjunto WT1 .....	45
Tabela 5	–	Índices de desempenho numérico para o modelo NoProp – conjunto WT1 .....	47
Tabela 6	–	Índices de desempenho numérico para o modelo RVFL (variantes) – conjunto WT1 .....	48
Tabela 7	–	Sumário dos melhores modelos e suas características com médias dos índices numéricos – conjunto WT1 .....	50
Tabela 8	–	Sumário dos melhores modelos e outras estatísticas tais como valores de mínimo, máximo e desvio padrão dos índices numéricos – conjunto WT1 .....	50
Tabela 9	–	Sumário com conjunto de treino (CT) e teste (CTT), e outras estatísticas tais como valores da média, mínimo, máximo e desvio padrão dos índices numéricos – conjunto WT1 .....	51
Tabela 10	–	Sumário do melhores modelos e suas características – conjunto Delfos .....	52
Tabela 11	–	Sumário dos melhores modelos e suas características com mínimo, máximo e desvio padrão dos índices numéricos – conjunto Delfos .....	52
Tabela 12	–	Sumário com conjunto de treino (CT) e teste (CTT), e outras estatísticas tais como valores da média, mínimo, máximo e desvio padrão dos índices numéricos – conjunto Delfos .....	52

## LISTA DE ABREVIATURAS E SIGLAS

BIP	<i>Batch Intrinsic Plasticity</i>
CPA	Curva de Potência de Aerogerador
ELM	<i>Extreme Learning Machine</i>
IEC	International Electrotechnical Commission
LMS	<i>Least Mean Squares</i>
MLP	<i>Multilayer Perceptron</i>
MLP-linear	Perceptron Multicamadas com Saída Linear
MLP-logsig	Perceptron Multicamadas com Saída Sigmoidal
MQO	Mínimos Quadrados Ordinários
NLMS	LMS Normalizada
NoProp	<i>No-Propagation</i> com Saída Linear
NoProp-logsig	<i>No-Propagation</i> com Saída Sigmoidal
RBF	<i>Radial-basis Function</i>
RMSE	<i>Root Mean Square Error</i>
RNRs	Redes Neurais Randomizadas
RVFL	<i>Random Vector Functional Link</i>
RVFL-LMS	<i>Random Vector Functional Link</i> com Saída Linear
RVFL-logsig	<i>Random Vector Functional Link</i> com Saída Sigmoidal
SLFN	<i>Single-layer Feedforward Network</i>
TC	Tempo Computacional
WTPC	<i>Wind Turbine Power Curve</i>

## LISTA DE SÍMBOLOS

$R^2$	Coeficiente de determinação.
$P$	Potência.
$v$	Velocidade do vento.
$A$	Área varrida pelas pás (hélices) do aerogerador de eixo horizontal.
$\rho$	Densidade do ar.
$t$	Medida de tempo.
$E_c$	Energia cinética do movimento.
$M$	Massa
$\Delta E_c$	Variação da energia cinética.
$\Delta t$	Variação da medida de tempo.
$P_d$	Potência eólica disponível.
$P_a$	Potência absorvida aproveitada pelo movimento do rotor.
$C_p$	Coeficiente de potência.
$\alpha$	Eficiência do rotor.
$P_{eg}$	Potência elétrica gerada.
$v_{ci}$	Velocidade de corte inferior.
$v_n$	Velocidade nominal.
$v_{cs}$	Velocidade de corte superior.
$P_{max}$	Potência máxima.
$v_i$	A $i$ -ésima velocidade do vento observada.
$P_i$	A $i$ -ésima potência observada.
$P(v)$	Potência em função da velocidade do vento.
$P'(v)$	Derivada de $P(v)$ em relação a velocidade do vento.
$\exp(-v)$	Exponencial de $-v$ .
$N$	Quantidade de padrões entrada-saída ou pares ordenados de um conjunto de dados.
$\mathbf{x}_n$	A $n$ -ésima entrada de dimensão $p$ .
$d_n$	A $n$ -ésima saída alvo de dimensão 1.
$\mathbf{X}_{p \times N_1}$	Matriz com $N_1$ vetores de entrada com dimensão $p$ organizados ao longo das colunas – idem $\mathbf{X}$ .

$\mathbf{d}_{N_1 \times 1}$	Vetor com $N_1$ saídas alvo correspondentes aos vetores da matriz $\mathbf{X}_{p \times N_1}$ – idem $\mathbf{d}$ .
$T$	Sobrescrito para denotar a transposição da matriz ou vetor.
$\phi$	Rede neural artificial qualquer bem como função de ativação sigmoide logística com derivada $\phi'$ .
$q$	Quantidade de neurônios ocultos e grau do polinômio.
$y_n$	Saída genérica do polinômio regressor e da rede neural artificial $\phi$ com entrada $\mathbf{x}_n$ .
$\delta_n$	Variável aleatória que denota o erro de aproximação pelo modelo polinomial com média 0 e variância constante $\sigma_\delta^2$ .
$w_j$	O $j$ -ésimo componente do vetor $\mathbf{w}$ .
$\mathbf{w}$	Vetor de coeficientes de acordo com o modelo utilizado.
$\mathbf{y}$	Vetor de saída genérica de acordo com o modelo utilizado.
$\mathbf{Z}$	Matriz de entrada transformada de acordo com o modelo utilizado.
$\mathbf{z}_n$	Vetor de $\mathbf{Z}$ transformado.
$\boldsymbol{\delta}$	Vetor de erros de aproximação do modelo polinomial.
$J$	Função custo com objetivo de minimizar norma quadrática dos erros de aproximação.
$\frac{\partial J(\mathbf{w})}{\partial \mathbf{w}}$	Primeira derivada da função custo $J$ em relação a $\mathbf{w}$ .
$\frac{\partial^2 J(\mathbf{w})}{\partial \mathbf{w}^2}$	Segunda derivada da função custo de $J$ em relação a $\mathbf{w}$ .
$r$	Número de neurônios de saída da rede MLP.
$u$	Argumento da função de ativação $\phi$ .
$u_n$	O $n$ -ésimo argumento da função de ativação $\phi$ .
$w_{ij}$	Peso conectando a $j$ -ésima unidade de entrada com o $i$ -ésimo neurônio oculto na rede MLP.
$w_{i0}$	Limiar para o $i$ -ésimo neurônio oculto na rede MLP.
$m_{ki}$	Peso conectando o $i$ -ésimo neurônio oculto ao $k$ -ésimo neurônio de saída na rede MLP.
$\theta_k$	Limiar do $k$ -ésimo neurônio de saída na rede MLP.
$L$	Número de parâmetros ajustáveis da rede MLP.
$y_n^{(1)}$	A $n$ -ésima saída linear da rede RVLFF.

$y_n^{(2)}$	A $n$ -ésima saída não-linear das redes RVFL, ELM e NoProp.
$\mathbf{h}_n$	O $n$ -ésimo vetor de ativação oculto das redes RVFL, ELM e NoProp.
$\mathbf{M}$	Matriz de pesos das redes RVFL, ELM e NoProp.
$\mathbf{m}_j^T$	O $j$ -ésimo vetor de pesos da matriz $\mathbf{M}$ com $j = 1, \dots, q$ .
$b_j$	Limiar do $j$ -ésimo neurônio oculto das redes RVFL, ELM e NoProp e técnica BIP com $j = 1, \dots, q$ .
$U(-1,1)$	Distribuição uniforme no intervalo $(-1,1)$ .
$N(0,1)$	Distribuição normal padronizada ( <i>média</i> = 0, <i>variância</i> = 1).
$\lambda$	Parâmetro de regularização (de Tikhonov) com $0 < \lambda \leq 1$ .
$\mathbf{I}$	Matriz identidade de dimensão adequada ao modelo utilizado.
$\mathbf{H}$	Matriz com $N_1$ colunas formada pelos vetores concatenados $\mathbf{h}_n$ – idem $\mathbf{H}_{q \times N_1}$ .
$\mathbf{w}_n$	O vetor de coeficientes na $n$ -ésima iteração de acordo com o modelo sequencial utilizado.
$J(\mathbf{w}_n)$	Função custo dependente de $\mathbf{w}_n$ com objetivo de minimizar os erros de aproximação pelos erros quadráticos médios, e sua primeira derivada é denotada por $\frac{\partial J(\mathbf{w}_n)}{\partial \mathbf{w}_n}$ .
$e_n$	O $n$ -ésimo erro instantâneo na iteração $n$ para modelos sequenciais.
$\eta$	Taxa de aprendizagem dos modelos sequenciais.
$a_j$	Parâmetro de forma da função sigmoide logística da técnica BIP – idem $a$ .
$f_{des}$	Função distribuição exponencial desejada.
$\mathbf{u}_j$	Ativação do $j$ -ésimo neurônio oculto.
$\mathbf{T}_{f_{des}}$	Vetor de alvos aleatórios gerados a partir de $f_{des}$ .
$\Phi(\mathbf{u}_j)$	Matriz com soma sináptica e 1's.
$\mathbf{c}_j$	Vetor de parâmetros com os parâmetros $a_j$ e $b_j$ .
$f^{-1}$	Função inversa da função de ativação.
$\hat{y}_n$	A $n$ -ésima saída estimada.

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b> .....	<b>16</b>
1.1	Objetivo geral .....	18
1.2	Objetivos específicos .....	18
1.3	Organização do restante da dissertação .....	18
<b>2</b>	<b>GERAÇÃO EÓLICA E SEUS FUNDAMENTOS</b> .....	<b>20</b>
2.1	Aerogeradores – Modelagem típica da curva de potência .....	20
2.2	Características operacionais e padronização .....	22
2.3	Função logística, parâmetros e observações .....	24
2.4	Conclusão .....	25
<b>3</b>	<b>MODELOS DE ESTIMAÇÃO CPA</b> .....	<b>26</b>
3.1	Modelo de regressão polinomial .....	26
3.2	Rede <i>multilayer perceptron</i> (MLP) .....	27
3.3	Rede <i>random vector functional link</i> (RVFL) .....	29
3.4	Rede <i>extreme learning machine</i> (ELM) .....	30
3.5	Rede <i>No-Propagation</i> (NoProp) .....	30
3.6	Regra LMS generalizada .....	31
3.7	<i>Batch intrinsic plasticity</i> (BIP) .....	32
3.8	Discussão e conclusão .....	34
<b>4</b>	<b>METODOLOGIA DE SIMULAÇÃO</b> .....	<b>36</b>
4.1	Índices de validação e transformação escalar .....	36
4.2	Modelos avaliados .....	37
4.3	Ambiente computacional e conjuntos de dados .....	38
<b>5</b>	<b>SIMULAÇÕES E RESULTADOS</b> .....	<b>41</b>
5.1	Resultados do modelo de regressão polinomial .....	41
5.2	Resultados do modelo MLP .....	43
5.3	Resultados do modelo ELM .....	45
5.4	Resultados dos modelos NoProp e NoProp-logsig .....	46
5.5	Resultados dos modelos RVFL .....	48
5.6	Sumário dos resultados dos modelos – conjunto WT1 .....	50
5.7	Sumário dos resultados dos modelos – conjunto Delfos .....	51
<b>6</b>	<b>CONCLUSÕES E TRABALHOS FUTUROS</b> .....	<b>55</b>

<b>REFERÊNCIAS .....</b>	<b>56</b>
--------------------------	-----------

## 1 INTRODUÇÃO

A geração de energia, seja a partir de uma única turbina em regiões/comunidades isoladas ou a partir de grandes parques eólicos, vem ganhando importância substancial em vários países em comparação com fontes de geração de energia mais convencionais, tais como nuclear, termelétrica e hidrelétrica. Essa penetração de energia eólica em sistemas de energia colocam questões críticas para o projeto eficiente de energia baseada em plantas de produção eólica. Uma dessas questões surge do fato de que as curvas de potência fornecidas pelo fabricante da turbina são construídas sob condições controladas. No entanto, em aplicações práticas, as turbinas são instaladas em diversas condições climáticas e terrenos complexos (por exemplo, dunas), causando um afastamento considerável da energia efetivamente produzida em relação aos valores teóricos esperados. Na realidade, a relação entre a velocidade do vento no hub da turbina e a saída de energia não é determinística e, portanto, os dados medidos de velocidade do vento e potência estão dispersos em torno da curva de potência do fabricante.

A curva de potência de uma turbina eólica é uma representação gráfica ou matemática da relação entre a potência elétrica gerada (saída) e a velocidade do vento de altura do hub (entrada) (DAS, 2014). Por ser uma característica essencial da turbina, a curva de potência é utilizada para diversos fins, tais como (i) avaliação e previsão de energia eólica; (ii) monitoramento online das curvas de potência; (iii) estimativa de vento e fator de capacidade da turbina; e (iv) seleção de aerogeradores. A natureza não determinística do vento, variação e a presença onipresente de valores discrepantes nos valores medidos de vento/energia são fontes significativas de incertezas que prejudicam o projeto de parques eólicos confiáveis e estáveis.

Devido à discrepância entre os valores de saída reais e esperados, várias técnicas de modelagem têm sido propostas para construir modelos adequados para curvas de potência a partir de dados locais. Na verdade, a modelagem da curva de potência de aerogeradores (CPA) é um campo de pesquisa envolvendo conhecimentos de engenharia, estatística, física e matemática aplicada, como pode ser inferido pela extensa literatura sobre o assunto, tais como em Eminoglu E Turksoy (2019), Shokrzadeh *et al.* (2014), Sohoni *et al.* (2014), e Wang *et al.* (2019). A regressão polinomial é geralmente a primeira escolha para modelagem CPA devido à sua disponibilidade em pacotes de *software* populares, como *MS Excel*, *Matlab* e R, mas hoje em dia, o campo prospera com novas técnicas sendo introduzidas em um ritmo acelerado, especialmente modelos de aprendizado de máquina, como redes neurais artificiais (BAI *et al.*, 2019; MANOBEL *et al.*, 2018; YAN *et al.*, 2019), modelos logísticos (VILANUEVA e FEIJÓO, 2018), regressão baseada em spline (MEHRJOO *et al.*, 2020), máquinas vetoriais de

suporte (HU *et al.*, 2020), processos gaussianos e modelos bayesianos (GUO e INFIELD, 2020), floresta aleatória (PANDIT *et al.*, 2019), e modelos baseados em lógica fuzzy (GONZÁLEZ-CARRATO, 2018).

Uma nova classe de arquiteturas de rede neurais supervisionadas do tipo *feedforward* de camada única, genericamente chamadas redes neurais randomizadas (RNRs), tem atraído muita atenção da comunidade de inteligência computacional nos últimos anos. Alguns exemplos de tais arquiteturas são a rede RVFL (*random vector functionl link*) (PAO *et al.*, 1994), máquina de aprendizado extremo (ELM, *extreme learning machine*) (HUANG *et al.*, 2006) e a rede *No-Propagation* (NoProp) (WIDROW *et al.*, 2013). Todo este interesse parece ser motivado principalmente pela forma muito rápida com que estas redes são treinadas, sem recorrer a um longo processo de aprendizado em várias épocas de treinamento, conforme exigido pelo algoritmo de retropropagação do erro (RUMELHART *et al.*, 1986). Mesmo sendo este um argumento válido, muitas aplicações do mundo real apresentam recursos desafiadores que exigem adaptações nos algoritmos de aprendizagem das RNRs acima mencionadas. Uma dessas aplicações é exatamente a modelagem da curva de potência, especialmente a partir de conjuntos de dados ruidosos de grande escala ou de conjuntos de dados contendo valores atípicos (i.e., *outliers*).

Nesse sentido, em uma comunidade inundada com uma infinidade de modelos de aprendizado de máquina para regressão, muitas abordagens teóricas tem sido propostas sem realmente considerar as necessidades do usuário final interessado na solução de um determinado problema de engenharia. Em outras palavras, está se tornando cada vez mais difícil decidir qual modelo se encaixa melhor em uma determinada aplicação. Este é o caso da modelagem de CPAs, em que muita atenção tem sido dada à precisão do modelo por meio de índices numéricos de desempenho, como a raiz quadrada da média quadrática do erro (RMSE) e o coeficiente de determinação ( $R^2$ ), enquanto pouca atenção - ou nenhuma - tem sido dedicada a uma avaliação qualitativa (por simples inspeção visual, por exemplo) da curva de potência estimada e sua adequação ao problema em questão. Alguns modelos permitem a ocorrência de soluções patológicas; isto é, soluções com precisão numérica aceitável, mas inadequadas ao problema real, pois eles geram previsões anômalas para a energia gerada. Do ponto de vista do usuário, a ocorrência de soluções patológicas é altamente indesejável porque leva a previsões não confiáveis.

Tendo isso em mente, o objetivo geral e os objetivos específicos deste trabalho de mestrado são apresentados a seguir.

## 1.1 Objetivo geral

O objetivo geral desta dissertação consiste em um amplo estudo de desempenho de modelos de RNRs para modelagem da curva de potência de aerogeradores, também chamados de turbinas eólicas, a partir de dados medidos em turbinas reais, sujeitos a incertezas e *outliers*.

Para isso, utilizamos dois conjuntos de dados desafiadores (pelo grande volume e dificuldade em ser tratado) para testar a hipótese de trabalho deste estudo. O primeiro é um conjunto de dados ruidosos de grande escala disponível publicamente e usado em Lee *et al.* (2015), enquanto o outro é fornecido por uma empresa privada de TI<sup>1</sup> exclusivamente para uso acadêmico. A tarefa de estimativa de CPA escolhida envolve apenas uma entrada (velocidade do vento) e uma saída (energia elétrica), uma vez que este ainda é o cenário mais comum em aplicações do mundo real.

## 1.2 Objetivos específicos

Os objetivos específicos estão ligados aos resultados oriundos de uma abrangente comparação de desempenho envolvendo três modelos de rede neural randomizadas e variantes desenvolvidas ao longo desta pesquisa, nomeadamente, as redes RVFL, ELM e NoProp. Por uma questão de completude, também serão avaliados o modelo de regressão polinomial e a rede neural MLP como alternativas do estado da arte. Estes modelos serão comparados ao longo de três dimensões críticas: (i) acurácia numérica do ajuste da curva, (ii) sensibilidade aos hiperparâmetros do modelo, e (iii) ocorrência de soluções patológicas.

## 1.3 Organização do restante da dissertação

O restante dos capítulos desta dissertação estão organizados como segue. O Capítulo 2 traz uma breve apresentação dos fundamentos envolvidos na produção de energia eólica e sobre a física do fenômeno. O Capítulo 3 descreve os modelos polinomial, MLP, e RNRs avaliados neste trabalho e as variantes desenvolvidas. O Capítulo 4 apresenta a metodologia usada na avaliação dos modelos e também uma descrição dos conjuntos de dados usados nos experimentos de simulação computacional. O Capítulo 5 apresenta os resultados do estudo comparativo envolvendo as diversas RNRs estudadas, discutindo-os no contexto do

---

<sup>1</sup> Delfos IM (DELFOSS INTELLIGENT MAINTENANCE, 2017-). Disponível em: "<http://www.delfosim.com/>". Acesso em: 15 de janeiro de 2022.

problema de interesse. As conclusões da pesquisa e trabalhos futuros são apresentados no Capítulo 6.

## 2 GERAÇÃO EÓLICA E SEUS FUNDAMENTOS

No presente capítulo faz-se uma apresentação da problemática da geração eólica destacando-se alguns pontos relevantes. O conteúdo conceitual apresentado neste capítulo está baseado nas dissertações de mestrado de Maya (2018) e Virgolino (2020), que também desenvolveram técnicas de aprendizado para a modelagem da curva de potência de aerogeradores. Mais especificamente, Maya (2018) desenvolveu uma técnica de estimação automática da ordem de modelos polinomiais usando algoritmos genéticos, enquanto Virgolino (2020) aplicou processos gaussianos, um método de kernel probabilístico.

O objetivo deste capítulo é fornecer uma visão geral sobre a questão da modelagem da curva de potência de aerogeradores (turbinas eólicas), bem como discutir as características operacionais dos aerogeradores e a função matemática utilizada na modelagem dos dados.

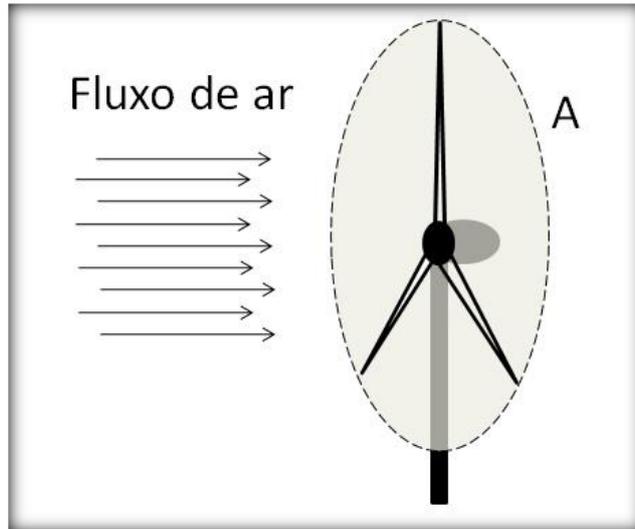
### 2.1 Aerogeradores - Modelagem típica da curva de potência

Os aerogeradores geram energia elétrica a partir do movimento de suas pás que é causado pela ação do vento. A geração dessa energia chamada de potência  $P$  (em Watts) sofre influência direta da velocidade do vento  $v$  (em m/s), os aerogeradores por sua vez são testados em condições mais controladas do que aquelas encontradas no local onde serão/estão efetivamente instalados e utilizados. Por isso, há a necessidade e interesse em monitorar o bom funcionamento dos aerogeradores visando a sua melhor utilização, desempenho e preservação no local de uso.

Analisar e prever a potência  $P$  dadas as condições locais de vento  $v$ , temperatura e densidade do ar, são de extrema importância. Nesta dissertação utilizaremos apenas  $v$  e  $P$  para a tarefa de modelagem da curva de potência do aerogerador por ser a abordagem mais comum em aplicações práticas.

Considere um aerogerador de eixo horizontal com área  $A$  (em  $m^2$ ) varrida por suas pás (hélices), instalado em local com densidade do ar constante  $\rho$  (em  $kg/m^3$ ) e que a componente de velocidade do vento  $v$  que atravessa perpendicularmente a área  $A$  é assumidamente constante, conforme ilustrado na Figura 1.

Figura 1 – Esquema da passagem do fluxo do ar pela área  $A$  varrida do aerogerador



Fonte: Adaptada de Maya (2018).

A partir da mecânica clássica, a potência  $P$  pode ser calculada como

$$P = \frac{\Delta E_c}{\Delta t}, \quad (2.1)$$

com  $t$  a medida do tempo (em segundos) e  $E_c$  a energia cinética do movimento (em Joules).

Considerando  $M$  a massa (em kg) e sendo  $v$  constante, sabe-se também que

$$E_c = \frac{1}{2} M v^2, \quad (2.2)$$

e que a variação  $\Delta E_c$  dessa energia pode ser obtida por

$$\Delta E_c = \frac{1}{2} \Delta M v^2. \quad (2.3)$$

Assim, pode-se calcular  $P$  com as devidas substituições como segue:

$$\begin{aligned} P &= \frac{\Delta E_c}{\Delta t} \\ P &= \frac{1}{2} \frac{\Delta M}{\Delta t} v^2 \\ P &= \frac{1}{2} (\rho A v) v^2 \\ P &= \frac{1}{2} \rho A v^3. \end{aligned} \quad (2.4)$$

A dedução da Eq. (2.4) pode ser encontrada em Maya (2018). O que ocorre no uso do aerogerador em um parque real é que existem  $P_d$ , potência eólica disponível, e  $P_a$ , potência absorvida, que é aproveitada para o movimento do rotor do aerogerador. Por este motivo, defini-se o coeficiente de potência  $C_p$ , dado por

$$C_p = \frac{P_a}{P_d}, \quad (2.5)$$

como uma medida adimensional responsável por indicar a capacidade de absorção do

aerogerador da potência eólica disponível (SHOKRZADEH *et al.*, 2014, apud MAYA, 2018). Ademais, o valor  $C_p$  pode ser obtido a partir dos dados do fabricante, sendo que seu máximo teórico, chamado de limite de Betz, é dado por  $16/27 \cong 0,5926$  (BETZ, 1920, apud MAYA, 2018), e a eficiência do rotor  $\alpha$ , então pode-se obter a potência elétrica gerada  $P_{eg}$  pelo fluxo de vento por meio da seguinte equação:

$$P_{eg} = \frac{1}{2} \alpha C_p \rho A v^3, \quad (2.6)$$

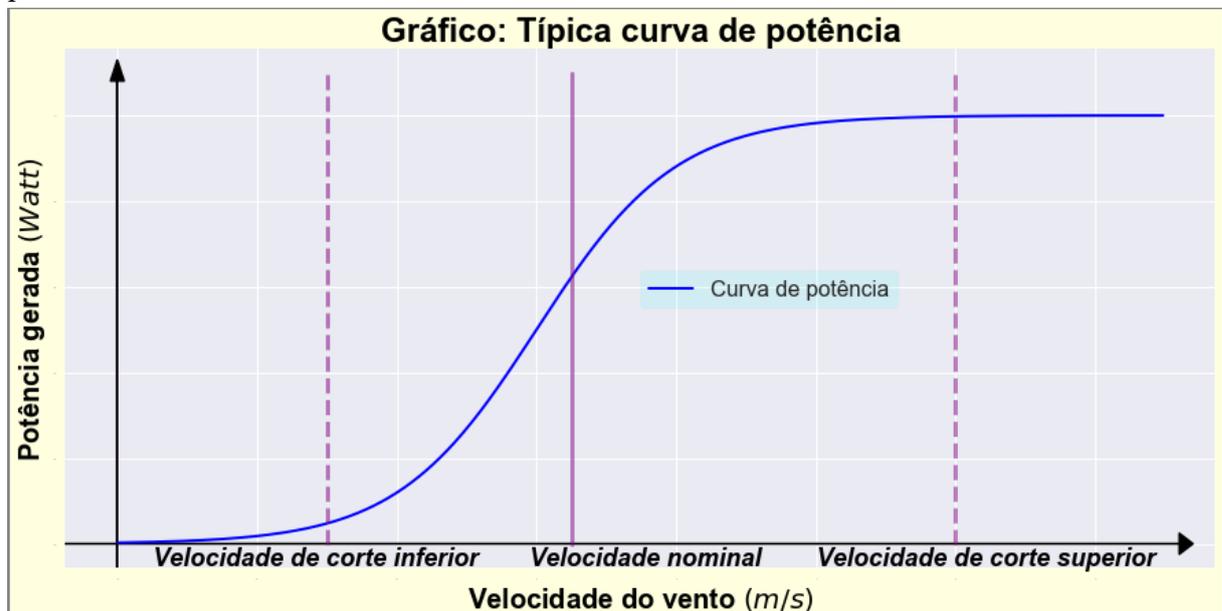
sendo a Eq. (2.6) obtida através da multiplicação de  $C_p$  e  $\alpha$  pela Eq. (2.4).

A ideia aqui exposta está de acordo com um tipo de aerogerador. Existem outros tipos, diferentes fabricantes, o que influencia  $C_p$  bem como os dados disponíveis usados no problema de regressão. Para este trabalho, as velocidade do vento e a potência gerada de saída serão os dados analisados para a modelagem usando redes neurais randomizadas.

## 2.2 Características operacionais e padronização

Uma típica curva de potência é mostrada na Figura 2.

Figura 2 – Gráfico com típica forma da curva de potência de um aerogerador com controle de pitch



Fonte: Adaptada de Apata e Oyedokun (2020).

Nota-se que existem três regiões que destacam o funcionamento do aerogerador. Quando a velocidade do vento é menor que a velocidade de corte inferior  $v_{ci}$ , também chamada de velocidade de partida, não se tem produção de potência no aerogerador. A velocidade

nominal  $v_n$  é a velocidade a qual o aerogerador produz potência. Acima da velocidade de corte superior  $v_{cs}$  o aerogerador aciona seu mecanismo de proteção para não exceder a velocidade máxima suportada. É possível observar que

$$\begin{aligned} v_{ci} \leq v_n \leq v_{cs} &\Leftrightarrow P(v_{ci}) \leq P(v_n) \leq P(v_{cs}) \\ 0 &\leq P(v_n) \leq P_{max}, \end{aligned} \quad (2.7)$$

quando  $v \rightarrow v_{cs}$  então  $P(v_{cs})$  tende à produção de máxima potência, porém os valores próximos às velocidades de corte são difíceis de se modelar pois podem ser negativos abaixo de  $v_{ci}$  e ainda tem o efeito de histerese da velocidade  $v_{cs}$  que ocorre durante o desligamento e reinicialização da turbina, o que afeta sua produtividade.

São vários os fatores que influenciam a geração eólica através de aerogeradores, sendo que destacamos condições do vento local, densidade do ar, extrapolação da velocidade do vento, e condição da turbina. Todos de suma importância para a realização da tarefa de modelagem da curva de potência.

Por reconhecer a importância da modelagem da curva de potência, a International Electrotechnical Commission (IEC) propôs métodos padronizados amplamente aceitos na indústria de energia eólica. A IEC-61400-12-2 (INTERNATIONAL ELECTROTECHNICAL COMMISSION, 2013) é uma padronização internacional que descreve métodos de adquirir dados e procedimentos de ajuste de curva (modelagem de curva) para aerogeradores de modo individual. Um dos métodos utilizados é o chamado *método de caixas*, no qual são observados médias de velocidades e de potências, armazenadas em intervalos, o que discretiza os valores observados dessas medidas.

A padronização IEC exige que as observações de velocidade do vento e potência de saída  $(v_i, P_i)$  sejam obtidos a partir de médias contínuas de valores no intervalo de 10 minutos, o que quase todos os fabricantes de turbinas eólicas seguem. Uma das dificuldades em modelar a partir de dados são os *outliers*, que são registros de dados com valores que em tese não eram para constar nas medidas realizadas, ou seja, a partir da inspeção das medidas é percebida uma anomalia e de modo que um *outlier* pode estar presente nas medidas. Grande número de dados a serem tratados é outro fator que dificulta a modelagem pois exige maior poder computacional em relação ao tempo de processamento dos dados, exigindo máquinas de maior capacidade de memória, armazenamento e velocidade de CPU, fatores estes que encarecem e podem eventualmente inviabilizar uma execução rápida e eficiente da tarefa de modelagem da curva de potência.

### 2.3 Função logística, parâmetros e observações

Uma escolha comum para a solução do problema de modelagem de curva de potências de aerogeradores é a função logística. Aqui, considerando a potência de saída em função da velocidade do vento, tem-se

$$P(v) = 1/(1 + \exp(-v)), \quad (2.8)$$

contínua e derivável com derivada dada em função da própria  $P(v)$ , ou seja,

$$P'(v) = \left( \frac{1}{1 + \exp(-v)} \right)' = \frac{0 \cdot (1 + \exp(-v)) - 1(1 + \exp(-v))'}{(1 + \exp(-v))^2}$$

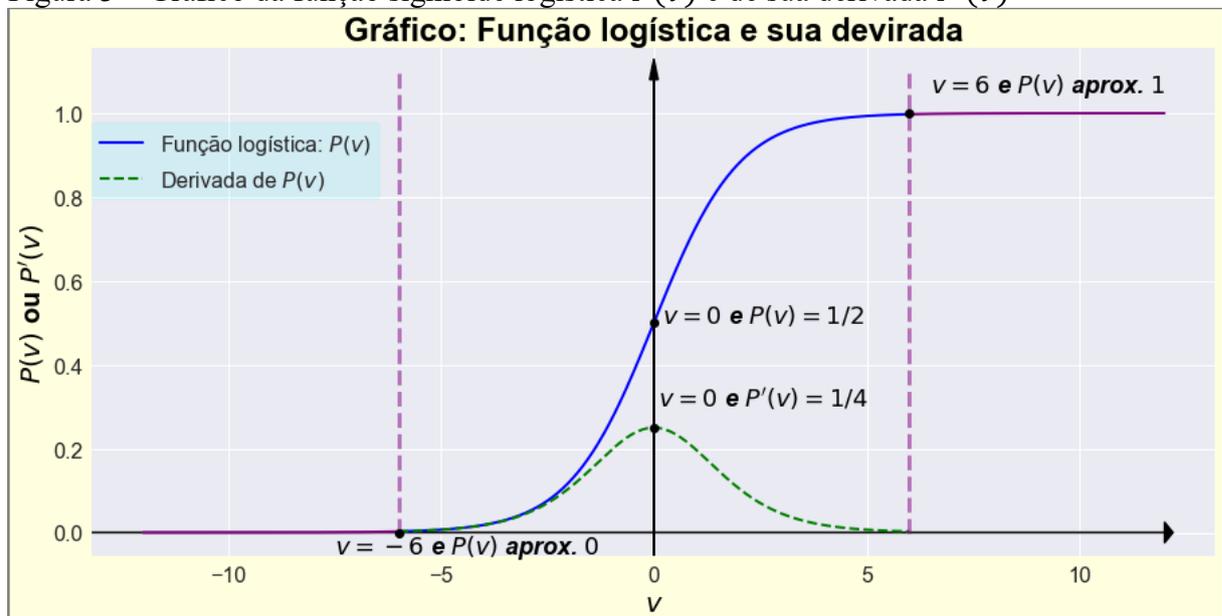
$$P'(v) = \frac{\exp(-v)}{1 + \exp(-v)} \times \frac{1}{1 + \exp(-v)} = \frac{1 + \exp(-v) - 1}{1 + \exp(-v)} P(v)$$

$$P'(v) = P(v)(1 - P(v)), \quad (2.9)$$

uma notável vantagem matemática.

Por análise gráfica, a partir da Figura 3 e considerando  $v \in [-10,10]$  a seguir, é perceptível a existência de pontos matematicamente interessantes, do comportamento suave da função  $P(v)$  e de sua derivada.

Figura 3 – Gráfico da função sigmoide logística  $P(v)$  e de sua derivada  $P'(v)$



Fonte: Elaborada pelo autor.

Destaca-se que a função para valores próximos e maiores que 6 assume valor cada vez mais próximo de 1, quando próximo a  $-6$  e menores o valor que assume é próximo de 0, em  $v = 0$  ocorre seu ponto de inflexão com valor  $P(0) = 0,5$ , observe seu comportamento

suave ao longo do intervalo. Sua derivada tem comportamento suave, não ocorre grande variação do seu valor, e que próximo a 6 e para valores maiores, e a partir de  $-6$  e por valores menores  $P'(v)$  tende a 0. Veja que  $P(v)$  possui forma semelhante a forma típica da curva de potência, e que ao utilizar técnicas de modelagem com informação do gradiente (primeira derivada) de  $P$  espera-se um comportamento suave por possuir derivada com essas características. Tudo isso torna  $P(v)$  matematicamente interessante ao domínio do problema de modelagem.

## 2.4 Conclusão

No presente capítulo foi definida, de forma geral, a curva de potência de aerogeradores. Buscou-se um panorama simples visando o bom entendimento do problema deste tipo de modelagem e com destaque aos pontos mais importantes necessários ao entendimento. Na contemporaneidade, existe uma necessidade de novas fontes de energias e que causem menos danos ambientais a fim de que as sociedades vivam em mais harmonia e equilíbrio com a natureza.

Até aqui abordou-se uma visão das necessidades do problema, inicialmente uma forma de entendimento pela mecânica clássica com abordagem de um tipo de aerogerador, com auxílio gráfico e as formulações matemáticas com as características e observações quanto ao uso em campo. Seguindo, foi mostrado um pouco das características operacionais e de padronização por meio gráfico e matemático das faixas operacionais e do padrão IEC. Por fim, encerrou-se com um necessário estudo da função logística e de sua derivada.

O próximo capítulo abordará as técnicas utilizadas para a modelagem da curva de potência de aerogerador (turbina eólica) onde se dará destaque as técnicas que visam a utilização de redes neurais artificiais randomizadas, principal objetivo deste trabalho.

### 3 MODELOS DE ESTIMAÇÃO CPA

Devido a discrepância entre valores reais de saída e esperados, várias técnicas de modelagem de curva de potência de aerogeradores (CPA) têm sido utilizadas para construir modelos cada vez mais precisos para curvas de potências a partir de dados *in loco*.

Como já dito, aos modelos são ajustados dados coletados *in loco*. Em geral, considere que tem-se um conjunto de dados com  $N$  observações entrada-saída (pares ordenados) de  $\mathbf{x}_n \in \mathbb{R}^p$  (entrada de dimensão  $p$ ) e respectiva saída alvo  $d_n \in \mathbb{R}$  (saída real de dimensão 1), ou seja, tem-se  $\{(\mathbf{x}_n, d_n)\}_{n=1}^N$  com  $N > 0$  e natural (geralmente grande). Então, selecionando-se (aleatoriamente)  $N_1$  padrões entrada-saída ( $N_1 < N$ ) de treinamento dos dados e organizando os vetores de entrada ao longo das colunas da matriz  $\mathbf{X}_{p \times N_1}$ , enquanto as saídas alvo são organizadas no vetor coluna  $\mathbf{d}_{N_1 \times 1}$ , tem-se

$$\mathbf{X} = [\mathbf{x}_1 | \mathbf{x}_2 | \dots | \mathbf{x}_{N_1}] \quad \text{e} \quad \mathbf{d} = [d_1, \dots, d_{N_1}]^T, \quad (3.1)$$

onde  $T$  denota a transposição do vetor/matriz.

#### 3.1 Modelo de regressão polinomial

Para um modelo de regressão polinomial, considera-se que existe um polinômio de grau arbitrário  $q$  o qual descreve bem a relação entrada-saída. Em Maya (2018), usa-se tal modelo e o modelo polinomial robusto, dentre outros para a modelagem de CPA. Para o modelo de regressão polinomial considere que a relação entrada  $x_n$  e saída  $y_n$  pode ser descrita como

$$y_n = w_0 + w_1 x_n + w_2 x_n^2 + \dots + w_q x_n^q, \quad (3.2)$$

com  $w_i \in \mathbb{R}$ ,  $i = 0, 1, \dots, q$ , onde tem-se  $q + 1$  coeficientes  $w_i$  do polinômio a serem estimados.

Para melhor entendimento, somente para este modelo considere  $x_n \in \mathbb{R}$  e  $x_n > 0$ . O polinômio a ser determinado é linear nos parâmetros  $w_i$ 's e pode ser obtido pelo método dos mínimos quadrados ordinários (MQO). Para tanto, considere a expressão matricial a seguir

$$\mathbf{y} = \mathbf{Z}\mathbf{w} + \boldsymbol{\delta}, \quad (3.3)$$

onde  $\mathbf{y}$  possui dimensão  $N_1 \times 1$  e  $\mathbf{y} = [y_1, \dots, y_{N_1}]^T$ ,  $\mathbf{w} \in \mathbb{R}^{q+1}$  e a matriz  $\mathbf{Z} = [\mathbf{z}_0 | \mathbf{z}_1 | \dots | \mathbf{z}_q]$  de dimensão  $N_1 \times (q + 1)$  com colunas  $\mathbf{z}_i = [(x_1)^i, \dots, (x_{N_1})^i]$ ,  $i = 0, 1, \dots, q$ . O vetor de coeficientes  $\mathbf{w}$  pode ser determinado a partir dos dados medidos alocados na matriz  $\mathbf{Z}$  e no vetor  $\mathbf{d}$  como

$$\mathbf{d}_{N_1 \times 1} = \mathbf{Z}_{N_1 \times (q+1)} \mathbf{w}_{(q+1) \times 1}$$

$$\begin{aligned}\mathbf{Z}^T \mathbf{d} &= \mathbf{Z}^T \mathbf{Z} \mathbf{w} \\ \mathbf{w} &= (\mathbf{Z} \mathbf{Z}^T)^{-1} \mathbf{Z}^T \mathbf{d}.\end{aligned}\quad (3.4)$$

Pode-se mostrar que o estimador mostrado na Eq. (3.4) provê uma solução ótima segundo o critério MQO. Para isto, deve-se minimizar os erros quadráticos pela função custo  $J(\mathbf{w}) = \|\boldsymbol{\delta}\|^2 = \boldsymbol{\delta}^T \boldsymbol{\delta}$  com  $\boldsymbol{\delta} = \mathbf{d} - \mathbf{Z} \mathbf{w}$ . Assim, a função custo pode ser escrita como

$$(\mathbf{d} - \mathbf{Z} \mathbf{w})^T (\mathbf{d} - \mathbf{Z} \mathbf{w}) = (\mathbf{d}^T - (\mathbf{Z} \mathbf{w})^T) (\mathbf{d} - \mathbf{Z} \mathbf{w}) = \mathbf{d}^T \mathbf{d} - 2 \mathbf{d}^T \mathbf{Z} \mathbf{w} + \mathbf{w}^T \mathbf{Z}^T \mathbf{Z} \mathbf{w},$$

sendo que  $\mathbf{d}^T \mathbf{Z} \mathbf{w} = \mathbf{w}^T \mathbf{Z}^T \mathbf{d}$  pois representam o mesmo número real. Em geral, a função  $J$  na variável  $\mathbf{w}$  comporta-se como uma função do segundo grau em que o termo quadrático da expressão é  $\mathbf{w}^T \mathbf{Z}^T \mathbf{Z} \mathbf{w}$ , e a matriz de Vandermond  $\mathbf{Z}$  obtida dos valores  $x_n > 0$ , possui entradas estritamente positivas. A função  $J$  possui ponto de mínimo global. Derivando duas vezes  $J$  em relação a  $\mathbf{w}$  obtém-se

$$\frac{\partial J(\mathbf{w})}{\partial \mathbf{w}} = [\mathbf{d}^T \mathbf{d} - 2 \mathbf{d}^T \mathbf{Z} \mathbf{w} + \mathbf{w}^T \mathbf{Z}^T \mathbf{Z} \mathbf{w}]' = -2 \mathbf{d}^T \mathbf{Z} + 2 \mathbf{w}^T \mathbf{Z}^T \mathbf{Z}, \quad \text{e} \quad \frac{\partial^2 J(\mathbf{w})}{\partial \mathbf{w}^2} = 2 \mathbf{Z}^T \mathbf{Z} > 0,$$

e igualando a primeira derivada ao vetor zero  $\mathbf{0}$  obtém-se a Eq. (3.4) (a menos de transposição) e a segunda derivada sendo maior que zero indica que  $J$  possui ponto de mínimo global (como anteriormente afirmado).

### 3.2 Rede *multilayer perceptron* (MLP)

A rede MLP consiste de  $p$  entradas efetivas, uma camada oculta com  $q$  neurônios, e uma camada de saída com  $r$  neurônios, e denota-se tal rede como  $MLP(p, q, r)$ . Na sua formulação mais geral, neurônios ocultos e de saída tem função de ativação não-linear denotada por  $\phi(\cdot)$ . Uma escolha comum é  $\phi(u) = 1/(1 + \exp(-u))$ , função sigmoide logística, onde  $u$  é o argumento da função. Assim, a saída do  $k$ -ésimo neurônio  $y_k$ , com  $k = 1, \dots, r$ , é dada por

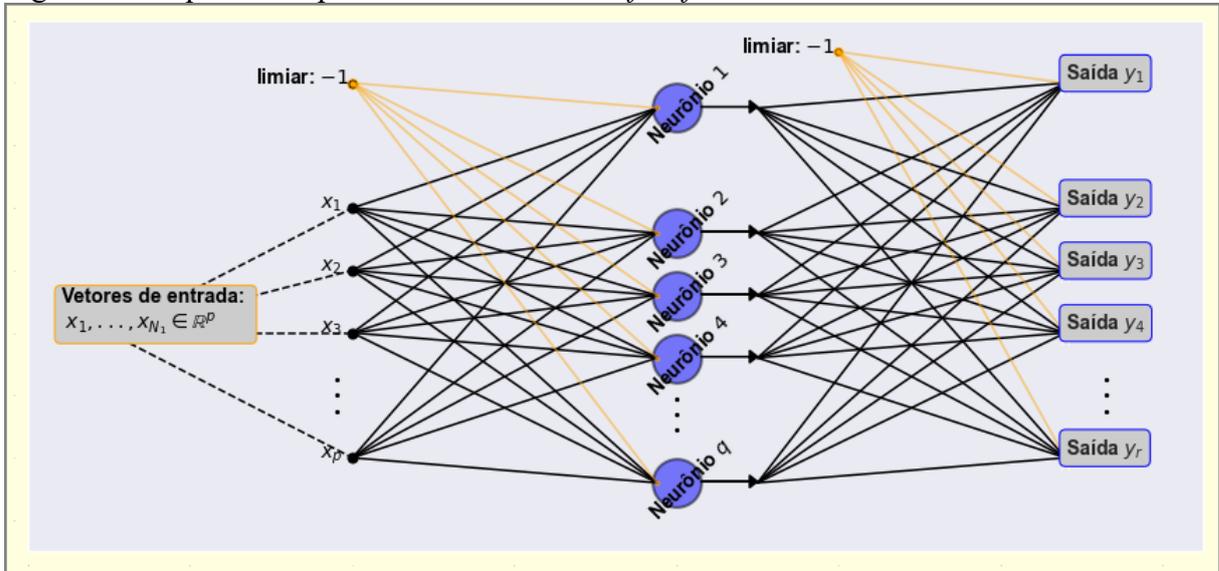
$$y_k = \phi\left[\sum_{i=1}^q m_{ki} \phi\left(\sum_{j=1}^p w_{ij} x_j - w_{i0}\right) - \theta_k\right], \quad (3.5)$$

em que  $x_j$  é a  $j$ -ésima unidade de entrada,  $w_{ij}$  o peso conectando a  $j$ -ésima unidade de entrada com o  $i$ -ésimo neurônio oculto, e  $m_{ki}$  o peso conectando o  $i$ -ésimo neurônio oculto ao  $k$ -ésimo neurônio de saída. O parâmetro  $\theta_k$  é o limiar do  $k$ -ésimo neurônio de saída, enquanto que  $w_{i0}$  é o limiar para o  $i$ -ésimo neurônio oculto. O número  $L$  de parâmetros ajustáveis da rede  $MLP(p, q, r)$  é  $L = (p + 1)q + (q + 1)r$ , ou seja, depende do número de entradas, neurônios ocultos e de saída. Os pesos e limiares do modelo MLP são computados através do algoritmo de retropropagação do erro (RUMELHART *et al.*, 1986), que é baseado no método do gradiente descendente, um simples mas eficiente algoritmo de otimização baseado em derivadas de

primeira ordem.

A arquitetura básica de uma rede neural  $\phi$  de camada oculta única com  $q$  neurônios é ilustrada pela Figura 4 a seguir:

Figura 4 – Arquitetura típica de uma rede neural *feedforward* com uma única camada oculta



Fonte: Elaborada pelo autor.

Pela Figura 4, na rede entra  $\mathbf{x}_n$ , manipula-se através da função  $\phi$  não-linear adicionando uma entrada constante  $-1$ , para se obter a saída  $y_n = \phi(\mathbf{x}_n)$ . Sua camada oculta e única é representada pelos  $q$  neurônios em que cada um deles recebe as  $p$  componentes de  $\mathbf{x}_n$  e limiar tal que cada neurônio contribui diretamente, sem passar por outra camada de neurônios, para a saída  $y_n$  onde atua também o limiar.

A Figura 4 representa a arquitetura básica de uma rede neural artificial. Pode-se observar que  $\mathbf{x}_n$ , juntamente com o limiar são processados pelos  $q$ 's neurônios ocultos para então todos esses contribuírem para a computação da saída (ou neurônio de saída)  $y_1$ . E assim ocorre para cada um dos  $N_1$  vetores de entrada da rede. A rede MLP aprende um mapeamento entrada-saída através da minimização de uma função objetivo escolhida, geralmente o erro médio quadrático.

Para a modelagem de CPA, objeto de interesse desta dissertação, com o modelo MLP, temos uma variável de entrada unidimensional a velocidade do vento, e uma saída unidimensional  $y = P$  a potência gerada. Para este cenário, tem-se a representação do modelo como  $MLP(1, q, 1)$  e sua saída estimada por

$$\hat{y} = \phi\left[\sum_{i=1}^q m_i \phi(w_{i1}x - w_{i0}) - \theta\right], \quad (3.6)$$

sendo  $x_1 = x$ ,  $\theta = -1$ . O número de parâmetros ajustáveis para o presente modelo na modela-

gem de CPA é  $L = (1 + 1)q + (q + 1)1 = 3q + 1$ .

### 3.3 Rede *random vector functional link* (RVFL)

A rede RVFL (PAO *et al.*, 1994; ZHANG e SUGANTHAN, 2016) é uma rede randomizada *feedforward* com uma camada oculta com dois caminhos para processar a informação de entrada para os neurônios de saída. Os dois caminhos são adicionados formando a saída da rede. Um caminho é linear, que conecta as unidades de entrada ao neurônio de saída. Matematicamente, tem-se

$$y_n^{(1)} = \mathbf{w}_1^T \mathbf{x}_n, \quad (3.7)$$

com  $\mathbf{w}_1 \in \mathbb{R}^p$  sendo o vetor de pesos. O segundo caminho processa os vetores de entrada através da camada oculta com  $q$  ( $q \geq 1$ ) neurônios não-lineares, isto é,

$$y_n^{(2)} = \mathbf{w}_2^T \mathbf{h}_n, \quad (3.8)$$

sendo  $\mathbf{w}_2 \in \mathbb{R}^q$  o correspondente vetor de pesos e  $\mathbf{h}_n \in \mathbb{R}^q$  o vetor de ativações dos neurônios ocultos, isto é, que contém os valores de saída dos neurônios ocultos em resposta ao vetor de entrada  $\mathbf{x}_n$ . O vetor  $\mathbf{h}_n$  é calculado como

$$\mathbf{h}_n = \phi(\mathbf{M}\mathbf{x}_n) = [\phi(\mathbf{m}_1^T \mathbf{x}_n + b_1), \dots, \phi(\mathbf{m}_q^T \mathbf{x}_n + b_q)]^T, \quad (3.9)$$

sendo  $\phi(v)$  (com o vetor  $v = \mathbf{m}_j^T \mathbf{x}_n + b_j$ ) uma função de ativação não-linear (sigmoideal por exemplo) operando em cada componente do vetor argumento  $v$ ,  $\mathbf{M}$  é uma  $q \times p$  matriz de pesos, e  $b_j$ , com  $j = 1, \dots, q$ , denota o limiar do  $j$ -ésimo neurônio oculto. Deve-se notar que  $\mathbf{w}_1$  e  $\mathbf{w}_2$  são estimados através dos dados,  $\mathbf{M}$  e  $b_j$  são construídos a partir de amostras obtidas aleatoriamente da distribuição uniforme  $U(-1,1)$  ou normal  $N(0,1)$ .

Ao adicionar as saídas dos dois caminhos obtém-se a saída final da rede, dada por

$$y_n = y_n^{(1)} + y_n^{(2)} = \mathbf{w}_1^T \mathbf{x}_n + \mathbf{w}_2^T \mathbf{h}_n = [\mathbf{w}_1^T | \mathbf{w}_2^T] \begin{bmatrix} \mathbf{x}_n \\ \mathbf{h}_n \end{bmatrix} = \mathbf{w}^T \mathbf{z}_n, \quad (3.10)$$

sendo que  $\mathbf{w}^T = [\mathbf{w}_1^T | \mathbf{w}_2^T]$  é um vetor-linha de dimensão  $1 \times (p + q)$  construído pela concatenação dos vetores de pesos  $\mathbf{w}_1$  e  $\mathbf{w}_2$ . Por semelhante motivo,  $\mathbf{z}_n$  é um vetor-coluna de dimensão  $(p + q) \times 1$  construído da concatenação dos vetores  $\mathbf{x}_n$  e  $\mathbf{h}_n$ .

O vetor de pesos  $\mathbf{w}$  pode ser estimado pelo método dos mínimos quadrados ordinários (MQO) por meio da seguinte expressão:

$$\mathbf{w} = (\mathbf{Z}\mathbf{Z}^T)^{-1} \mathbf{Z}\mathbf{d}, \quad (3.11)$$

com  $\mathbf{Z}_{(p+q) \times N_1} = [\mathbf{z}_1 | \dots | \mathbf{z}_{N_1}]$  sendo uma  $(p + q) \times N_1$  matriz com  $N_1$  colunas consistindo nos vetores concatenados  $\mathbf{z}_n^T = [\mathbf{x}_n^T | \mathbf{h}_n^T]^T \in \mathbb{R}^{p+q}$ ,  $n = 1, \dots, N_1$ , sendo  $N_1$  o número de padrões de

entrada do treinamento e o vetor  $\mathbf{d}$  anteriormente definido pela Eq. (3.1). Para evitar problemas causados por um mau-condicionamento da matriz  $\mathbf{Z}\mathbf{Z}^T$ , a versão regularizada da Eq. (3.11) é comumente utilizada, sendo dada por

$$\mathbf{w} = (\mathbf{Z}\mathbf{Z}^T + \lambda\mathbf{I})^{-1}\mathbf{Z}\mathbf{d}, \quad (3.12)$$

sendo  $0 \leq \lambda \ll 1$  o parâmetro de regularização.

### 3.4 Rede *extreme learning machine* (ELM)

A rede ELM é uma arquitetura *feedforward* randomizada introduzida por Huang *et al.* (2006), na qual os pesos da entrada para os neurônios ocultos são escolhidos aleatoriamente, enquanto que os pesos dos neurônios ocultos para a saída são determinados analiticamente. Conseqüentemente, o modelo ELM oferece significativas vantagens, tais como rápida velocidade de aprendizado, fácil implementação e menor intervenção humana quando comparada a outras redes neurais convencionais, tais como a rede MLP e a rede RBF.

Do ponto de vista arquitetural, a rede ELM pode ser entendida como uma rede RVFL na qual o caminho linear da Eq. (10) é removido, então a partir da entrada  $\mathbf{x}_n$  obtém-se a saída

$$y_n = y_n^{(2)} = \mathbf{w}_2^T \mathbf{h}_n, \quad (3.13)$$

e a estimação do vetor de pesos  $\mathbf{w}_2$  é obtido por

$$\mathbf{w}_2 = (\mathbf{H}\mathbf{H}^T)^{-1}\mathbf{H}\mathbf{d}, \quad (3.14)$$

com  $\mathbf{H}_{q \times N_1} = [\mathbf{h}_1 | \dots | \mathbf{h}_{N_1}]$  a  $q \times N_1$  matriz com  $N_1$  colunas, sendo os vetores concatenados  $\mathbf{h}_n \in \mathbb{R}^q$ ,  $n = 1, \dots, N_1$ , sendo  $N_1$  o número de padrões de entrada do treinamento.

### 3.5 Rede *No-Propagation* (NoProp)

Em algumas aplicações como equalização de canal adaptativo e identificação recursiva de sistemas dinâmicos, regras de aprendizagem adaptativa são a melhor opção, uma vez que o vetor de pesos  $\mathbf{w}$  é atualizado de acordo com o padrão de cada entrada apresentado e, após a atualização, o padrão anterior é completamente descartado. Uma das regras de aprendizagem sequencial é conhecida como o algoritmo *least mean squares* (LMS), também conhecida por Widrow-Hoff ou regra delta, a qual foi utilizada por Widrow *et al.* (2013), para introduzir a rede NoProp. Basicamente, a NoProp pode ser entendida como uma rede ELM tal que os pesos de saída  $\mathbf{w}$  são estimados por meio de uma regra de atualização sequencial dos

pesos em vez do estimador MQO como nas redes RVFL e ELM.

Com a ideia de permitir uma rede ELM (ou RVFL) processar dados de forma sequencial, troca-se a regra de estimação MQO pela LMS. Para este propósito, considere primeiramente a função objetivo do erro quadrático instantâneo associada ao neurônio de saída para o  $n$ -ésimo vetor de entrada apresentado:

$$J(\mathbf{w}_n) = \frac{1}{2} e_n^2 = \frac{1}{2} (d_n - y_n)^2 = \frac{1}{2} (d_n - \mathbf{w}_n^T \mathbf{h}_n)^2, \quad (3.15)$$

com  $\mathbf{w}_n \in \mathbb{R}^{p+q}$  o vetor de pesos do neurônio de saída na iteração  $n$ , e  $e_n = d_n - y_n$  é o erro instantâneo desse neurônio da iteração  $n$ . A ideia é minimizar  $J(\cdot)$  e, conseqüentemente o erro  $e_n$ , pois a função representa um custo ao aproximar  $d_n$  por  $y_n$  através de  $\mathbf{w}_n$ . Então a partir da derivada de  $J$  em relação a  $\mathbf{w}_n$ ,

$$\frac{\partial J(\mathbf{w}_n)}{\partial \mathbf{w}_n} = \frac{2}{2} (d_n - y_n) \frac{\partial y_n}{\partial \mathbf{w}_n} = -e_n \mathbf{h}_n, \quad (3.16)$$

obtem-se a regra de aprendizagem LMS recorrendo ao gradiente descendente estocástico recursivo, ou seja

$$\mathbf{w}_{n+1} = \mathbf{w}_n - \eta \frac{\partial J(\mathbf{w}_n)}{\partial \mathbf{w}_n} = \mathbf{w}_n + \eta e_n \mathbf{h}_n, \quad (3.17)$$

com  $0 < \eta \ll 1$  sendo a taxa de aprendizagem. Uma variante da regra LMS amplamente utilizada é a regra LMS normalizada (NLMS) dada pela seguinte expressão (CHOI *et al.*, 2006):

$$\mathbf{w}_{n+1} = \mathbf{w}_n + \frac{\eta}{\varepsilon + \|\mathbf{h}_n\|^2} e_n \mathbf{h}_n = \mathbf{w}_n + \frac{\eta}{\varepsilon + \mathbf{h}_n^T \mathbf{h}_n} e_n \mathbf{h}_n, \quad (3.18)$$

sendo  $\varepsilon$  uma constante positiva e muito pequena necessária para evitar a divisão por zero. Os pontos fortes dos algoritmos LMS e NLMS são sua fácil implementação e desempenho sob certas condições teóricas (WIDROW, 2005). A regra tem sido amplamente utilizado em processamento de sinais e filtragem adaptativa, mesmo que tais condições não sejam totalmente verificadas na prática. Para exemplificar, é utilizado em quase todo modem para equalização de canal e cancelamento de eco.

### 3.6 Regra LMS generalizada

A regra LMS mostrada na Eq. (3.17) utiliza um neurônio de saída linear. Considere agora, um neurônio não-linear, i.e.,  $y_n = \phi(u_n) = \phi(\mathbf{w}_n^T \mathbf{h}_n)$ , em que  $\phi: \mathbb{R} \rightarrow \mathbb{R}$  é uma função de ativação não-linear, tal como a função sigmóide logística. Para este tipo de neurônio de saída, a original regra LMS não é mais aplicável. Para a presente  $\phi$ , considere a função objetivo com neurônio de saída não-linear de acordo com o  $n$ -ésimo vetor de entrada apresentado:

$$J(\mathbf{w}_n) = \frac{1}{2} e_n^2 = \frac{1}{2} (d_n - y_n)^2 = \frac{1}{2} (d_n - \phi(u_n))^2 = \frac{1}{2} (d_n - \phi(\mathbf{w}_n^T \mathbf{h}_n))^2. \quad (3.19)$$

Então, com o auxílio da regra da cadeia, o vetor gradiente desta função é dado por

$$\frac{\partial J(\mathbf{w}_n)}{\partial \mathbf{w}_n} = \frac{\partial J}{\partial e_n} \frac{\partial e_n}{\partial y_n} \frac{\partial y_n}{\partial u_n} \frac{\partial u_n}{\partial \mathbf{w}_n} = e_n(-1)\phi'(u_n)\mathbf{h}_n, \quad (3.20)$$

com  $u_n = \mathbf{w}_n^T \mathbf{h}_n$  e  $\phi'(u_n) = \frac{\partial y_n}{\partial u_n}$ . Desta maneira, a regra LMS generalizada é dada por

$$\mathbf{w}_{n+1} = \mathbf{w}_n - \eta \frac{\partial J(\mathbf{w}_n)}{\partial \mathbf{w}_n} = \mathbf{w}_n + \eta e_n \phi'(u_n) \mathbf{h}_n, \quad (3.21)$$

sendo que o produto  $e_n \phi'(u_n)$  é chamado de gradiente local do neurônio de saída.

A derivada da função de ativação pode ser facilmente calculada em alguns casos especiais, como o caso da Eq. (2.8) dada pela Eq. (2.9).

Nesta dissertação, a versão original da rede NoProp, que usa um neurônio de saída linear, será denotada por NoProp. A variante desta rede que usa um neurônio não-linear com função de ativação sigmoide logística será denotada por NoProp-logsig. Vale lembrar que esta variante é uma proposta de modelagem da curva de potência de aerogeradores.

### 3.7 *Batch intrinsic plasticity (BIP)*

A presente subseção é baseada no trabalho de Freire (2015). Consiste na ideia de que, pode-se melhorar o desempenho das redes neurais randomizadas descritas nas sessões anteriores otimizando os parâmetros (inclinação e limiar) das funções de ativação.

O neurônio de McCulloch-Pitts, usado como elemento construtivo básico até os dias atuais de uma gama de arquiteturas de redes neurais artificiais, incluindo as redes randomizadas, toma como ponto de partida o neurônio biológico, e com isso em mente, ressalta-se que o vetor de pesos  $\mathbf{m}_j$ ,  $j = 1, \dots, q$ , representa as sinapses do neurônio. As sinapses de um neurônio conectam-se a vários outros neurônios. Uma boa parte de modelos de redes neurais artificiais são desenvolvidos para permitir a adaptação sináptica, ou seja, a plasticidade sináptica do neurônio, motivado pelo fato de que na formação da memória do cérebro normal adulto é essa plasticidade específica que predomina.

Biologicamente, tal plasticidade, ocorre para ajudar o neurônio a manter níveis adequados de atividade/excitabilidade elétrica, deslocando posições e/ou inclinações das curvas de resposta para manter as regiões sensíveis dessas curvas bem correspondidas com a distribuição dos estímulos de entrada. O que observou-se em estudos é que a distribuição dessa excitabilidade neuronal a estímulos possui picos em janelas de tempo com uma distribuição aproximadamente exponencial, e desempenha um papel regulador da plasticidade neuronal. A ideia matemática é tentar tornar a taxa de disparo média de cada neurônio oculto de uma rede

neural randomizada com distribuição aproximadamente exponencial. E como a distribuição exponencial tem maior entropia entre todas as distribuições com média fixa, tenta-se maximizar a informação transmitida enquanto mantém uma taxa média de disparo fixa, ou equivalentemente, minimiza essa taxa média de disparo enquanto carrega uma capacidade de informação fixa. Baseado nisso, desenvolveu-se o método *batch intrinsic plasticity* (BIP), o qual se propõe a regular a distribuição da taxa de disparo média do neurônio a partir do ajuste dos parâmetros da função de ativação sigmoideal de cada neurônio de uma rede neural randomizada, como as redes descritas neste capítulo.

Triesch (2005) propôs uma regra de aprendizagem baseada em plasticidade intrínseca por meio do gradiente para otimizar a transmissão de informação de um simples neurônio adaptando a inclinação  $a$  e o limiar  $b$  da função sigmoide logística de forma que os neurônios ocultos de saída tornam-se exponencialmente distribuídos. O BIP foi proposto baseado nesta ideia e usado para otimizar a camada oculta da rede ELM, calculando os parâmetros  $a$  e  $b$  de uma só vez, pois as entradas são apresentadas simultaneamente (NEUMANN e STEIL, 2011, apud FREIRE, 2015). A rede ELM possui pesos  $\mathbf{m}_j$ ,  $j = 1, \dots, q$ , ocultos randomizados, que possivelmente podem ter neurônios saturados ou respostas quase lineares o que compromete a capacidade de generalização. Porém, isso pode ser evitado utilizando funções de ativação com parâmetros  $a_j$  e  $b_j$  devidamente sintonizados de modo a evitar tais situações.

Considere a função sigmoide logística a seguir:

$$\phi(\mathbf{m}_j^T \mathbf{x}_n) = 1/(1 + \exp(a_j(\mathbf{m}_j^T \mathbf{x}_n) + b_j)), \quad (3.22)$$

em que  $j$  representa o  $j$ -ésimo neurônio oculto. A abordagem BIP consiste em forçar a ativação do  $j$ -ésimo neurônio oculto a se aproximar de uma função de distribuição exponencial desejada  $f_{des}$ . Desta maneira, para cada neurônio oculto, coleta-se a soma sináptica de entrada  $\mathbf{u}_j = \mathbf{m}_j^T \mathbf{X}$ , lembrando que a matriz  $\mathbf{X}$  é  $p \times N_1$ . Então alvos aleatórios virtuais  $\mathbf{T}_{f_{des}} = (t_1, \dots, t_{N_1})^T$  da distribuição exponencial de saída desejada são construídos, e ambos  $\mathbf{u}_j$  e  $\mathbf{T}_{f_{des}}$  são ordenados de forma crescente. Em seguida, constrói-se a matriz  $\Phi(\mathbf{u}_j) = (\mathbf{u}_j, (1, \dots, 1))^T$  e o vetor de parâmetros  $\mathbf{c}_j = (a_j, b_j)^T$ . A solução para o  $\mathbf{c}_j$  ótimo é obtida pela computação da pseudoinversa de Moore-Penrose, que é dada por

$$\mathbf{c}_j = (a_j, b_j)^T = (\Phi(\mathbf{u}_j)^T \Phi(\mathbf{u}_j) + \lambda \mathbf{I})^{-1} \Phi(\mathbf{u}_j)^T f^{-1}(\mathbf{T}_{f_{des}}), \quad (3.23)$$

com  $\lambda > 0$  denotando o parâmetro de regularização,  $\mathbf{I}^{q \times q}$  a matriz identidade, e  $f^{-1}$  a inversa

da função de ativação.

É importante notar que, no caso de usar sigmoide logística, os alvos  $\mathbf{T}_{f_{des}}$  estão em  $[0,1]$  e, por isso, somente distribuições de probabilidades truncadas são aplicadas. Segue o Algoritmo 1 com a abordagem BIP.

---

**Algoritmo 1** – *Batch Intrinsic Plasticity*

---

- 1: **Entrada:**  $\mathbf{X} = [\mathbf{x}_1 | \mathbf{x}_2 | \dots | \mathbf{x}_{N_1}]$ ;
  - 2: **para** cada neurônio oculto  $j$  **fazer**
  - 3: |  $\mathbf{u}_j = \mathbf{m}_j^T \mathbf{X}$ ; {coletar a soma sináptica com  $j = 1, \dots, q$ }
  - 4: |  $\mathbf{T}_{f_{des}} = (t_1, \dots, t_{N_1})^T$ ; {saída desejada de uma distribuição desejada  $f_{des}$ }
  - 5: | Ordenar em ordem crescente  $\mathbf{u}_j$  e  $\mathbf{T}_{f_{des}}$
  - 6: | Construir a matriz  $\Phi(\mathbf{u}_j) = (\mathbf{u}_j, (1, \dots, 1))^T$
  - 7: | Calcular a Eq. (3.23)
  - 8: **fim**
  - 9: **retornar**  $\mathbf{c} = \{\mathbf{a}, \mathbf{b}\}$ .
- 

### 3.8 Discussão e conclusão

Esta subseção é baseada em Huang, Zhu e Siew (2006). É importante ressaltar que teoricamente, a bom saber, as redes randomizadas de uma camada oculta, por ajustarem somente os pesos de saída são de treinamento muito rápido, diferentemente das tradicionais redes MPL com algoritmo de retropropagação as quais ajustam todos os pesos envolvidos. Hornick *et al* (1991) provou que em redes neurais que aproximam uma função contínua, se a função de ativação dos neurônios ocultos da rede é contínua, limitada e não constante, então a função contínua pode ser aproximada com acurácia arbitrária pela rede neural sobre um conjunto de entradas compacto. Leshno *et al* (1993) provou que redes *feedforward* com função de ativação não polinomiais podem aproximar funções contínuas. Huang e Babri (1998) mostraram que a tarefa de aproximar funções em um conjunto de treinamento finito, pode ser realizada por redes *feedforward* de uma camada oculta com no máximo  $N$  neurônios e com quase qualquer função de ativação não-linear, dadas  $N$  observações distintas de treinamento. Bartlett (1998) abordou teoricamente a generalização de redes *feedforward* em que concluiu que as redes que alcançaram menor erro de treinamento e menor norma dos pesos possuem

melhor desempenho de generalização.

Em geral, uma rede de camada oculta única, com função de ativação  $\phi$  infinitamente diferenciável requer um número  $q$  de neurônios ocultos com  $q \leq N$ , e que, dado  $\varepsilon \geq 0$  obtém-se  $\|\mathbf{y} - \mathbf{d}\| < \varepsilon$  com  $q \leq N$  (HUANG, ZHU e SIEW (2006, p. 491), respectivamente Teorema 2.1 e Teorema 2.2). Importante também destacar que a solução por mínimos quadrados regularizados (Tikonov) é a solução que minimiza a norma quadrática dos pesos  $\mathbf{w}$ , que contém os menores erros  $\sum_{i=1}^N \|d_i - y_i\| = \varepsilon$ ,  $\varepsilon \geq 0$ , e é única.

O presente capítulo mostrou os principais fundamentos básicos para as redes randomizadas do tipo *feedforward* e com uma única camada. Foram apresentadas três variações das redes, a saber, as redes RVFL, ELM e NoProp, bem como o método de otimização dos parâmetros das funções de ativação BIP. Em relação às redes RVFL e ELM, que a estimação dos pesos de saída se dá pelo uso do método MQO ou MQR. Já para a rede NoProp, os pesos de saída são estimados pelo algoritmo LMS, que é uma técnica recursiva de estimação de parâmetros baseada no método gradiente descendente. O método BIP foi detalhado e seu uso é indicado para melhorar o desempenho de aproximação de redes neurais randomizadas. Ainda, encerramos com comentários teóricos pertinentes o que justifica e motiva o uso desse tipo de redes neurais artificiais. O seguinte capítulo tratará de simulações usadas para este trabalho.

## 4 METODOLOGIA DE SIMULAÇÃO

A proposta deste capítulo é descrever como os experimentos foram realizados. A tarefa da modelagem de curva de potência de aerogeradores foi realizada a partir de modelos individuais de turbinas eólicas. Em geral, para um aerogerador, tem-se  $N$  dados de entrada-saída das medidas de velocidade do vento (em m/s) e sua respectiva potência de saída (em kW). Para uso dos modelos, algoritmos e índices de validação numérica, considere que a velocidade do vento é  $x_n$  e sua respectiva potência de saída  $d_n$ . No capítulo anterior foram descritas técnicas cujos desempenhos serão avaliados nesta dissertação. A uma solução com índices de validação numérica considerados bons, mas sendo incorreta por gerar soluções não factíveis dá-se o nome de solução patológica. As técnicas antes descritas serão avaliadas de maneiras exaustivas, considerando inclusive cenários em que foram utilizadas sem e com o auxílio do método BIP.

Uma série de experimentos foram realizados para avaliar os modelos descritos anteriormente. A ideia principal é encontrar um ou mais modelos que satisfaçam todos os requisitos da tarefa de modelagem CPA do ponto de vista do usuário final do modelo e da solução por ele fornecida, especialmente o(a) engenheiro(a) responsável por garantir a produção de energia requerida. Com base em nossa experiência, tais requisitos são modelos rápidos e acurados. Existe pouco espaço para incertezas na modelagem CPA. Além disso, não são permitidas soluções patológicas (baixo valor do RMSE, mas curva de potência visualmente incorreta; ou seja, gerando valores não factíveis de potência) porque o setor elétrico é altamente regulamentado, de modo que grandes impostos e multas são aplicados às companhias de geração que não fornecem a potência contratada.

### 4.1 Índices de validação e transformação escalar

A qualidade do ajuste é observado numericamente por dois índices numéricos a raiz do erro médio quadrático (RMSE, *root mean squared error*) e o coeficiente de determinação  $R^2$ , que são dados, respectivamente, por

$$RMSE = \sqrt{\frac{1}{N} \sum_{n=1}^N (d_n - \hat{y}_n)^2} \quad \text{e} \quad R^2 = 1 - \frac{\sum_{n=1}^N (d_n - \hat{y}_n)^2}{\sum_{n=1}^N (d_n - \bar{d})^2}, \quad (4.1)$$

sendo  $N$  a quantidade de dados a serem avaliados, com  $d_n$  a  $n$ -ésima medida de potência,  $\hat{y}_n$  denotando o valor predito de potência, e  $\bar{d}$  a média amostral da potência. Observe que para o índice RMSE quanto mais próximo de 0 melhor a qualidade do ajuste da curva aos dados. Por

outro lado, o índice  $R^2$  quanto mais próximo de 1 melhor.

Para a utilização dos modelos com função de ativação sigmoide logística, os dados foram transformados para a escala no intervalo  $[0,1]$  por meio das seguintes transformações lineares:

$$x_{trf} = \frac{x - x_{min}}{x_{max} - x_{min}} \quad \text{e} \quad d_{trf} = \frac{d - d_{min}}{d_{max} - d_{min}}, \quad (4.2)$$

sendo que os subscritos *min* e *max* representam o mínimo e o máximo da correspondente variável, e *trf* para indicar que a variável está na escala  $[0,1]$ . Após a estimação, o valor predito é transformado para a escala original por meio da seguinte expressão:

$$\hat{y} = d_{min} + \hat{y}_{trf}(d_{max} - d_{min}). \quad (4.3)$$

## 4.2 Modelos avaliados

Um experimento com modelo de regressão polinomial foi utilizado como base de referência para os testes com os demais modelos RNRs. Durante os experimentos, cada modelo foi avaliado utilizando para treinamento  $N = 100\%$  dos dados. Cabe aqui uma explicação para a escolha desta abordagem. O objetivo da tarefa é construir um modelo para ajuste de curva de potência aos dados e explorá-la visualmente em busca de soluções patológicas. A predição acurada de valores de potência para novos dados em um conjunto de teste é feito apenas para os melhores modelos avaliados visualmente. Para este cenário, é realizada uma divisão dos dados em dois subconjuntos, um de treino (com 80% das observações) e outro de teste (com 20% das observações). A acurácia no conjunto de teste é avaliada pelo RMSE. Cada modelo em rede neural randomizada foi individualmente avaliado com o total de 20 rodadas independentes de treinamento e ao final das rodadas computou-se a média aritmética dos índices de validação para cada modelo, tempo computacional (em segundos), mínimo, máximo e desvio padrão. Também observou-se graficamente a ocorrência de solução patológica – baixo valor do RMSE mas solução visual incorreta.

Nesta dissertação, avaliou-se um total de 12 variantes das redes ELM, RVFL e NoProp. Uma listagem dos modelos avaliados com breves descrições de suas arquiteturas e regras de aprendizagem está mostrada abaixo:

- a) ELM: Modelo ELM original proposto por Huang, Zhu e Siew (2006) com neurônio de saída linear e regra MQO para estimar os pesos do neurônio de saída;
- b) ELM-BIP: Variante do modelo ELM original mais o algoritmo BIP para otimizar

- os parâmetros da função de ativação dos neurônios ocultos;
- c) NoProp: Modelo NoProp original. Foi proposto por Widrow, Greenblatt, Kim e Park (2013) com neurônio de saída linear e regra LMS para estimar os pesos do neurônio de saída;
  - d) NoProp-BIP: Variante do modelo NoProp original mais o algoritmo BIP para otimizar os parâmetros da função de ativação dos neurônios ocultos;
  - e) NoProp-logsig: Variante do modelo NoProp com neurônio de saída com função de ativação sigmoide logística, regra LMS generalizada para estimar os pesos do neurônio de saída;
  - f) NoProp-logsig-BIP: Variante do modelo NoProp-logsig mais o algoritmo BIP para otimizar os parâmetros da função de ativação dos neurônios ocultos;
  - g) RVFL: Modelo RVFL original proposto por Pao, Park e Sobajic (1994) com neurônio de saída linear e regra MQO para estimar os pesos do neurônio de saída;
  - h) RVFL-BIP: Variante do modelo RVFL original mais o algoritmo BIP para otimizar os parâmetros da função de ativação dos neurônios ocultos;
  - i) RVFL-linear: Variante do modelo RVFL original com neurônio de saída linear regra LMS para estimar os pesos do neurônio de saída;
  - j) RVFL-linear-BIP: Variante do modelo RVFL-linear mais o algoritmo BIP para otimizar os parâmetros da função de ativação dos neurônios ocultos;
  - k) RVFL-logsig: Variante do modelo RVFL com neurônio de saída com função de ativação sigmoide logística, regra LMS generalizada para estimar os pesos do neurônio de saída;
  - l) RVFL-logsig-BIP: Variante do modelo RVFL-logsig mais o algoritmo BIP para otimizar os parâmetros da função de ativação dos neurônios ocultos.

### 4.3 Ambiente computacional e conjunto de dados

Para realizar os experimentos, o sistema operacional utilizado foi *Windows 10 Home Single Language*, com códigos e algoritmos implementados e obtidos a partir da linguagem de programação *Python 3.6.4* em *Jupyter 3.4.0* com bibliotecas de programação científica *NumPy 1.14.2* e *SciPy 1.0.0*, com códigos de gráficos e figuras utilizando as bibliotecas *Matplotlib 2.2.2* e *Seaborn 0.8.1* obtidos pela distribuição gratuita padrão do ambiente *Anaconda*. Com processador Intel(R) Core(TM) i7-5500U e CPU @ 2,40GHz, e me-

mória RAM de 16GB, com um tipo de arquitetura de 64bits.

Dois conjuntos de dados foram utilizados. O primeiro foi obtido a partir de quatro estações meteorológicas, denominadas de WT1 a WT4, e dois mastros meteorológicos. Nesta dissertação, usou-se apenas o conjunto de dados da WT1 para os experimentos. Em particular, WT1 possui ao todo 47.542 pares de entrada-saída  $(x_n, d_n)$  de velocidade do vento  $x$  e potência de saída  $d$ , coletados em 30/07/2010 a 31/07/2011. Dados ambientais de  $x$  foram coletados por sensores nos mastros meteorológicos, enquanto a potência de saída  $d$  foi medida no aerogerador. Como cada mastro meteorológico possui duas turbinas eólicas associadas, os valores de  $x$  medidos no mastro são emparelhados com os valores medidos de  $d$  dessas turbinas associadas. As amostras de dados disponíveis correspondem aos valores médios em intervalos de 10 minutos. As especificações do parque eólico responsável pelos dados medidos são mostrados na Tabela 1. Para certas entradas, os valores são aproximados a fim de proteger a identidade dos aerogeradores e parque eólico. Mais detalhes sobre este conjunto de dados podem ser obtidos em Lee *et al.* (2015).

Tabela 1 – Especificações do parque eólico dos WT's

<b>Características</b>	<b>Valores</b>
Número de mastros meteorológicos	Múltiplos
Número de aerogeradores	> 200
Altura do eixo	80 m
Diâmetro do rotor	≈ 80 m
Vel. de corte do vento - entrada	3,5 m/s
Vel. de corte do vento – saída	20 m/s
Vel. média do vento ( $\pm$ desv. padrão)	8,2 $\pm$ 2,9 m/s
Velocidade nominal do vento	≈ 13 m/s
Potência nominal	1,5 a 2,0 MW
Potência média ( $\pm$ desv. padrão)	47,5 $\pm$ 34,0 MW
Local	Inland US

Fonte: Adaptada de Lee *et al.* (2015).

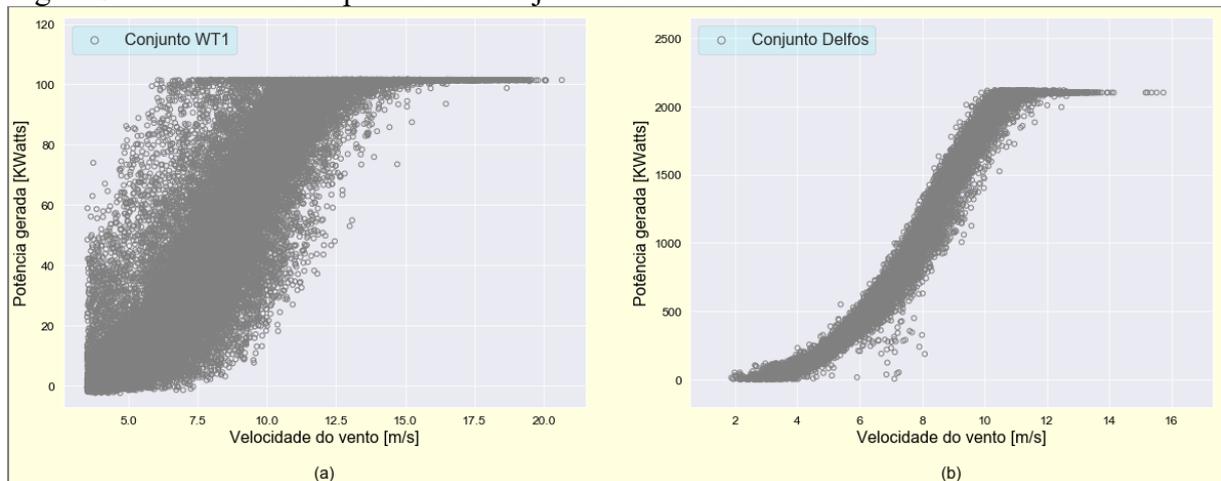
Este primeiro conjunto de dados foi escolhido por impor diversas mudanças ao problema de regressão usado para a modelagem de CPA. Primeiramente, é um conjunto de dados muito ruidoso, no sentido de que a dispersão em torno dos valores medidos de velocidade e potência é alta. Em segundo, o conjunto de dados é consideravelmente grande (maior que

40.000 amostras), e mesmo com o alto poder de processamento nos dias atuais, esse número ainda apresenta dificuldades para alguns tipos de modelos sofisticados de aprendizado de máquina Virgolino *et al.* (2020).

O segundo conjunto de dados, doravante referido como Delfos<sup>2</sup>, que foi gentilmente cedido por uma empresa privada brasileira que atualmente desenvolve soluções baseadas em inteligência artificial para a indústria de energia – Eólica, Óleo e Gás, e nos cedeu os dados para propósitos científicos. Este conjunto de dados foi originalmente utilizado em Virgolino *et al.* (2020) para modelagem de curva de potência robusta com presença de *outliers*. Possui no total  $N = 11.552$  valores de entrada-saída, e a ideia de incluir este conjunto de dados está em validar a hipótese de que soluções patológicas possivelmente ocorrem para diferentes dados, tal que, os usuários devem dar a devida atenção a tais ocorrências.

A Figura 5 mostra os gráficos de dispersão dos dois conjuntos de dados utilizados nesta dissertação.

Figura 5 – Gráficos de dispersão dos conjuntos de dados amostrais



Fonte: Elaborada pelo autor.

(a) Gráfico de dispersão do conjunto WT1; (b) Gráfico de dispersão do conjunto Delfos.

<sup>2</sup> Delfos IM (DELLOS INTELLIGENT MAINTENANCE, 2017-). Disponível em: "<http://www.delfosim.com/>". Acesso em: 15 de janeiro de 2022.

## 5 SIMULAÇÕES E RESULTADOS

Uma série de experimentos foram realizados com os modelos antes descritos. A partir da metodologia e ambiente computacional previamente estabelecidos procurou-se os melhores resultados bem como uma precisa interpretação dos resultados obtidos pela ótica metodológica escolhida. Resultados individuais dos modelos e de comparação serão apresentados a seguir.

### 5.1 Resultados do modelo de regressão polinomial

O modelo de regressão polinomial provavelmente é a primeira escolha para a estimar a curva de potência para modelagem de CPA através de modelos de regressão. Duas razões para isso são a alta flexibilidade do modelo para se ajustar aos dados e sua ampla disponibilidade em *softwares* populares, tais como *Matlab*, *R* e *MS Excel*. Contudo, como será visto nos primeiros experimentos, o modelo de regressão polinomial apresenta baixo desempenho para os conjuntos de dados utilizados.

Para os experimentos que seguem, o grau do polinômio regressor do modelo  $q$  varia de 4 até 7. Para cada valor de  $q$  o processo de estimação é realizado para  $T = 20$  rodadas de treinamento. Os  $q + 1$  parâmetros foram estimados utilizando o método MQO. Os resultados serão mostrados na Tabela 2.

Tabela 2 – Índices de desempenho numérico para o modelo de regressão polinomial de grau  $q$  – conjunto WT1

Grau	RMSE	$R^2$	TC (s)
$q = 4$	13,0675	0,8519	0,0358
$q = 5$	13,0490	0,8523	0,0373
$q = 6$	13,0378	0,8525	0,0428
<b><math>q = 7</math></b>	<b>13,0295</b>	<b>0,8527</b>	<b>0,0501</b>

Fonte: Elaborada pelo autor.

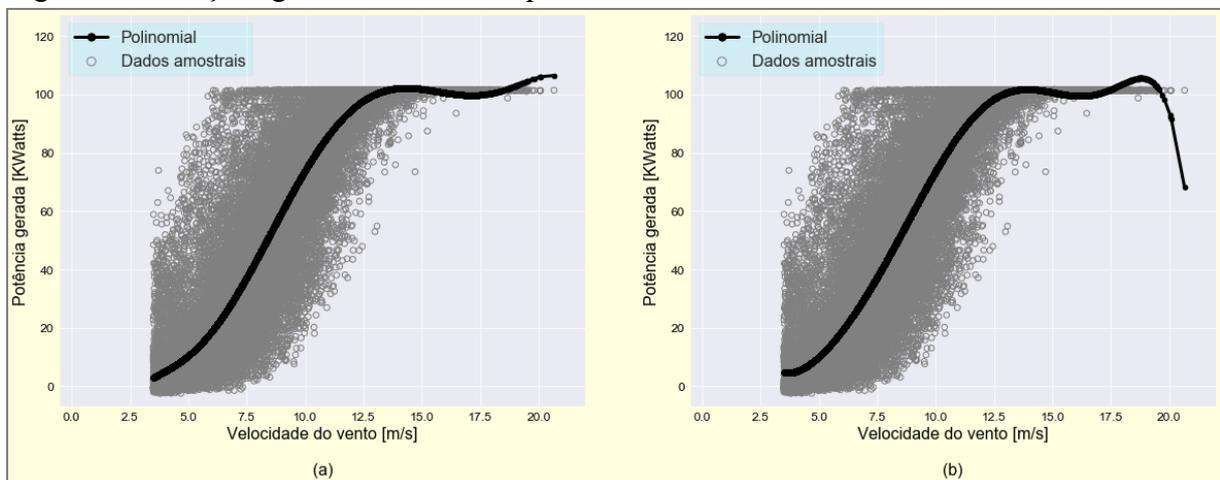
Nota: Valores da média aritmética do RMSE,  $R^2$  e tempo computacional (TC, em segundos).

Como pode ser observado, a discrepância dos valores RMSE e  $R^2$  para os diferentes modelos é praticamente inexistente. Isto era de se esperar pois não existe inicialização dos pesos e nenhum método iterativo envolvido para estimação dos parâmetros. No entanto, mesmo

assim, decidimos mostrar tal discrepância para comparar com os resultados produzidos pelos modelos de redes neurais cujo processo de estimação envolve algoritmos iterativos-recursivos de aprendizagem. Um olhar mais atento à Tabela 2 revela que a diferença entre RMSE e  $R^2$  é muito pequena para diferentes valores de  $q$ , o mesmo ocorrendo para o tempo computacional (TC). O polinômio de grau 7 obteve o melhor desempenho pelos índices RMSE e  $R^2$ . Contudo, é importante destacar uma particularidade da tarefa de modelagem de CPA, em especial aquela com um grande número de amostras ruidosas, como estamos abordando nesta dissertação.

No que tange a hipótese de trabalho desta dissertação, se basear apenas pelos índices de validação numérica pode ser enganoso no sentido de que a curva de potência estimada (a solução de fato do problema) pode ser inadequada para fins práticos. Em outras palavras, o ajuste de curva aos dados observados pode não estar correto, apesar das indicações dos índices numéricos de desempenho. Doravante, este tipo de ocorrência será chamada de solução patológica. Então, uma inspeção visual da curva estimada deve ser realizada. Para exemplificar a inspeção visual, as curvas estimadas pelo modelo com  $q = 6$  e  $q = 7$  são mostradas na Figura 6 respectivamente.

Figura 6 – Soluções gráficas do modelo polinomial



Fonte: Elaborada pelo autor.

(a) Solução patológica produzida pelo modelo polinomial de ordem  $q = 6$ ; (b) Solução patológica produzida pelo modelo polinomial de ordem  $q = 7$  (melhor modelo pelos índices numéricos – RMSE e  $R^2$ ).

Pode-se inferir a partir dessas figuras que ambas as soluções são inadequadas ao problema de interesse, sendo consideradas soluções patológicas. Uma explicação plausível está relacionada a uma provável instabilidade numérica causada por polinômios de grau elevado. Para estimar os parâmetros do modelo de regressão polinomial, deve-se construir uma matriz de Vandermonde, cuja  $j$ -ésima coluna é formada pelas velocidades do vento elevada a  $(j - 1)$ -

ésima potência com  $j = 1, \dots, k + 1$ , sendo  $k$  o grau do polinômio. Como a velocidade do vento atinge valores maiores que 10m/s, a matriz de Vandermonde torna-se mal-condicionada à medida que  $k$  aumenta, causando um ajuste inadequado da curva aos dados nas regiões com poucas medidas, por exemplo, na parte superior esquerda da figura.

## 5.2 Resultados do modelo MLP

Para o problema de interesse, especificou-se uma topologia de rede neural com uma camada oculta, uma única entrada (a velocidade do vento), e uma única saída (a potência gerada). Em resumo, denotou-se tal arquitetura como  $MLP(1, q, 1)$ , sendo  $q$  o número de neurônios ocultos, um hiperparâmetro do modelo MLP que tem que ser especificado antes do treinamento. Para este propósito, buscou-se o valor adequado de  $q$  no intervalo dos naturais  $2 \leq q \leq 5$ , usando todos os dados para treinamento, pois nosso objetivo é o ajuste de curva ao total de observações medidas. O algoritmo padrão de treinamento escolhido foi o de retropropagação, com a taxa de aprendizagem igual 0,1 para o modelo com saída linear e 0,01 para o modelo com saída não-linear, e foi realizada uma mudança de escala dos dados para  $[0,1]$ . A Tabela 3 contém resultados dos índices com o modelo MLP.

Tabela 3 – Índices de desempenho numérico para o modelo  $MLP(1, q, 1)$  – conjunto WT1

Topologia	RMSE	$R^2$	TC (s)
<i>MLP(1,2,1)-linear</i>	13,6244	0,8383	68,9715
<i>MLP(1,3,1)-linear</i>	13,6162	0,8388	60,5336
<i>MLP(1,4,1)-linear</i>	13,4027	0,8440	58,0806
<b><i>MLP(1,5,1)-linear</i></b>	<b>13,3731</b>	<b>0,8447</b>	<b>60,7115</b>
<i>MLP(1,2,1)-logsig</i>	13,1563	0,8498	66,4221
<i>MLP(1,3,1)-logsig</i>	13,1379	0,8503	66,4345
<i>MLP(1,4,1)-logsig</i>	13,1259	0,8505	68,6215
<b><i>MLP(1,5,1)-logsig</i></b>	<b>13,1207</b>	<b>0,8506</b>	<b>71,0139</b>

Fonte: Elaborada pelo autor.

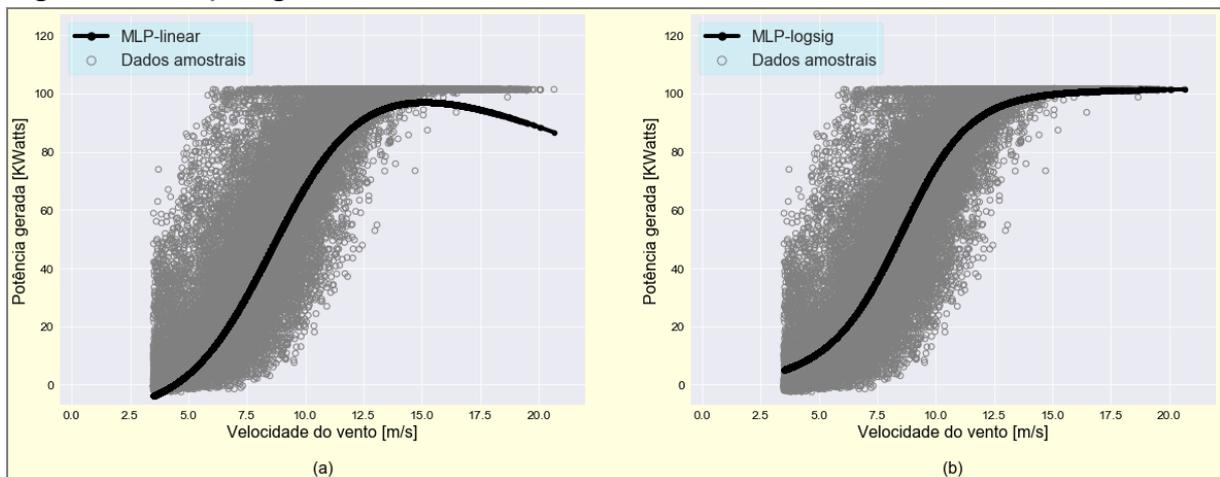
Nota: Modelo MLP com função de ativação linear e sigmoideal do neurônio de saída e valores da média aritmética do RMSE,  $R^2$  e tempo computacional (TC, em segundos).

Foram avaliadas duas situações envolvendo o neurônio de saída. Primeiramente, o modelo MLP foi treinado com função de ativação linear para o neurônio de saída. Depois,

modificou-se a função de ativação do neurônio de saída para a função sigmoide logística. A Tabela 3 mostrou os resultados numéricos dos RMSE,  $R^2$  e tempo computacional (TC) para quantidade de neurônios ocultos  $q = 2, \dots, 5$ . Pode ser visto, diferentemente do modelo polinomial, que existe uma discrepância relevante, porém ainda pequena, para os valores sobre as 20 rodadas independentes devido a inicialização dos pesos.

De acordo com os valores RMSE e  $R^2$ , os melhores resultados numéricos são os dos modelos  $MLP(1,5,1)$ -linear e  $MLP(1,5,1)$ -logsig. Sem dúvida que os resultados são estatisticamente equivalentes. Assim, pode-se concluir que qualquer um desses modelos é um bom modelo. Esse resultado está de acordo com recomendações amplamente aceitas de utilizar neurônio com saída linear em problemas de regressão, que é difundida por vários livros-textos sobre redes neurais e pacotes de *softwares*. Contudo, o modelo  $MLP(1,5,1)$ -linear produz soluções patológicas que não são reveladas pelos índices de desempenho numéricos escolhidos, conforme pode ser visto na Figura 7.

Figura 7 – Soluções gráficas do modelo MLP



Fonte: Elaborada pelo autor.

(a) Solução patológica produzida pelo modelo  $MLP(1,5,1)$ -linear; (b) Típica solução aceitável pelo modelo  $MLP(1,5,1)$ -logsig.

As soluções patológicas são detectadas somente por inspeção visual vide Figura 7a, como mostrado nessa solução produzida pelo modelo  $MLP(1,5,1)$ -linear. Com frequência tal modelo produz as soluções patológicas. Já o modelo  $MLP(1,5,1)$ -logsig não produz soluções patológicas – ver Figura 7b, entre todas, porque sua saída sigmoide já segue o formato típico de uma curva de potência de um aerogerador, sendo assim um modelo mais adequado a tarefa de modelagem de CPA.

### 5.3 Resultados do modelo ELM

Para este modelo usa-se a notação  $ELM(1, q, 1)$  por simplicidade. Os neurônios ocultos da rede ELM utilizam a função de ativação sigmoide logística. A mudança de escala dos dados para o intervalo  $[0,1]$  não é necessária. A quantidade de neurônios ocultos  $q \in \{5,10,15\}$  foi utilizada. O grande valor de  $q$  necessário para a rede ELM, em comparação ao utilizado na rede MLP, ocorre principalmente por sua natureza randomizada, a fim de compensar a perda de eficiência causada pela substituição do ajuste dos pesos da camada de entrada por uma inicialização aleatória sem correção de valores. A utilização do BIP é destacada como  $ELM(1, q, 1)$ -BIP. Por um lado, a randomização dos parâmetros acelera a velocidade de treinamento da rede, mas por outro, não garante bons resultados para pequenos valores de  $q$ . Os resultados numéricos para os RMSE e  $R^2$  são descritos na Tabela 4.

Tabela 4 – Índices de desempenho numérico para o modelo  $ELM(1, q, 1)$  – conjunto WT1

Topologia	RMSE	$R^2$	TC (s)
$ELM(1,5,1)$	13,0841	0,8515	0,0198
<b><math>ELM(1,10,1)</math></b>	<b>13,0225</b>	<b>0,8529</b>	<b>0,0364</b>
$ELM(1,15,1)$	13,0201	0,8529	0,0663
$ELM(1,5,1)$ -BIP	17,6598	0,6990	0,0413
$ELM(1,10,1)$ -BIP	17,1740	0,7169	0,0803
$ELM(1,15,1)$ -BIP	21,2285	0,5727	0,1262

Fonte: Elaborada pelo autor.

Nota: Modelo ELM com função de ativação sigmoide do neurônio oculto e valores da média aritmética do RMSE,  $R^2$  e tempo computacional (TC, em segundos).

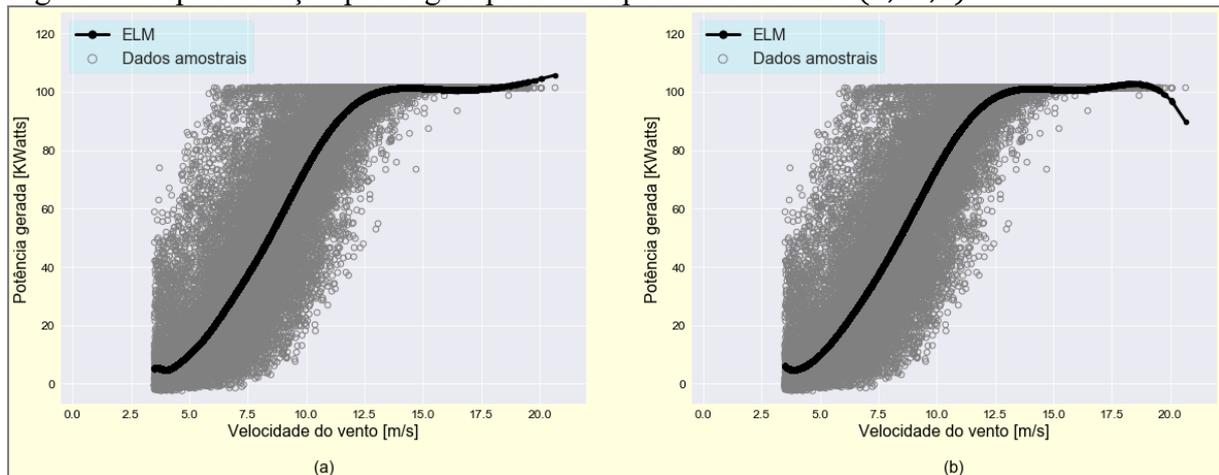
Os melhores índices de desempenho são para  $q = 10$  e  $q = 15$ , sem o BIP. No entanto, é interessante observar que para esses valores de  $q$  os resultados são estatisticamente equivalentes quando utilizado o modelo sem o BIP. Neste caso, dá-se então preferência ao modelo com menor arquitetura, a saber, o  $ELM(1,10,1)$  destacado em negrito. Este modelo de regressão baseado em rede ELM alcançou melhor desempenho do que o modelo neural MLP. O modelo com o BIP, ou seja, ELM-BIP, gerou resultados insatisfatórios, ou seja, raramente produziu soluções aceitáveis. Ressalta-se, entretanto, que apesar dos resultados numéricos semelhantes (modelo sem BIP), deve-se ter atenção quanto as ocorrências de soluções

patológicas como no modelo baseado em MLP. De fato, as soluções patológicas para o modelo ELM ocorrem muito frequentemente (de 10 a 11 vezes em 20 execuções).

A geração de soluções patológicas pelo modelo ELM pode ser explicada considerando dois aspectos. A primeira, está relacionada à arquitetura do modelo, que nesse sentido, segue a mesma do modelo  $MLP(1, q, 1)$ -linear que também gerou soluções patológicas. A segunda, está relacionada a natureza randomizada da arquitetura da rede ELM. Embora a randomização dos pesos de entrada da camada oculta acelere o processamento da rede, também favorece ocorrência patológica de soluções. Em outras palavras, para algumas seleções de pesos de entrada da camada oculta, há produção de soluções patológicas. Isto equivale a dizer que o presente modelo ELM é muito sensível à inicialização dos pesos.

A Figura 8 a seguir exemplifica graficamente uma solução patológica produzida pelo modelo ELM.

Figura 8 – Típica solução patológica produzida pelo modelo  $ELM(1,10,1)$



Fonte: Elaborada pelo autor.

#### 5.4 Resultados dos modelos NoProp e NoProp-logsig

Para estes experimentos mantém-se a notação padrão anterior, ou seja, o modelo  $NoProp(1, q, 1)$  representará a rede NoProp com uma entrada,  $q$  neurônios ocultos com função de ativação sigmoidal e 1 neurônio de saída com função de ativação linear. Semelhantemente,  $NoProp(1, q, 1)$ -logsig é a notação para a rede NoProp com função de ativação sigmoidal na camada oculta e de saída. Por simplicidade, foram treinadas as variantes deste modelo com 10 neurônios ocultos, pois foi o número de neurônios com o melhor desempenho pelos índices numéricos para a rede ELM cujos resultados foram discutidos na Seção 5.3. A taxa de

aprendizagem foi definida com  $\eta = 0,1$  para todas as variantes com ou sem BIP e foram escolhidas 20 épocas de treinamento. Para  $NoProp(1, q, 1)$  com ou sem BIP foi utilizada a regra LMS de aprendizagem da Eq. (3.17), e, de maneira semelhante,  $NoProp(1, q, 1)-logsig$  foi treinada com a regra LMS generalizada mostrada na Eq. (3.21). Os resultados numéricos são apresentados na Tabela 5 a seguir.

Tabela 5 – Índices de desempenho numérico para o modelo NoProp – conjunto WT1

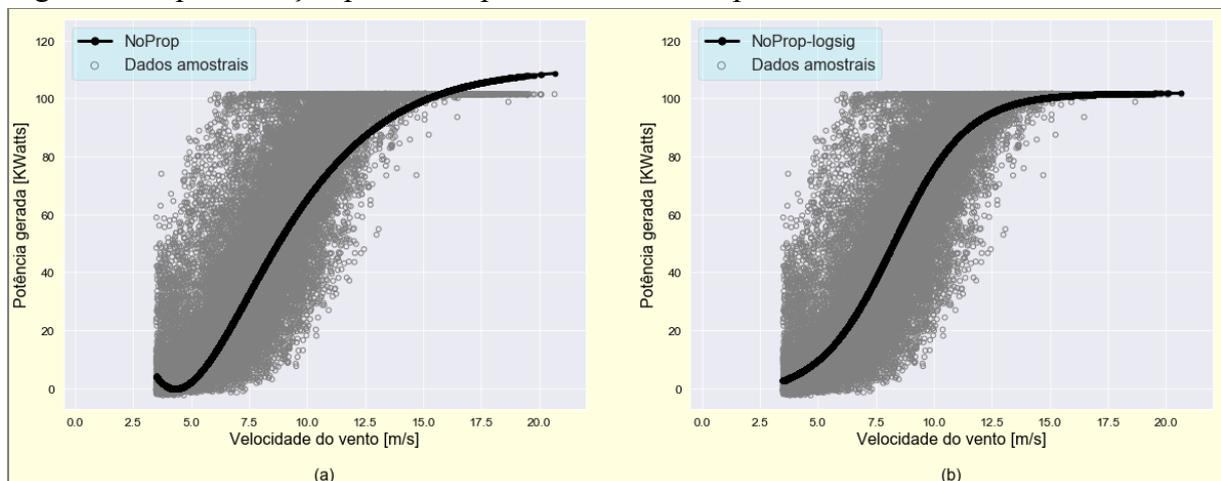
Topologia	RMSE	$R^2$	TC (s)
<b><math>ELM(1,10,1)</math></b>	<b>13,0225</b>	<b>0,8529</b>	<b>0,0364</b>
$NoProp(1,10,1)$	32,7612	0,0685	13,5490
<b><math>NoProp(1,10,1)-logsig</math></b>	<b>13,1586</b>	<b>0,8498</b>	<b>21,1167</b>
$NoProp(1,10,1)-BIP$	26,8816	0,2545	13,0954
<b><math>NoProp(1,10,1)-logsig-BIP</math></b>	<b>20,2902</b>	<b>0,6047</b>	<b>23,4157</b>

Fonte: Elaborada pelo autor.

Nota: Modelo NoProp com  $NoProp(1,10,1)$  com função de ativação linear do neurônio de saída e  $NoProp(1,10,1)-logsig$  com função de ativação sigmoideal do neurônio de saída.

Considerando os índices RMSE e  $R^2$ , nenhuma das versões do modelo NoProp de fato obteve bom desempenho com a adição do BIP. A versão linear conseguiu um ganho de desempenho, porém ainda é baixo, e a versão sigmoideal não manteve o bom desempenho do modelo  $NoProp(1,10,1)-logsig$  e sim piorou. A Figura 9 a seguir mostra típicas soluções pelo modelo NoProp (sem BIP).

Figura 9 – Típica solução produzida pelo modelo NoProp



Fonte: Elaborada pelo autor.

(a)  $NoProp(1,10,1)$ ; (b)  $NoProp(1,10,1)-logsig$ .

As variantes do modelo NoProp com ou sem BIP, e NoProp-logsig com BIP possuem um desempenho numérico inferior se comparado aos modelos ELM e NoProp-logsig. Isso é confirmado pela Figura 9a onde a curva de potência estimada é bastante incorreta. Os resultados do NoProp-logsig são melhores em termos numéricos, mas tais índices numéricos à primeira vista se mostram piores do que os do modelo ELM. Os tempos computacionais são bem superiores àqueles do modelo ELM. Deve-se notar, no entanto, que o modelo NoProp-logsig nunca produziu uma solução patológica ao longo das 20 execuções de treinamento. Uma típica solução pelo modelo *NoProp(1,10,1)-logsig* é exemplificada pela Figura 9b.

### 5.5 Resultados dos modelos RVFL

Foi mantido para esse modelo a mesma estrutura que as das redes ELM e NoProp. A notação  $RVFL(1, q, 1)$  representará a rede RVFL original com uma entrada,  $q$  neurônios ocultos e 1 neurônio de saída com função de ativação linear. Para estimar os pesos desta variante  $RVFL(1, q, 1)$  foi utilizada a Eq. (3.11), que faz uso da pseudoinversa. O modelo  $RVFL(1, q, 1)$ -LMS possui aprendizagem sequencial dos pesos  $\mathbf{w}$  pela regra LMS mostrada na Eq (3.17) com neurônio de saída com função de ativação linear. Não é preciso mudar a escala dos dados, e taxa de aprendizagem  $\eta = 10^{-6}$  (baixa pois favoreceu a ocorrência de bons índices numéricos). Para o modelo  $RVFL(1, q, 1)$ -logsig a rede também é formada a partir de Eq. (3.10), tem-se uma aprendizagem sequencial dos pesos  $\mathbf{w}$  pela regra LMS generalizada mostrada na Eq. (3.21), com taxa de aprendizagem  $\eta = 0,1$  e mudança de escala dos dados para o intervalo  $[0,1]$ . Para os modelos sequenciais permanece ainda 20 épocas de treinamento. A Tabela 6 a seguir exhibe seus resultados numéricos.

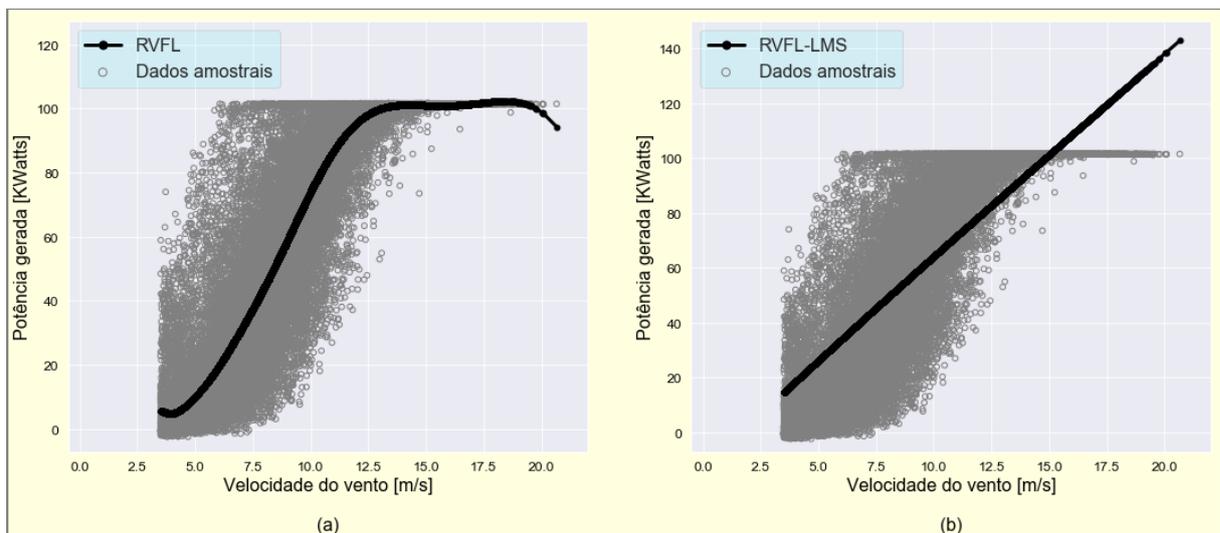
Tabela 6 – Índices de desempenho numérico para o modelo RVFL (variantes) – conjunto WT1

Topologia	RMSE	$R^2$	TC (s)
<b><i>ELM(1,10,1)</i></b>	<b>13,0225</b>	<b>0,8529</b>	<b>0,0364</b>
<b><i>NoProp(1,10,1)-logsig</i></b>	<b>13,1586</b>	<b>0,8498</b>	<b>21,1167</b>
<b><i>RVFL(1,10,1)-logsig</i></b>	<b>13,1298</b>	<b>0,8504</b>	<b>21,2288</b>
<i>RVFL(1,10,1)-LMS</i>	23,6678	0,5096	18,2788
<b><i>RVFL(1,10,1)</i></b>	<b>13,0220</b>	<b>0,8529</b>	<b>0,0521</b>
<b><i>RVFL(1,10,1)-logsig-BIP</i></b>	<b>13,2520</b>	<b>0,8476</b>	<b>22,6567</b>
<i>RVFL(1,10,1)-LMS-BIP</i>	16,2424	0,7695	13,5555
<b><i>RVFL(1,10,1)-BIP</i></b>	<b>13,9082</b>	<b>0,8317</b>	<b>0,0863</b>

Fonte: Elaborada pelo autor.

Na Tabela 6 anterior, os modelos  $RVFL(1, q, 1)$  e  $RVFL(1, q, 1)-LMS$  possuem função de ativação linear no neurônio de saída, e  $RVFL(1, q, 1)-logsig$  possui função de ativação sigmoideal no neurônio de saída. Vê-se que os índices de desempenho numéricos das variantes dos modelos RVFL-LMS com ou sem BIP é baixo. Para as variantes com função de ativação linear a CPA estimada pela regra MQO e sigmoideal com ou sem BIP possuem bons índices de desempenho numérico, semelhantes àqueles das redes ELM e NoProp-logsig de melhor desempenho. Como já anteriormente destacado, os modelos de rede com saída linear apresentam índices de desempenho numérico satisfatórios, porém, a partir de inspeção gráfica, pode-se constatar que as soluções muitas vezes obtidas são patológicas, não podendo ser aceitas como soluções factíveis pelo modelo. Os valores numéricos para as variantes RVFL-logsig, com ou sem BIP, exibem valores satisfatórios quando comparados aos modelos ELM e NoProp-logsig com melhor desempenho. Ressalta-se que o modelo RVFL-logsig, com ou sem BIP, com base nos resultados observados nos experimentos não produziu soluções patológicas, mesmo com o pré-processamento realizado antes da estimação com o BIP. Uma solução típica do RVFL-logsig é semelhante àquela mostrada na Figura 8b para o modelo  $NoProp(1,10,1)-logsig$ . Uma típica solução patológica com o modelo RVFL (MQO ou LMS, com ou sem BIP) é mostrada na Figura 10.

Figura 10 – Típica solução patológica produzida pelo modelo RVFL com neurônio de saída linear



Fonte: Elaborada pelo autor.

(a)  $RVFL(1,10,1)$ ; (b)  $RVFL(1,10,1)-LMS$ .

Eventualmente, o modelo RVFL original com estimação de pesos pela regra MQO (sem BIP) produz soluções como na Figura 10a, que são até aceitáveis, pois apresentam patologia, digamos, leve. Mesmo assim, na maioria das execuções (18 de 20), esta variante gera

soluções patológicas não aceitáveis.

## 5.6 Sumário dos resultados dos modelos – conjunto WT1

A Tabela 7 exibe um resumo dos melhores resultados em termos de acurácia média, medida pelos índices RMSE e  $R^2$ , para o conjunto de dados WT1. Nesta tabela, dá-se destaque na última coluna à ocorrência ou não de solução patológica produzida pelo modelo.

Tabela 7 – Sumário dos melhores modelos e suas características com médias dos índices numéricos – conjunto WT1

Topologia	$q$	RMSE	$R^2$	TC (s)	Solução patológica?
Polinomial	7	13,0295	0,8527	0,0501	Sim
MLP-linear	5	13,3731	0,8447	60,7115	Sim
<b>MLP-logsig</b>	<b>5</b>	<b>13,1207</b>	<b>0,8506</b>	<b>71,0139</b>	<b>Não</b>
ELM	10	13,0225	0,8529	0,0364	Sim
<b>NoProp-logsig</b>	<b>10</b>	<b>13,1586</b>	<b>0,8498</b>	<b>21,1167</b>	<b>Não</b>
<b>RVFL-logsig</b>	<b>10</b>	<b>13,1298</b>	<b>0,8504</b>	<b>21,2288</b>	<b>Não</b>
RVFL	10	13,0220	0,8529	0,0521	Sim
<b>RVFL-logsig-BIP</b>	<b>10</b>	<b>13,2520</b>	<b>0,8476</b>	<b>22,6567</b>	<b>Não</b>
RVFL-BIP	10	13,9082	0,8317	0,0863	Sim

Fonte: Elaborada pelo autor.

A Tabela 8 a seguir mostra os valores mínimo, máximo e desvio padrão dos índices numéricos RMSE e  $R^2$ .

Tabela 8 – Sumário dos melhores modelos e outras estatísticas tais como valores de mínimo, máximo e desvio padrão dos índices numéricos – conjunto WT1

Topologia	$q$	RMSE			$R^2$		
		mínimo	máximo	desv. padrão	mínimo	máximo	desv. padrão
Polinomial	7	13,0295	13,0295	0,0000	0,8527	0,8527	0,0000
MLP-linear	5	13,0496	14,6402	0,4263	0,8140	0,8523	0,0102
<b>MLP-logsig</b>	<b>5</b>	<b>13,1038</b>	<b>13,1632</b>	<b>0,0157</b>	<b>0,8497</b>	<b>0,8510</b>	<b>0,0004</b>
ELM	10	13,0219	13,0243	0,0007	0,8528	0,8529	0,0000
<b>NoProp-logsig</b>	<b>10</b>	<b>13,1153</b>	<b>13,3194</b>	<b>0,0537</b>	<b>0,8461</b>	<b>0,8508</b>	<b>0,0012</b>
<b>RVFL-logsig</b>	<b>10</b>	<b>13,1098</b>	<b>13,2495</b>	<b>0,0355</b>	<b>0,8477</b>	<b>0,8509</b>	<b>0,0008</b>
RVFL	10	13,0216	13,0223	0,0002	0,8529	0,8529	0,0000
<b>RVFL-logsig-BIP</b>	<b>10</b>	<b>13,0884</b>	<b>13,7421</b>	<b>0,1722</b>	<b>0,8362</b>	<b>0,8515</b>	<b>0,0040</b>
RVFL-BIP	10	13,0295	14,8085	0,7847	0,8098	0,8527	0,0190

Fonte: Elaborada pelo autor.

Pela Tabela 7, os índices numéricos apresentam estatísticas equivalentes em relação

aos índices de desempenho RMSE e  $R^2$ , contudo, os destacados em negrito não produziram soluções patológicas. Na Tabela 8, em negrito estão destacados os modelos que não produziram solução patológica. Pelos índices numéricos mostrados, os modelos são estatisticamente equivalentes. Em relação ao RMSE, os únicos modelos que apresentaram valor da diferença máximo menos o mínimo maior que 1 são somente os modelos MLP-linear e RVFL-BIP, mas produziram soluções patológicas.

Para a Tabela 9 a seguir o conjunto WT1, que possui no total 47.542 amostras, foi dividido aleatoriamente em um conjunto de treino (CT) com 38.034 amostras e um outro para teste (CTT) com 9.508 amostras, respectivamente 80% e 20% dos dados amostrais. Foram mantidas as 20 rodadas independentes de treinamento e teste com duas das melhores topologias e comparada com o modelo de regressão polinomial de acordo com a Tabela 8 anterior.

Tabela 9 – Sumário com conjunto de treino (CT) e teste (CTT) e outras estatísticas tais como valores da média, mínimo, máximo e desvio padrão dos índices numéricos – conjunto WT1

Topologia	Conj.	RMSE				$R^2$			
		média	mín.	máx.	d. p.	média	mín.	máx.	d. p.
Polinomial	CT	13,048	12,968	13,111	0,033	0,852	0,851	0,854	0,001
	CTT	<b>12,996</b>	<b>12,743</b>	<b>13,315</b>	<b>0,131</b>	<b>0,853</b>	<b>0,846</b>	<b>0,858</b>	<b>0,003</b>
NoProp- logsig	CT	13,167	13,067	13,316	0,056	0,850	0,846	0,851	0,001
	CTT	<b>13,465</b>	<b>12,857</b>	<b>15,135</b>	<b>0,487</b>	<b>0,842</b>	<b>0,799</b>	<b>0,856</b>	<b>0,012</b>
RVFL- logsig	CT	13,142	13,072	13,265	0,050	0,850	0,847	0,852	0,001
	CTT	<b>13,429</b>	<b>13,066</b>	<b>13,817</b>	<b>0,250</b>	<b>0,843</b>	<b>0,835</b>	<b>0,851</b>	<b>0,005</b>

Fonte: Elaborada pelo autor.

Na Tabela 9 destacado em negrito estão os resultados do conjunto de teste (CTT). A hipótese de que somente os índices numéricos não são suficientes para obtenção de uma solução factível do problema se mantém, pois apesar dos bons índices permanecerem a solução patológica pode ocorrer para qualquer conjunto de dados. O modelo RVFL-logsig parece ter maior capacidade de generalização que o modelo NoProp-logsig para o conjunto WT1 pois seus resultados no conjunto de teste se mantiveram tão bons quanto os resultados obtidos durante o treinamento.

## 5.7 Sumário dos resultados dos modelos – conjunto Delfos

As Tabela 10 e Tabela 11 a seguir mostram um sumário com os melhores resultados para o conjunto de dados Delfos.

Tabela 10 – Sumário do melhores modelos e suas características – conjunto Delfos

Topologia	$q$	RMSE	$R^2$	TC (s)	Solução patológica?
MLP-linear	5	105,8997	0,9778	16,3149	Sim
<b>MLP-logsig</b>	<b>5</b>	<b>114,8223</b>	<b>0,9744</b>	<b>18,2955</b>	<b>Não</b>
ELM	10	91,2851	0,9838	0,0056	Sim
NoProp	10	146,7664	0,9573	3,1265	Sim
<b>NoProp-logsig</b>	<b>10</b>	<b>128,5476</b>	<b>0,9675</b>	<b>5,1636</b>	<b>Não</b>
<b>RVFL-logsig</b>	<b>10</b>	<b>106,4900</b>	<b>0,9779</b>	<b>4,8167</b>	<b>Não</b>
RVFL	10	91,0586	0,9839	0,0116	Sim

Fonte: Elaborada pelo autor.

Tabela 11 – Sumário dos melhores modelos e outras estatísticas tais como valores de mínimo, máximo e desvio padrão dos índices numéricos – conjunto Delfos

Topologia	$q$	RMSE			$R^2$		
		mínimo	máximo	desv. padrão	mínimo	máximo	desv. padrão
MLP-linear	5	96,0712	152,8282	15,4103	0,9546	0,9821	0,0073
<b>MLP-logsig</b>	<b>5</b>	<b>111,3806</b>	<b>118,4237</b>	<b>1,5868</b>	<b>0,9727</b>	<b>0,9759</b>	<b>0,0007</b>
ELM	10	91,0420	92,0312	0,2218	0,9835	0,9839	0,0001
NoProp	10	125,5970	209,7990	20,8937	0,9144	0,9693	0,0133
<b>NoProp-logsig</b>	<b>10</b>	<b>113,1611</b>	<b>160,5039</b>	<b>13,7317</b>	<b>0,9499</b>	<b>0,9751</b>	<b>0,0073</b>
<b>RVFL-logsig</b>	<b>10</b>	<b>105,8824</b>	<b>108,3505</b>	<b>0,5547</b>	<b>0,9772</b>	<b>0,9782</b>	<b>0,0002</b>
RVFL	10	91,0077	91,1934	0,0571	0,9838	0,9839	0,0000

Fonte: Elaborada pelo autor.

Para a Tabela 12 o conjunto Delfos, que possui no total 11.552 amostras, foi dividido aleatoriamente em um conjunto de treino (CT) com 9.242 amostras e um outro para teste (CTT) com 2.310 amostras, respectivamente 80% e 20% dos dados amostrais. Em negrito estão os resultados do conjunto de teste (CTT). Foram mantidas as 20 rodadas independentes de treinamento e teste, e três topologias de rede neurais randomizadas de acordo com a Tabela 11 anterior foram utilizadas para comparação.

Tabela 12 – Sumário com conjunto de treino (CT) e teste (CTT) e outras estatísticas tais como valores da média, mínimo, máximo e desvio padrão dos índices numéricos – conjunto Delfos

Topologia	Conj.	RMSE				$R^2$			
		média	mín.	máx.	d. p.	média	mín.	máx.	d. p.
ELM	CT	90,822	89,391	91,629	0,587	0,984	0,984	0,984	0,000
	<b>CTT</b>	<b>92,615</b>	<b>88,615</b>	<b>97,310</b>	<b>2,212</b>	<b>0,983</b>	<b>0,982</b>	<b>0,985</b>	<b>0,001</b>
NoProp-logsig	CT	141,26	114,90	209,30	22,14	0,960	0,914	0,974	0,014
	<b>CTT</b>	<b>152,81</b>	<b>115,01</b>	<b>244,88</b>	<b>38,40</b>	<b>0,952</b>	<b>0,881</b>	<b>0,975</b>	<b>0,027</b>
RVFL-logsig	CT	106,71	105,34	107,97	0,698	0,978	0,977	0,978	0,000
	<b>CTT</b>	<b>174,27</b>	<b>106,88</b>	<b>293,29</b>	<b>73,32</b>	<b>0,931</b>	<b>0,831</b>	<b>0,978</b>	<b>0,057</b>

Fonte: Elaborada pelo autor.

Sem perda de generalidade, para o conjunto de dados Delfos são relatados apenas os resultados referentes aos modelos baseados em rede neurais. Como mencionado anteriormente, a ocorrência de solução patológica pode acontecer para qualquer conjunto de dados se o modelo aplicado à tarefa de modelagem não for devidamente adequado e avaliado. Por devidamente adequado e avaliado, entende-se, uma avaliação guiada por índices de desempenho quantitativo e inspeção qualitativa da acurácia e da factibilidade da solução.

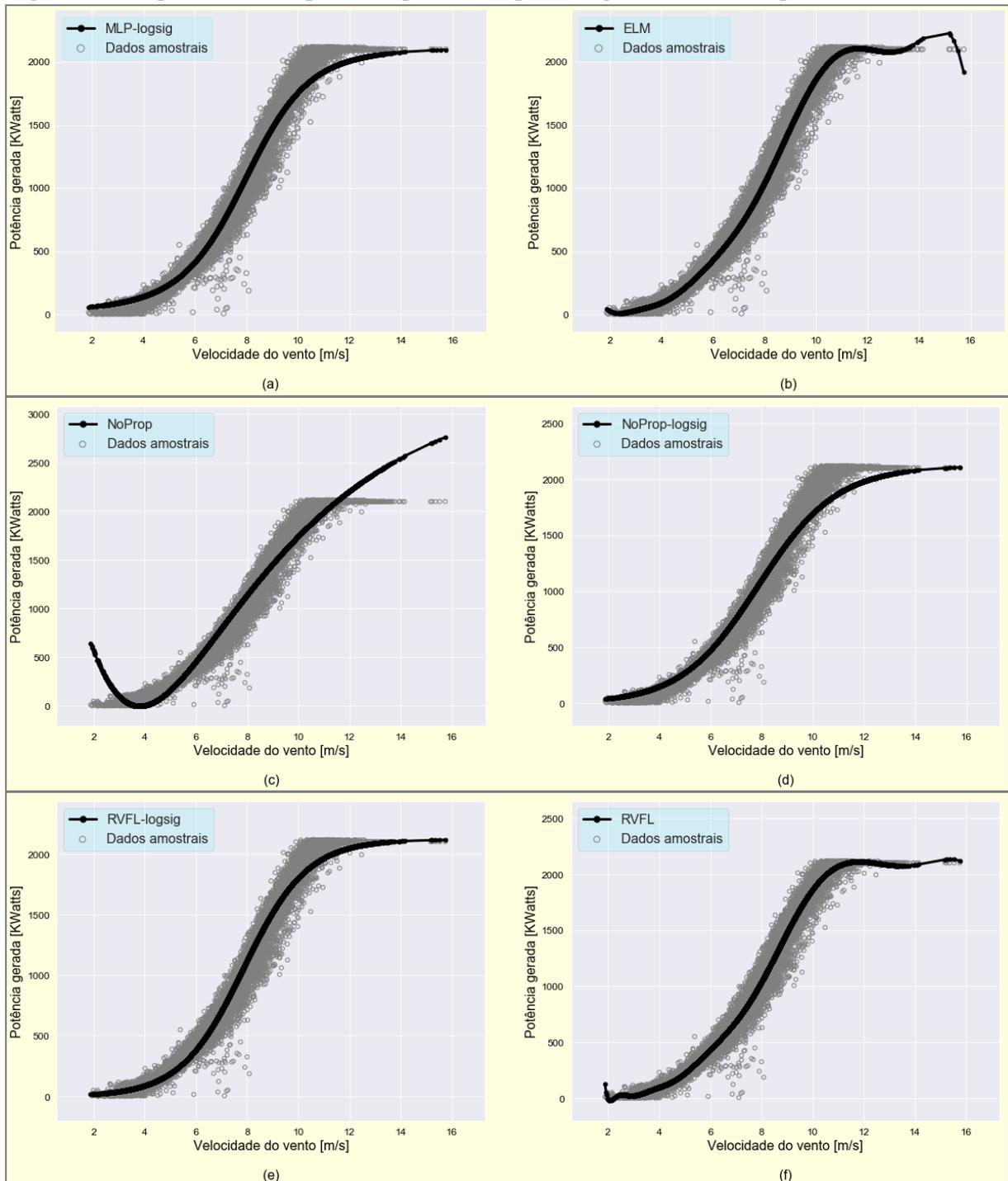
Pela melhoria dos resultados, o modelo MLP-linear foi treinado com taxa de aprendizagem igual a 0,2, e nos demais modelos foram mantidos os mesmos hiperparâmetros que os modelos na Tabela 7 e Tabela 8 do conjunto de dados WT1 para realizar os experimentos com os dados Delfos.

Na Tabela 12 destacado em negrito estão os resultados do conjunto de teste (CTT). O modelo NoProp-logsig parece ter maior capacidade de generalização que o modelo RVFL-logsig para o conjunto Delfos pois seus resultados no conjunto de teste se mantiveram tão bons quanto os resultados obtidos durante o treinamento.

Na Figura 11 a seguir, são mostradas seis curvas de potência estimadas pelos seguintes modelos de rede neural: MLP-logsig, ELM, Noprop, NoProp-logsig, RVFL-logsig e RVFL. Como esperado, existem soluções patológicas produzidas pelos modelos lineares ELM, NoProp e RVFL e são mostradas nas Figuras 11b, 11c e 11f, respectivamente, e elas parecem ser bastante incorretas. Por esta razão, estas soluções são entendidas como soluções patológicas. As soluções alcançadas pelos modelos MLP-logsig, NoProp-logsig e RVFL-logsig, apontam que os modelos apresentam resultados muito bons, ao menos por meio de inspeção visual.

Baseado apenas pelos resultados numéricos apresentados pelas Tabela 10 e Tabela 11, se forem consideradas apenas as redes que não produzem soluções patológicas, a melhor solução é fornecida pelo modelo RVFL-logsig, seguido pelos modelos MLP-logsig e NoProp-logsig nesta ordem. Os resultados numéricos podem ser melhorados, porém, é uma questão a ser abordada na continuação deste estudo. Segue a Figura 11 com as típicas soluções para os dados Delfos.

Figura 11 – Típica curva de potência produzida pelos seguintes modelos para os dados Delfos



Fonte: Elaborada pelo autor.

(a) Modelo MLP-logsig; (b) Modelo ELM; (c) Modelo NoProp; (d) Modelo NoProp-logsig; (e) Modelo RVFL-logsig; (f) Modelo RVFL; Os hiperparâmetros para este treinamento foram os mesmos utilizados para conseguir os resultados mostrados nas Tabela 7 e Tabela 8 para o conjunto de dados WT1.

Ressalta-se ainda que o modelo RVFL por vezes estima erroneamente (visto por inspeção gráfica) uma reta como CPA.

## 6 CONCLUSÕES E TRABALHOS FUTUROS

Neste estudo, foi mostrado que frequentemente os modelos de redes neurais randomizadas produzem resultados patológicos quando aplicados a tarefa de modelagem CPA. Tais ocorrências foram anteriormente observadas para modelos completamente treináveis, como a rede MLP, e para apenas um modelo randomizado, a saber, a rede ELM. No entanto, variantes de ambos os paradigmas de redes neurais são capazes de produzir soluções muito confiáveis nas quais a ocorrência de soluções patológicas são evitadas, a saber, os modelos MLP-logsig, NoProp-logsig, e RVFL-logsig (com ou sem BIP). Para tanto, foram usados dois conjuntos de dados desafiadores para este estudo.

Todavia, ainda entende-se que são necessárias mais investigações sobre o tema. Até agora, foram utilizadas três modelos de redes randomizadas, são elas, RVFL, ELM e NoProp. Os índices de validação numérica se mostraram úteis, mas ainda assim é fundamental uma inspeção gráfica para escolher os melhores resultados. Há grande vantagem na utilização das redes randomizadas tendo em vista que seu custo computacional é reduzido pois há atualização dos pesos somente dos neurônios de saída. A estratégia em utilizar o BIP precisa ser melhorada para produzir resultados aceitáveis.

Os próximos passos deste estudo visam avaliar mais modelos de redes randomizadas, e também diferentes estratégias de treinamento que permitam a essa classe de modelos de redes neurais produzirem melhores soluções para a tarefa de interesse. Uma lista não exclusiva de tarefas a fazer é dada a seguir. 1) Testar diferentes estratégias de treinamento, como descrito em Freire *et al.* (2020), visando melhorar o desempenho das redes randomizadas deste estudo. 2) Avaliar a robustez das redes neurais randomizadas implementadas em conjunto de dados contendo *outliers* (BARRETO e BARROS, 2015). Avaliar capacidade de generalização das redes randomizadas implementadas para um grande número de neurônios ocultos.

## REFERÊNCIAS

- ANACONDA. **Anaconda INC.**: Develop and deploy secure python solutions. 2018. Disponível em: <https://www.anaconda.com>. Acesso em 16 de jan. de 2018.
- APATA, O. & OYEDOKUN, D.T.O. An overview of control techniques for Wind turbine systems. **Science African**, v. 10, 2020.
- BAI, L.; CRISOSTOMI E.; RAUGI M; & TUCCI, M. Wind turbine power curve estimation based on earth mover distance and artificial neural networks. **IET Renewable Power Generation**, v. 13, i. 15, p. 2939–2946, 2019.
- BARRETO, G. A. & BARROS, A. L. B. On the design of robust linear pattern classifiers based on M-estimators. **Neural Processing Letters**, v. 42, p. 119-137, 2015.
- BARTLETT, P. L. The sample complexity of pattern classification with neural networks: the size of the weights is more importante than size of the network. **IEEE Trans. Inf. Theory**, v. 44, i. 2, p. 525-536, 1998.
- CARRILO, C.; OBANDO MONTAÑO, A. F.; CIDRÁS, J. & DÍAZ-DORADO, E. Review of power curve modelling for wind turbines. **Renewable and Sustainable Energy Reviews**, v. 21, p. 572–581, 2013.
- CHOI, Y.-S.; SHIN, H.-C. & SON, W.-J. Robust Regularization for Normalized LMS Algorithms. **IEEE Transactions on Circuits and Systems-II**, v. 53, i. 8, p. 627–631, 2006.
- CLIFTON, A.; KILCHER, L.; LUNDQUIST, J. K. & FLEMING, P. Using machine learning to predict wind turbine power output. **Environmental Research Letters**, v. 8, 2013.
- COSTARELLI, D. Neural network operators: Constructive interpolation of multivariate functions. **Neural Networks**, v. 67, p. 28–36, 2015.
- DAS, A. K. An empirical model of power curve of a wind turbine. **Energy Systems**, v. 5, i. 3, p. 507-518, 2014.
- EMINOGLU, U. & TURKSOY, O. (2019). Power curve modeling for wind turbine systems: a comparison study. **International Journal of Ambient Energy**, v. 42, p. 1912-1921, 2021.
- FREIRE, A. L. **On The Efficient Design of Extreme Learning Machines Using Intrinsic Plasticity and Evolutionary Computation Approaches**. 2015. Tese (Doutorado em Engenharia de Teleinformática) – Centro de Tecnologia, Programa de Pós-Graduação em Engenharia de Teleinformática, Universidade Federal do Ceará, Fortaleza, 2015.
- FREIRE, A. L.; ROCHA-NETO, A. R. & BARRETO, G. A. On robust randomized neural networks for regression: a comprehensive review and evaluation. **Neural Computing and Applications**. v. 32, p. 16931-16950, 2020.
- GONZÁLEZ-CARRATO, R. R. H. Wind farm monitoring using Mahalanobis distance and fuzzy clustering. **Renewable Energy**, v. 123, p. 526–540, 2018.

GOLUB, G. H. & VAN LOAN, C. F. **Matrix computations**. 4th ed. Baltimore, Maryland: Johns Hopkins University Press, 2012.

GUO, P. & INFELD, D. Wind Turbine Power Curve Modeling and Monitoring With Gaussian Process and SPRT. **IEEE Transactions on Sustainable Energy**, v. 11, i. 1, p. 107–115, 2020.

HAGAN, M. T., & MENHAJ, M. B. Training feedforward networks with the Marquardt algorithm. **IEEE Transactions on Neural Networks**, v. 5, i. 6, p. 989-993, 1994.

HAYKIN, S. **Neural Networks and Learning Machines**. 2. ed. [S. l.]: Prentice Hall, 1999. ISBN 8178083000.

HORNICK, K., Approximation Capabilities of Multilayer Feedforward Networks. **Neural Networks**, vol. 4, p. 251-257, 1991.

HORNICK, K.; STINCHCOMBE, M. and WHITE, H. Multilayer Feedforward Networks are Universal Approximators. **Neural Networks**, vol. 2, i. 5, p. 359-366, 1989.

HU, Y.; XI, Y.; PAN, C.; LI, G. & CHEN, B. Daily condition monitoring of grid-connected wind turbine via high-fidelity power curve and its comprehensive rating. **Renewable Energy**, v. 146, p. 2095–2111, 2020.

HUANG, G.-B. & BABRI, H.A. Upper bounds on the number of hidden neurons in feedforward networks with arbitrary bounded nonlinear activation functions. **IEEE Trans Neural Networks**, v. 9, i. 1, p. 224-229, 1998.

HUANG, G.-B.; ZHU, Q.-Y. & SIEW, C.-K. Extreme learning machine: theory and applications. **Neuro-computing**, v. 70, p. 489–501, 2006.

JOHN. D. Hunter, Matplotlib: A 2D Graphics Environment, **Computing in Science & Engineering**, v. 9, n. 3, 90-95, 2007.

JAFARIAN, M. & RANJBAR, A. M. Fuzzy modeling techniques and artificial neural networks to estimate anual energy output of a wind turbine. **Renewable Energy**, v. 35, p. 2008–2014, 2010.

JONES, E.; OLIPHANT, T.; PETERSON, P. et al. **SciPy 1.0: Fundamental Algorithms for scientific computing in Python**. **Nature Methods**, v. 17, p. 261-272, 2020.

JUPYTER. **Jupyter**. United States, 2018. Disponível em: <https://www.jupyter.org>. Acesso em 16 de jan. de 2018.

LEE, G.; DING, Y.; GENTON, M. G. & XIE, L. Power curve estimation with multivariate environmental factors for inland and offshore wind farms. **Journal of the American Statistical Association**, v. 110, n. 509, p. 56-67, 2015.

LESHNO, M.; LIN, V.Y.; PINKUS, A. & SCHOCKEN, S. Multilayer FeedForward Networks with Non-Polynomial Activation Function Can Approximate Any Function. **Neural networks**, vol. 6, p. 861-867, 1993.

LI, S.; WUNSCH, D. C.; OHAIR, E. & GIESSELMANN, M. G. Comparative analysis of regression and artificial neural network models for wind turbine power curve estimation. **Journal of Solar Energy Engineering**, v. 123, i. 4, p. 327-332, 2001.

LYDIA, M.; KUMAR, S.; S., SELVAKUMAR, A. I., & KUMAR, G. E. P. (2014). A comprehensive review on wind turbine power curve modeling techniques. **Renewable and Sustainable Energy Reviews**, v. 30, p. 452-460, 2014.

LYDIA, M.; SELVAKUMAR, A. I.; KUMAR, S. S. & KUMAR, G. E. P. Advanced algorithms for wind turbine power curve modeling. **IEEE Transactions on Sustainable Energy**, v. 4, i. 3, p. 827-835, 2013.

MANOBEL, B.; SEHNKE, F.; LAZZÚS, J. A.; SALFATE, I.; FELDER, M. & MONTECINOS, S. Wind turbine power curve modeling based on Gaussian Processes and Artificial Neural Networks. **Renewable Energy**, v. 125, p. 1015–1020, 2018.

MAYA, H. C., **Revisitando o Problema de Estimação de Curvas de Potência de Turbinas Eólicas via Modelos Polinomiais Esparsos**. 2018. Dissertação (Mestrado em Engenharia de Teleinformática) – Centro de Tecnologia, Programa de Pós-Graduação em Engenharia de Teleinformática, Universidade Federal do Ceará, Fortaleza, 2018.

MEHRJOO, M.; JOZANI, M. J. & PAWLAK, M. Wind turbine power curve modeling for reliable power prediction using monotonic regression. **Renewable Energy**, v. 147, p. 214–222, 2020.

MOODY, J. & DARKEN, C. J. Fast learning in networks of locally tuned processing units. **Neural Computation**, v. 1, p. 281–294, 1989.

OLIPHANT, T. **NumPy: A guide to NumPy**. 2006. USA: Trelgol Publishing. Disponível em: <http://www.numpy.org/>. Acesso em 16 de janeiro de 2018.

OUYANG, T., KUSIAK, A. & HE, Y. Modeling wind-turbine power curve: A data partitioning and mining approach. **Renewable Energy**, v. 102-Part A, p. 1–8, 2017.

PANDIT, R. K., & INFIELD, D. Comparative analysis of Gaussian Process power curve models based on different stationary covariance functions for the purpose of improving model accuracy. **Renewable Energy**, v. 140, p. 190–202, 2019.

PANDIT, R. K., INFIELD, D., & KOLIOS, A. Comparison of advanced non-parametric models for wind turbine power curves. **IET Renewable Power Generation**, v. 13, i. 9, p. 1503–1510, 2019.

PAO, Y.-H.; PARK, G.-H. & SOBAJIC, D. J. Learning and generalization characteristics of the random vector Functional-link net. **Neurocomputing**, v. 6, p. 163–180, 1994.

PEI, S. & LI, Y. (2018). Wind Turbine Power Curve Modeling with a Hybrid Machine Learning Technique. **Applied Sciences**, v. 9, n. 22, 2019.

POGGIO, T. & GIROSI, F. Networks for approximation and learning. **Proceedings of the IEEE**, v. 78, i. 9, p. 1484–1487, 1990.

PYTHON, **Python Foundation Software**. United States of America. 2018. Disponível em: <https://www.python.org>. Acesso em 16 de janeiro de 2018.

RUMELHART, D. E.; HINTON, G. E. & WILLIAMS, R. J. Learning representations by back-propagating erros. **Nature**, v. 323, p. 533–536, 1986.

SCHLECHTINGEN, M.; SANTOS, I. F. & ACHICHE, S. Using data-mining approaches for wind turbine power curve monitoring: a comparative study. **IEEE Transactions on Sustainable Energy**, v. 4, i. 3, p. 671-679, 2013.

SCHLECHTINGEN, M. & SANTOS, I. F. Comparative analysis of neural network and regression based condition monitoring approaches for wind turbine fault detection. **Mechanical Systems and Signal Processing**, v. 25, i. 5, p. 1849-1875, 2011.

SHOKRZADEH, S.; JOZANI, M. J., & BIBEAU, E. Wind turbine power curve modeling using advanced parametric and nonparametric methods. **IEEE Transactions on Sustainable Energy**, v. 5, i. 4, p. 1262-1269, 2014.

SOHONI, V.; GUPTA, S. C. & NEMA, R. K. A critical review on wind turbine power curve modelling techniques and their applications in wind based energy systems. **Journal of Energy**, v. 2016, 2016.

SUGENO, M. & KANG, G. Structure identification of fuzzy model. **Fuzzy Sets and Systems**, v. 28, i. 1, p. 15–33, 1988.

TAKAGI, T. & SUGENO, M. Fuzzy identification of systems and its applications to modeling and control. **IEEE Transactions on Systems, Man, and Cybernetics**, v. 15, i. 1, p. 116–132, 1985.

ÜSTÜNTAS, T. & SAHIN, A. D. Wind turbine power curve estimation based on cluster center fuzzy logic modeling. **Journal of Wind Engineering and Industrial Aerodynamics**, v. 96, i. 5, p. 611–620, 2008.

VILLANUEVA, D. & FEIJÓO, A. Comparison of logistic functions for modeling wind turbine power curves. **Electric Power Systems Research**, v. 155, p. 281–288, 2018.

VIRGOLINO, G. C. M. **Wind Turbine Power Curve Modeling with Gaussian Processes**. 2020. Dissertação (Mestrado em Modelagem e Métodos Quantitativos) – Centro de Ciências, Programa de pós-Graduação em Modelagem e Métodos Quantitativos, Universidade Federal do Ceará, Fortaleza, 2020.

VIRGOLINO, G. C. M.; MATTOS, C. L. C.; MAGALHÃES, J. A. F. & BARRETO, G. A. Gaussian processes with logistic mean function for modeling wind turbine power curves. **Renewable Energy**, v. 162, p. 458–465, 2020.

WASKOM, M. L. Seaborn: statistical data visualization. **Journal of Open Source Software**, v. 6, i. 60, id. 3021, 2021.

WANG, Y.; HU, Q.; LI, L.; FOLEY, A. M. & SRINIVASAN. Approaches to wind power curve modeling: A review and discussion. **Renewable and Sustainable Energy Reviews**, v. 116,

2019.

WIDROW, B. Thinking About Thinking: The Discovery of the LMS Algorithm. **IEEE Signal Processing Magazine**, v. 22, i. 1, p. 100–106, 2005.

WIDROW, B.; GREENBLATT, A.; KIM, Y. & PARK, D. The No-Prop Algorithm: A New Learning Algorithm for Multilayer Neural Networks. **Neural Networks**, v. 37, p. 182–188, 2013.

YAN, J.; ZHANG, H.; LIU, Y.; HAN, S. & LI, L. Uncertainty estimation for wind energy conversion by probabilistic wind turbine power curve modelling. **Applied Energy**, v. 239, p. 1356–1370, 2019.

YESILBUDAK, M. Implementation of novel hybrid approaches for power curve modeling of wind turbines. **Energy Conversion and Management**, v. 171, p. 156–169, 2018.

YOU, M.; LIU, B.; BYON, E.; HUANG, S. & JIN, J. Direction-dependent power curve modeling for multiple interacting wind turbines. **IEEE Transactions on Power Systems**, v. 33, i. 2, p. 1725–1733, 2018.

ZHANG, L. & SUGANTHAN, P. N. A comprehensive evaluation of random vector functional link networks. **Information Sciences**, v. 367–368, p. 1094–1105, 2016.