

Jerfson Bruno do Nascimento Honório

**Um Estudo Sobre Modelos Autorregressivos de  
Redes Neurais  
 $AR-NN(p)$**

Campina Grande-PB

21 de novembro de 2019



Jerfson Bruno do Nascimento Honório

**Um Estudo Sobre Modelos Autorregressivos de Redes  
Neurais  
 $AR-NN(p)$**

Trabalho de Conclusão de Curso apresentado  
ao Curso de Graduação em Estatística da Uni-  
versidade Federal de Campina Grande como  
sendo requisito parcial para obtenção do tí-  
tulo de Bacharel em Estatística.

Universidade Federal de Campina Grande - UFCG

Centro de Ciências e Tecnologia-CCT

Unidade Acadêmica de Estatística - UAEST

Orientador: Profa. Dra. Amanda Santos Gomes

Coorientador: Prof. Dr. Francisco Antônio Morais de Souza

Campina Grande-PB

21 de novembro de 2019

Jerfson Bruno do Nascimento Honório

# **Um Estudo Sobre Modelos Autorregressivos de Redes Neurais AR-NN( $p$ )**

Trabalho de Conclusão de Curso apresentado  
ao Curso de Graduação em Estatística da Uni-  
versidade Federal de Campina Grande como  
sendo requisito parcial para obtenção do tí-  
tulo de Bacharel em Estatística.

Trabalho aprovado. Campina Grande-PB, 21 de novembro de 2019:

---

**Profª. Dra. Amanda Santos Gomes**

/UAEst/CCT/UFCG

Orientadora

---

**Prof. Dr. Francisco Antônio Moraes de  
Souza**

/UAEst/CCT/UFCG

Orientador

---

**Prof. Dr. Damásio Fernandes Júnior**

/UAEE/CEEI/UFCG

Avaliador

Campina Grande-PB  
21 de novembro de 2019

*Este trabalho é dedicado às crianças adultas que,  
quando pequenas, sonharam em se tornar cientistas.*



# Agradecimentos

Gostaria de agradecer a todos os professores da Unidade Acadêmica de Estatística, em especial, a minha orientadora Profa. Amanda dos Santos Gomes (Amanda) e ao meu coorientador Prof. Francisco Antônio Morais de Souza (Chico) por terem aceitado o desafio de me orientar nesse trabalho. Obrigado, vocês são incríveis!

Também não poderia esquecer do meu Professor de Matemática do Ensino Médio Hércules do Nascimento Silva (Parceiro), pessoa que têm minha admiração e respeito, afinal, foi ele que me apresentou essa área da Matemática.

Ao meu tutor do grupo PET, Prof. Dr. José Lindomberg Possiano Barreiro (Lindomberg), pela sua infinita paciência e ajuda prestada sempre que solicitado.

Aos estudantes do curso que direta ou indiretamente estiveram envolvidos na minha caminhada até aqui. Dentre eles posso citar: Alan, Bianca, Cássia, Elias, Gabriel, Kleber, Paulo e Thallyta.

Agradecer também a todos os amigos que partilham comigo os melhores momentos da minha vida. É difícil citar todos vocês, mas é mais do que justo explicitar que Bruno Santos (Malto), David Willian (Xadrez), Dimas Lucena (Boi), Edson Lourenço (Edy), Emanuel França (Manuka), Igor Araújo (Galego), João Viera (Papi), Lucas Lima (Mzr), Lucas Silva (Doido), Júlio César (Julhin) e Tércio Correia (Zaú), são pessoas que têm minha admiração e respeito.

Um agradecimento especial as três pessoas que são essenciais na minha vida, minha mãe Josimere Marçal do Nascimento Honório, minha irmã Rhuana Gabriela do Nascimento Honório e meu padastro Valmir Borba Gomes de Moura. Obrigado por sempre cuidarem de mim, por toda a minha vida serei grato.

A minha namorada Maria Eduarda Albuquerque pelo companheirismo e paciência durante vários anos.

Por fim, a todas as pessoas que fazem parte do departamento de estatística, os colegas de curso, os professores, os servidores. Durante os anos de convivência com vocês, deixo meu sincero agradecimento e gratidão por contribuir na minha graduação.





*“Um general nunca demonstra desespero.  
Ele inspira confiança em suas tropas.  
Ele os leva adiante, mesmo que seja para a morte.”*  
*As Crônicas dos Kane - O Trono de Fogo*



# Resumo

É de grande interesse o estudo de previsão de séries temporais, ou seja, conseguir antecipar características do processo num momento futuro. Para isso, é necessário estimar com precisão, ou pelo menos com uma boa aproximação, o processo gerador dos dados. Nos últimos anos, modelos de redes neurais artificiais vêm desempenhando um papel crescente na abordagem e solução de problemas estatísticos importantes. As redes neurais têm se mostrado uma alternativa vantajosa, em alguns casos, em relação aos modelos lineares tradicionais. Os modelos autorregressivos de redes neurais são pouco explorados pela comunidade estatística. Por isso, esta monografia tem como objetivo compreender sua fundamentação teórica, principais arquiteturas e algoritmos de aprendizagem. Ainda, é realizado um estudo de desempenho preditivo desses modelos, tanto no caso de séries temporais com tendência e sazonalidade como para séries de alta volatilidade. São utilizadas séries sobre vendas mensais de medicamentos orais para diabéticos na Austrália, que contém tendência e sazonalidade, e a série sobre retornos das ações preferenciais da Petrobras (PETR4), que contém alta volatilidade.

Os resultados mostram de maneira geral que os modelos autorregressivos de redes neurais são vantajosos em relação a modelos lineares, e apresentam bons resultados em diferentes tipos de dados.

**Palavras-chave:** Redes neurais, Modelos autorregressivos, Modelos autorregressivos de redes neurais.



# Abstract

It is of great interest to study time series prediction, that is, to be able to anticipate process characteristics at a future time. For this, it is necessary to accurately estimate, or at least with a good approximation, the data generating process. In recent years, artificial neural network models have played an increasing role in addressing and solving important statistical problems. Neural networks have proved to be an advantageous alternative, in some cases, over traditional linear models. The autoregressive models of neural networks are little explored by the statistical community. Therefore, this monograph aims to understand its theoretical foundation, main architectures and learning algorithms. Furthermore, a predictive performance study of these models is performed, both for trend and seasonality time series and for high volatility series. The series on monthly sales of oral diabetic drugs in Australia, which contains trend and seasonality, and the series on Petrobras preferred stock returns (PETR4), which contains high volatility.

The results show in general that the autoregressive models of neural networks are advantageous over linear models, and present good results in different types of data.

**Keywords:** Neural networks, autoregressive models, autoregressive models of neural networks.



# Lista de ilustrações

Figura 1 – Redes Neurais Simples. . . . .	31
Figura 2 – Redes Neurais <i>feed-forward</i> . . . . .	32
Figura 3 – Redes Neurais Recorrentes: <b>Jordan</b> . . . . .	33
Figura 4 – Redes Neurais Recorrentes: <b>Elman</b> . . . . .	34
Figura 5 – Função de Ativação Logística. . . . .	36
Figura 6 – Função de Ativação Tangente Hiperbólica. . . . .	37
Figura 7 – Função de Ativação ReLU. . . . .	38
Figura 8 – Função de Ativação Leaky ReLU. . . . .	39
Figura 9 – Representação gráfica de um modelo AR(4). . . . .	41
Figura 10 – Representação gráfica do AR-NN(4). . . . .	41
Figura 11 – Fluxograma da construção de modelo AR-NN ( $p$ ). . . . .	50
Figura 12 – Estimação interativa dos parâmetros. . . . .	62
Figura 13 – Fluxograma do algoritmo de Levenberg-Marquardt. . . . .	67
Figura 14 – Série sobre vendas mensais de medicamentos orais para diabéticos na Austrália em milhões de dólares ao longo dos meses. . . . .	74
Figura 15 – (a) Função de autocorrelação da série em relação as defasagens. (b) Função de autocorrelação parcial da série em relação as defasagens. . .	75
Figura 16 – Gráfico sazonal polar de todos os anos das vendas mensais de medicamentos ao longo dos meses. . . . .	76
Figura 17 – Valores estimados com o Modelo AR-NN(5) com 1 camada e 6 neurônios ocultos ajustado sobre a série vendas mensais de medicamentos para diabéticos na Austrália. . . . .	78
Figura 18 – (a) Quantis da distribuição dos resíduos contra os quantis da distribuição normal (QQ-plot). (b) Função de autocorrelação dos resíduos em relação as defasagens. (c) Função de autocorrelação parcial dos resíduos em relação as defasagens. . . . .	79
Figura 19 – (a) Função de autocorrelação dos resíduos ao quadrado em relação as defasagens. (b) Função de autocorrelação parcial dos resíduos ao quadrado em relação as defasagens. . . . .	80
Figura 20 – Representação gráfica da Tabela (2). . . . .	81
Figura 21 – Retornos das ações preferenciais da Petrobras (PETR4) ao longo dos meses. . . . .	83
Figura 22 – (a) Função de autocorrelação da série em relação as defasagens. (b) Função de autocorrelação parcial da série em relação as defasagens. . .	84
Figura 23 – Breve explicação do gráfico <i>box plot</i> . . . . .	85
Figura 24 – <i>Box plot</i> do retorno para cada mês. . . . .	85

Figura 25 – Valores estimados com o Modelo AR-NN(17) com 1 camada e 10 neurônios ocultos ajustado sobre a série das ações preferenciais da Petrobras - PETR4. . . . .	87
Figura 26 – (a) Quantis da distribuição dos resíduos contra os quantis da distribuição normal (QQ-plot). (b) Função de autocorrelação dos resíduos em relação as defasagens. (c) Função de autocorrelação parcial dos resíduos em relação as defasagens. . . . .	88
Figura 27 – (a) Função de autocorrelação dos resíduos ao quadrado em relação as defasagens. (b) Função de autocorrelação parcial dos resíduos ao quadrado em relação as defasagens. . . . .	89
Figura 28 – Representação gráfica da Tabela (4). . . . .	91



# Lista de tabelas

Tabela 1 – Termos em Diferentes Áreas. . . . .	40
Tabela 2 – Previsões. . . . .	80
Tabela 3 – Medidas de erro. . . . .	81
Tabela 4 – Previsões dos retornos mensais das ações PETR4. . . . .	90
Tabela 5 – Medidas de erro. . . . .	91
Tabela 6 – Estimativas dos Parâmetros. . . . .	100
Tabela 7 – Estimativas dos Parâmetros. . . . .	101
Tabela 8 – Estimativas dos Parâmetros. . . . .	102



# Lista de abreviaturas e siglas

AR-NN( $p$ )	Processo autorregressivo de redes neurais de ordem $p$ ;
AR( $p$ )	Processo autorregressivo de ordem $p$ ;
ARMA	Processo autorregressivo de médias móveis;
ADF	Teste de Dickey-Fuller;
AIC	Critério de informação Akaike;
AC	Coeficiente de autocorrelação;
ACF	Função de autocorrelação;
BIC	Critério de informação Schwarz-Bayesiano;
IC	Critério de informação;
LM	Multiplicador de Lagrange;
MA( $q$ )	Médias móveis com atraso $q$ ;
MAE	Média dos erros em valor absoluto;
MPE	Percentual médio;
MAPE	Percentual médio absoluto;
ME	Erro médio;
MI	Informação mútua;
MIC	Critério de informação mútua;
NLS	Métodos de mínimos quadrados não lineares;
PACF	Função de autocorrelação parcial;
PAC	Coeficiente de autocorrelação parcial;
PETR4	Ações preferenciais da Petrobras;
RADF	Teste de Dickey-Fuller <i>rank</i> ;
RMSE	Raiz quadrada da média dos erros quadráticos.



# Lista de símbolos

$X_t$	Processo estocástico de ordem $t$ ;
$\mu(t)$	Média no tempo $t$ ;
$\text{Var}(t)$	Variância no tempo $t$ ;
$\gamma(t_1, t_2)$	Covariância entre $X_{t_1}$ e $X_{t_2}$ ;
$E(X_t)$	Esperança de $X_t$ ;
$\varepsilon_t$	Ruído branco;
$\mathbf{x}_{t-p}$	Vetor de defasagens de tamanho $p$ ;
$\nabla(\cdot)$	Vetor gradiente;
$\Delta$	Variação;
$J(\cdot)$	Matriz jacobiana;
$\nabla^2(\cdot)$	Matriz hessiana;
$i$	Iteração;
$\hat{X}_t$	Valores previstos;
$p_t$	Erro percentual;
$Q(\Theta)$	Função desempenho;
$R_{n,t}$	<i>Rank</i> de $X_t$ ;
$u_t$	Termo residual;
$G(\cdot; \cdot)$	Função de dois argumentos;
$\phi_i$	Parâmetros da regressão linear artificial;
$w_i$	Pesos da rede neural;
$h_i^j$	Neurônio oculto de índice $i$ e camada $j$ ;
$\Psi(\cdot)$	Função de ativação;
$X_{t-i}$	Processo autorregressivo com atraso $i$ ;

$\alpha_i$	Parâmetros autorregressivos da parte linear;
$\gamma_{ij}$	Parâmetros autorregressivos da parte não linear;
$\gamma_{0j}$	Viés;
$\beta_j$	Pesos da saída não linear;
$\mathbf{A}$	Vetor de parâmetros autorregressivos lineares;
$\mathbf{\Gamma}_{pj}$	Vetor de parâmetros autorregressivos não lineares;
$\boldsymbol{\varepsilon}_t$	Vetor de ruídos;
$I^m$	Hipercubo unitário de dimensão $m$ ;
$C(I^m)$	Espaço das funções contínuas;
$\Gamma ; \mathcal{W} ; \mathcal{B}$	Espaços de pesos;
$h$	Número de neurônios ocultos;
$B$	Operador de atraso;
$\alpha(B)$	Operador autorregressivo de ordem $p$ ;
$\tilde{X}_t$	Processo estocástico padronizado de ordem $t$ .

# Sumário

<b>1</b>	<b>INTRODUÇÃO</b>	<b>23</b>
<b>2</b>	<b>PROCESSOS DE REDES NEURAIS AUTORREGRESSIVAS - AR- NN(<math>p</math>)</b>	<b>25</b>
<b>2.1</b>	<b>Processos Estocásticos</b>	<b>25</b>
2.1.1	Processos Lineares Estacionários	26
2.1.2	Processos Autorregressivos	28
2.1.3	Processos não lineares Autorregressivos	30
<b>2.2</b>	<b>Redes Neurais</b>	<b>31</b>
2.2.1	Redes Perceptron Simples	31
2.2.2	Redes Perceptron Multicamadas ( <i>Feed-Forward</i> )	32
2.2.3	Redes Recorrentes <i>Feed-backward</i>	33
2.2.4	Modelo Matemático para as Redes Neurais	34
2.2.5	Função de Ativação	35
2.2.5.1	Função Logística	36
2.2.5.2	Função Tangente Hiperbólica	37
2.2.5.3	Função ReLU	37
2.2.5.4	Função <i>Leaky</i> ReLU	38
<b>2.3</b>	<b>Redes Neurais Autorregressivas AR-NN(<math>p</math>)</b>	<b>39</b>
2.3.1	Forma Gráfica AR-NN( $p$ )	40
2.3.2	Equação AR-NN( $p$ )	42
2.3.3	O Teorema da Aproximação Universal	43
<b>2.4</b>	<b>Estacionariedade do Modelo</b>	<b>45</b>
2.4.1	Estacionariedade dos modelos AR-NN	45
2.4.2	O Teste de Classificação Aumentada de Dickey-Fuller <i>Rank</i>	47
<b>3</b>	<b>MODELAGEM UNIVARIADA DOS MODELOS AUTORREGRES- SIVOS DE REDES NEURAIS AR-NN (<math>p</math>)</b>	<b>49</b>
<b>3.1</b>	<b>Teste de não Linearidade</b>	<b>51</b>
3.1.1	O teste de White	51
<b>3.2</b>	<b>Seleção de Variáveis</b>	<b>53</b>
3.2.1	O Coeficiente de Autocorrelação	54
3.2.2	Informações Mútuas	55
<b>3.3</b>	<b>Estimação dos parâmetros</b>	<b>56</b>
3.3.1	Função Desempenho	57
3.3.2	Termos Matriciais Importantes	59

3.3.3	Características Básicas dos Algoritmos . . . . .	60
3.3.4	Métodos de Descida de Gradiente de Primeira Ordem . . . . .	62
3.3.5	Métodos de Descida de Gradiente de Segunda Ordem . . . . .	64
3.3.6	O Algoritmo Levenberg-Marquardt . . . . .	64
<b>3.4</b>	<b>Testes de Parâmetros . . . . .</b>	<b>68</b>
3.4.1	Testes de Parâmetro <i>Bottom-Up</i> . . . . .	68
3.4.1.1	Expansão de Taylor . . . . .	68
3.4.1.2	<i>Bottom-Up</i> . . . . .	69
<b>3.5</b>	<b>Medidas de Erro . . . . .</b>	<b>70</b>
<b>4</b>	<b>APLICAÇÃO . . . . .</b>	<b>73</b>
<b>4.1</b>	<b>Vendas mensais de medicamentos para diabéticos na Austrália, de 1991 a 2008. . . . .</b>	<b>73</b>
4.1.1	Tendência . . . . .	75
4.1.2	Sazonalidade . . . . .	75
4.1.3	Raiz Unitária . . . . .	76
4.1.4	Teste de White . . . . .	77
4.1.5	Modelo . . . . .	77
4.1.5.1	Previsões . . . . .	80
<b>4.2</b>	<b>Ações preferenciais da Petrobras - PETR4 . . . . .</b>	<b>82</b>
4.2.1	Tendência . . . . .	84
4.2.2	Sazonalidade . . . . .	84
4.2.2.1	Raiz Unitária . . . . .	86
4.2.2.2	Teste White . . . . .	86
4.2.3	Modelo . . . . .	86
4.2.4	Previsões . . . . .	89
<b>5</b>	<b>CONCLUSÕES . . . . .</b>	<b>93</b>
<b>5.1</b>	<b>Recomendações para Trabalhos Futuros . . . . .</b>	<b>93</b>
	<b>REFERÊNCIAS . . . . .</b>	<b>95</b>
	<b>ANEXO A – ESTIMATIVAS DOS PARÂMETROS DOS MODE- LOS: . . . . .</b>	<b>100</b>
<b>A.1</b>	<b>AR-NN(5) com 1 camada e 6 neurônios ocultos. . . . .</b>	<b>100</b>
<b>A.2</b>	<b>AR-NN(17) com uma camada e dez neurônios ocultos. . . . .</b>	<b>101</b>



# 1 Introdução

É de grande interesse o estudo de métodos de previsão de séries temporais, ou seja, conseguir antecipar algumas características do processo num momento futuro. As condições presentes determinam, em algum grau, o futuro, possivelmente envolvendo relações complexas entre as variáveis. Uma situação ideal para a realização de predições seria o conhecimento das equações que modelam os mecanismos responsáveis pela geração das séries temporais. No entanto, em muitos problemas reais essas informações não são indisponíveis, e não se tem condições ideais para construir equações que governem a dinâmica das variáveis de interesse. Quando isso acontece, o usual é utilizar uma abordagem baseada em modelos, na qual tenta-se identificar ou aproximar o processo gerador dos dados.

Redes Neurais Artificiais vêm desempenhando um papel crescente nos últimos anos na abordagem e solução de problemas estatísticos importantes. Sua contribuição se dá principalmente em problemas de reconhecimento de padrões e problemas de predição, tanto para dados transversos (regressão) quanto para dados de séries temporais. A estrutura mais básica de rede neural, as chamadas redes multicamadas (ou redes *feed-forward*), é vista como uma alternativa não linear aos modelos estatísticos lineares tradicionais, como por exemplo, modelos de regressão linear ou modelos autorregressivos.

As redes neurais vêm se mostrando uma alternativa vantajosa, em alguns casos, em relação aos modelos lineares tradicionais. A característica marcante desse tipo de abordagem é sua capacidade de modelar tanto estruturas lineares quanto não lineares, podendo, inclusive, aproximar, com qualquer grau de acurácia, uma função arbitrária.

Com isso, o principal objetivo dessa monografia, é o estudo dos modelos autorregressivos de redes neurais, o que compreende sua fundamentação teórica, principais arquiteturas e algoritmos de aprendizagem. Além disso, veremos o seu desempenho em diferentes tipos de dados.



## 2 Processos de Redes Neurais Autorregressivas - AR-NN( $p$ )

Neste capítulo será apresentada a teoria básica dos processos de redes neurais autorregressivas, AR-NN( $p$ ). Começaremos com uma definição de processos estocásticos, processos lineares estacionários e processos autorregressivos. Em contraste com a maioria das literaturas de séries temporais, usamos uma definição geral para garantir que processos autorregressivos não lineares também sejam processos autorregressivos por suas propriedades básicas. Além disso, é apresentada uma introdução aos problemas nas estimativas lineares e aos objetivos dos modelos não lineares em superá-los. A maioria dos modelos não lineares é dedicada a certas não linearidades específicas dos dados, como quebras estruturais no coeficiente ou na constante de regressão. Veremos que um AR-NN( $p$ ) é capaz de aproximar qualquer função e portanto, qualquer não linearidade. Também veremos que eles são paramétricos, o que os torna fáceis de manusear. Essas duas características são as principais razões pelas quais as redes neurais são usadas para superar o problema da não linearidade.

### 2.1 Processos Estocásticos

Quando as variáveis aleatórias associadas a um evento dependem do tempo é dito que estas variáveis são estocásticas. De acordo com [Morettin \(2008\)](#), uma definição para processo estocástico é dado como

**Definição 2.1.** Seja  $T$  um conjunto arbitrário. Um processo estocástico é uma família  $\{X_t, t \in T\}$ , tal que, para cada  $t \in T$ ,  $X_t$  é uma variável aleatória.

Nestas condições, um processo estocástico é uma família de variáveis aleatórias, que supomos ser definidas num mesmo espaço de probabilidades  $(\Omega, \mathcal{A}, P)$ . Normalmente, supõe-se que as variáveis aleatórias envolvidas sejam reais, mas elas também podem ser complexas.

O conjunto  $T$  é normalmente tomado como o conjunto dos números inteiros,  $\mathbb{Z} = 0, \pm 1, \pm 2, \dots$ , ou o conjunto dos reais,  $\mathbb{R}$ . Como para cada  $t \in T$ ,  $X_t$  é uma variável aleatória definida sobre  $\Omega$ , temos que  $X_t$  é uma função de dois argumentos,  $X(t, \omega)$ ,  $t \in T$ ,  $\omega \in \Omega$ . Então, para cada  $\omega \in \Omega$  fixado, obteremos uma função de  $t$ .

O conjunto de valores  $\{X_t, t \in T\}$  é chamado de espaço dos estados do processo estocástico, e os valores de  $X_t$  podem ser chamados de estados.

Se o conjunto  $T$  for finito ou enumerável, como  $T = \mathbb{Z}$ , o processo é dito ser a parâmetro discreto. Se  $T$  for um intervalo de  $\mathbb{R}$ , teremos um processo a parâmetro contínuo. No primeiro caso,  $X_t$  pode representar uma contagem como o número de transações de uma ação durante um dia. No segundo caso,  $X_t$  representa uma medida que varia continuamente, como o retorno de um ativo, ou o volume (em reais) negociado em cada dia de uma bolsa de valores.

### 2.1.1 Processos Lineares Estacionários

Dizemos que um processo  $X$  é estacionário se ele se desenvolve no tempo de modo que a escolha de uma origem dos tempos não é importante. Em outras palavras, as características de  $X_{t+\tau}$  para todo  $\tau$ , são as mesmas de  $X_t$ .

Tecnicamente, há duas formas de estacionariedade: fraca e estrita.

**Definição 2.2.** Um processo estocástico  $X = X_t, t \in T$  é dito ser estritamente estacionário se todas as distribuições permanecem as mesmas sob translações no tempo, ou seja,

$$F(x_1, \dots, x_n; t_1 + \tau, \dots, t_n + \tau) = F(x_1, \dots, x_n; t_1, \dots, t_n),$$

para quaisquer  $t_1, \dots, t_n, \tau \in T$ .

Isso significa, em particular, que todas as distribuições unidimensionais são invariantes sob translações no tempo, logo a média  $\mu(t)$  e variância  $\text{Var}(t)$  são constantes, isto é,

$$\mu(t) = \mu \quad \text{e} \quad \text{Var}(t) = \sigma^2,$$

para todo  $t \in T$ .

Do mesmo modo, todas as distribuições bidimensionais dependem de  $t_2 - t_1$ . De fato, como  $\gamma(t_1, t_2) = \gamma(t_1 + t, t_2 + t)$ , fazendo  $t = -t_2$  tem-se que

$$\gamma(t_1, t_2) = \gamma(t_1 - t_2, 0) = \gamma(\tau),$$

para  $\tau = t_1 - t_2$ . Logo,  $\gamma(t_1, t_2)$  é uma função de um só argumento, no caso do processo ser estritamente estacionário. Fazendo  $t = t_1$ , vemos que, na realidade  $\gamma(t_1, t_2)$  é função de  $|t_1 - t_2|$ .

**Definição 2.3.** Um processo estocástico  $X = X_t, t \in T$  é dito ser fracamente estacionário se e somente se

- $E(X_t) = \mu(t) = \mu$ , constante para todo  $t \in T$ ;

- $E(X_t^2) < \infty$ , para todo  $t \in T$ ;
- $\gamma(t_1, t_2) = \text{Cov}(X_{t_1}, X_{t_2})$  é uma função de  $|t_1 - t_2|$ .

A partir de agora estaremos interessados somente nessa classe de processos, que denominaremos simplesmente de processos estacionários.

**Sequência aleatória:** Consideremos  $\{X_n, n = 1, 2, 3, \dots\}$  uma sequência de variáveis aleatórias definidas no mesmo espaço amostral  $\Omega$ . Dizemos que  $\{X_n\}$  é um processo com parâmetro discreto, ou uma sequência aleatória. Para todo  $n \geq 1$ , podemos escrever

$$P(X_1 = a_1, \dots, X_n = a_n) = P(X_1 = a_1) \times P(X_2 = a_2 | X_1 = a_1) \times \dots \times P(X_n = a_n | X_1 = a_1, \dots, X_{n-1} = a_{n-1}), \quad (2.1)$$

em que os  $a_j$ 's representam estados do processo. Logo, o espaço dos estados pode ser tomado como o conjunto dos reais.

**Definição 2.4.** Dizemos que  $\{\varepsilon_t, t \in \mathbb{Z}\}$  é um ruído branco discreto se as variáveis aleatórias  $\varepsilon_t$  são não correlacionadas, isto é,  $\text{Cov}\{\varepsilon_t, \varepsilon_s\} = 0$  para todo  $t \neq s$ .

Um tal processo será estacionário se  $E(\varepsilon_t) = \mu$  e  $\text{Var}(\varepsilon_t) = \sigma^2$ , para todo  $t$ . Obviamente, se as variáveis aleatórias  $\varepsilon_t$  são independentes, elas também serão não correlacionadas. Uma sequência de variáveis aleatórias independentes e identicamente distribuídas, como definida acima, é chamado um processo puramente aleatório.

**Passeio Aleatório:** Considerando uma sequência  $\{\varepsilon_t, t \geq 1\}$ , de variáveis aleatórias independentes e identicamente distribuídas, com distribuição  $N(\mu_\varepsilon, \sigma_\varepsilon^2)$ . Para

$$X_t = \varepsilon_1 + \dots + \varepsilon_t = \sum_{i=1}^t \varepsilon_i, \quad (2.2)$$

segue-se que  $E(X_t) = t\mu_\varepsilon$  e  $\text{Var}(X_t) = t\sigma_\varepsilon^2$ , ou seja, ambas dependem de  $t$ . Logo, a função de autocorrelação de  $X_t$  para dois momentos distintos é dada por

$$\gamma_X(t, s) = \sigma_\varepsilon^2 \min(t, s),$$

- $E(X_t) = t\mu_\varepsilon$

$$\begin{aligned} E(X_t) &= E(\varepsilon_1 + \varepsilon_2 + \dots + \varepsilon_t) \\ &= E\left(\sum_{i=1}^t \varepsilon_i\right) \\ &= \sum_{i=1}^t E(\varepsilon_i) \\ &= t\mu_\varepsilon \end{aligned}$$

- $\text{Var}(X_t) = t\sigma_\varepsilon^2$

$$\begin{aligned}
 \text{Var}(X_t) &= \text{Var}(\varepsilon_1 + \varepsilon_2 + \dots + \varepsilon_t) \\
 &= \text{Var}\left(\sum_{i=1}^t \varepsilon_i\right) \\
 &= \sum_{i=1}^t \text{Var}(\varepsilon_i) \\
 &= t\sigma_\varepsilon^2
 \end{aligned}$$

- $\gamma_X(t, s) = \sigma_\varepsilon^2 \min(t, s)$

Sem perda de generalidade, consideraremos  $t \leq s$ .

$$\begin{aligned}
 \gamma_X(t, s) &= \text{Cov}(X_t, X_s) \\
 &= \text{Cov}\left(\sum_{i=1}^t \varepsilon_i, \sum_{j=1}^s \varepsilon_j\right) \\
 &= \sum_{i=1}^t \text{Cov}\left(\varepsilon_i, \sum_{j=1}^s \varepsilon_j\right) \\
 &= \sum_{i=1}^t \text{Cov}(\varepsilon_i, \varepsilon_i) + \\
 &\quad \sum_{i=1}^t \text{Cov}\left(\varepsilon_i, \sum_{j \neq i}^s \varepsilon_j\right) \\
 &= \sum_{i=1}^t \sigma_\varepsilon^2 + 0 \\
 &= t\sigma_\varepsilon^2
 \end{aligned}$$

Se fizermos o mesmo procedimento, usando  $t \geq s$  chegaremos que

$$\begin{aligned}
 \gamma_X(t, s) &= \text{Cov}(X_t, X_s) \\
 &= s\sigma_\varepsilon^2.
 \end{aligned}$$

Logo,

$$\begin{aligned}
 \gamma_X(t, s) &= \text{Cov}(X_t, X_s) \\
 &= \sigma_\varepsilon^2 \min(t, s).
 \end{aligned}$$

### 2.1.2 Processos Autorregressivos

A análise de séries temporais está envolvida na análise da dinâmica subjacente de um conjunto de valores de tempo passado, observados sucessivamente, chamados séries temporais.

Queremos identificar o processo que determina a série temporal, usando apenas a informação dada pela série. Portanto, o processo é separado em uma parte que podemos determinar ou prever e uma parte aleatória. Normalmente, a parte previsível determina a esperança condicionada por certas variáveis exógenas e a parte aleatória é responsável pelos desvios, ou em outras palavras, a variância.

A maneira mais simples, e provavelmente a mais comum, é construir o processo como uma função de  $p$  valores observados no passado da série temporal. Dessa forma, em geral estima-se essa função  $X_t$  regredindo em seus valores anteriores, esse processo é chamado autorregressivo  $AR(p)$  que é formalmente introduzido na Definição (2.5).

**Definição 2.5.** Um processo estocástico é chamado de processo autorregressivo de ordem  $p$ ,  $AR(p)$ , se for representado pela equação

$$X_t = F(\mathbf{x}_{t-p}) + \varepsilon_t, \quad (2.3)$$

em que  $\mathbf{x}_{t-p} = (X_{t-1}, X_{t-2}, \dots, X_{t-p})^\top$ ,  $F: \mathbb{R}^p \rightarrow \mathbb{R}$  e  $\varepsilon_t$  são variáveis aleatórias independentes e identicamente distribuídas.

*Observação 2.6.* Se  $F(\mathbf{x}_{t-p})$  é uma função linear, o processo é um  $AR(p)$  linear. Se  $F(\mathbf{x}_{t-p})$  não é linear, o processo  $AR(p)$  é não linear.

A influência da parte estocástica é apenas de natureza temporária e não contém tendências dependentes de tempo ou de variância. A esperança condicional de  $X_t$  é  $E(X_t|\mathbf{x}_{t-p}) = F(\mathbf{x}_{t-p})$ , já que a esperança condicional de  $\varepsilon_t$  é  $E(\varepsilon_t|\mathbf{x}_{t-p}) = 0$ . Isso significa que a entrada e a parte estocástica  $\varepsilon_t$  são não correlacionadas. Se um processo é um  $AR(p)$ , podemos dizer que o processo tem uma memória que remonta ao período  $p$ .

Uma outra definição usada por [Morettin \(2008\)](#), diz que  $X_t$  é um processo autorregressivo de ordem  $p$  se satisfizer a equação de diferenças

$$X_t - \mu = \alpha_1(X_{t-1} - \mu) + \alpha_2(X_{t-2} - \mu) + \dots + \alpha_p(X_{t-p} - \mu) + \varepsilon_t, \quad (2.4)$$

onde  $\mu, \alpha_1, \dots, \alpha_p$  são parâmetros reais e  $\varepsilon_t$  são ruídos brancos com distribuição  $N(0, \sigma^2)$ .

Temos que  $E(X_t) = \mu$  se escrevermos o processo da forma

$$X_t = \alpha_0 + \alpha_1 X_{t-1} + \dots + \alpha_p X_{t-p} + \varepsilon_t.$$

Então,

$$\mu = E(X_t) = \frac{\alpha_0}{1 - \alpha_1 - \dots - \alpha_p}.$$

Definimos o operador atraso  $B$  através de  $B^s X_t = X_{t-s}$ ,  $s \geq 1$ . Então a Equação (2.4) pode ser escrita como

$$\alpha(B)\tilde{X}_t = \varepsilon_t,$$

onde  $\alpha(B) = 1 - \alpha_1 B - \alpha_2 B^2 - \dots - \alpha_p B^p$  é o operador autorregressivo de ordem  $p$  e  $\tilde{X}_t = X_t - \mu$ .

$$\begin{aligned} X_t &= BX_t + \varepsilon_t \Rightarrow (1 - B)X_t = \varepsilon_t \\ &\Rightarrow X_t = (1 - B)^{-1} \varepsilon_t \\ &\Rightarrow X_t = (1 + B + B^2 + \dots + B^p) \varepsilon_t \\ &\Rightarrow X_t = \varepsilon_1 + \dots + \varepsilon_t. \end{aligned}$$

As aplicações mostram que, na maioria dos casos, os resíduos dificilmente satisfazem as suposições vistas na Definição (2.5). Uma solução linear desse problema são as classes de modelos autorregressivos de médias móveis, ARMA (do inglês *autoregressive moving average*) que assumem que o processo não consiste apenas em uma parte linear previsível, uma vez que a parte estocástica pode ser determinado por um processo de médias móveis, MA (BOX; JENKINS, 1976).

Um modelo ARMA ( $p, q$ ) é representado pela seguinte equação

$$X_t = \alpha_0 + \alpha_1 X_{t-1} + \alpha_2 X_{t-2} + \dots + \alpha_p X_{t-p} + \varepsilon_t + \psi_1^* \varepsilon_{t-1} + \psi_q^* \varepsilon_{t-q},$$

em que  $q$  indica o atraso máximo da parte MA.

Até hoje, os modelos ARMA são frequentemente aplicados na análise de séries temporais. O teorema da decomposição visto em Wold (1938) justifica que teoricamente pode-se estimar qualquer processo estacionário de covariância por um modelo ARMA. No entanto, de acordo com Lütkepohl e Tschernig (1996), os processos ARMA são apenas os melhores estimadores lineares. Às vezes, usar uma transformação logarítmica podem ajudar a linearizar alguns efeitos não lineares, mas a informação pode ser perdida pela transformação. A alternativa são os modelos não lineares (FAN; YAO, 2008).

### 2.1.3 Processos não lineares Autorregressivos

Modelos não lineares tentam superar o problema de características não padronizadas observadas nos modelos lineares. Eles podem ser interpretados como uma alternativa para modelos lineares com extensões na parte estocástica (ARMA), à medida que tentam melhorar a parte previsível para explicar o processo em vez de adicionar alguns componentes estocásticos ou introduzir algumas suposições que são difíceis de entender ou lidar. Por outro lado, é possível que um AR não linear tenha o  $\varepsilon_t$  em conformidade com as suposições da Definição (2.5). A modelagem não linear nos permite pensar em puros processos determinísticos. Por um lado, modelos não lineares são mais flexíveis do que modelos lineares, mas podem ser difícil a interpretação dos seus parâmetros (MEDEIROS; TERÄSVIRTA; RECH, 2006).

A totalidade das técnicas de modelagem não linear é grande. O primeiro passo para classificá-los é distinguir entre métodos paramétricos, semi paramétricos e não paramétricos.



A forma paramétrica significa que a estrutura da função para estimar e o número dos parâmetros relacionados são conhecidos. Modelos não paramétricos não se restringe a uma função qualquer de forma específica, e modelos semi paramétricos são descritos como uma combinação de partes paramétricas e não paramétricas (GRANGER; TERÄSVIRTA, 1993).

## 2.2 Redes Neurais

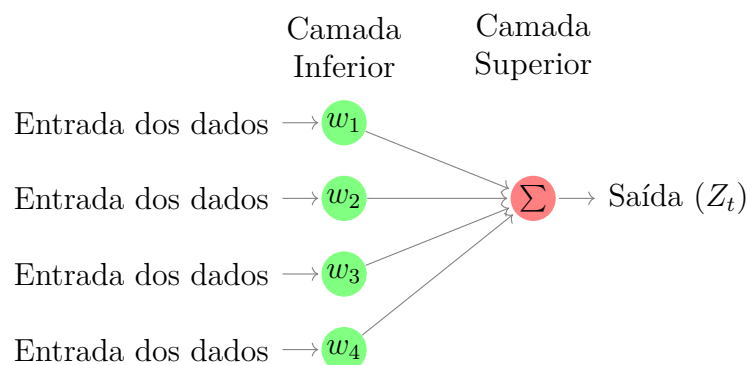
Uma rede neural pode ser pensada como uma rede de neurônios organizados em camadas. Os preditores, ou entradas, formam a camada inferior e as previsões, ou saídas, formam a camada superior. Também pode haver camadas intermediárias chamadas de camadas ocultas.

Kuan (2018), Granger e Teräsvirta (1993) classificam as redes neurais como modelos paramétricos, pois o modelo deve ser especificado. Como veremos na seção (2.3.3), as redes neurais possuem uma propriedade de aproximação universal. Isso significa que elas são capazes de aproximar qualquer função não especificada arbitrariamente com precisão. Essa propriedade pode ser vista como evidência para um modelo não paramétrico.

### 2.2.1 Redes Perceptron Simples

As redes neurais simples não contêm camadas ocultas e são equivalentes a regressões lineares. A Figura (1) mostra a versão da rede neural de uma regressão linear com quatro preditores. Os coeficientes ligados a esses preditores são chamados de pesos, e as previsões são obtidas por uma combinação linear das entradas. Os pesos são selecionados na estrutura da rede neural usando um algoritmo de aprendizagem que minimiza uma função custo. Os detalhes serão abordados na seção (3.3).

Figura 1 – Redes Neurais Simples.



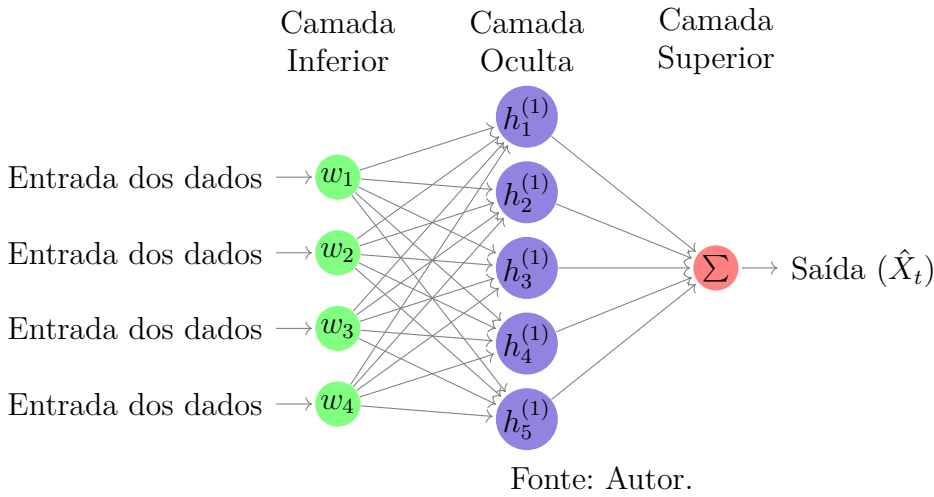
Fonte: Autor.

Para um melhor entendimento podemos considerar a Figura (1) como uma média ponderada com pesos  $w_i$ , obtidos do processo de aprendizagem da rede neural.

### 2.2.2 Redes Perceptron Multicamadas (*Feed-Forward*)

Com apenas um neurônio não se pode fazer muita coisa, mas podemos combiná-los em uma estrutura em camadas, cada uma com número diferente de neurônios, formando uma rede neural denominada Perceptron Multicamadas. Uma vez que adicionado uma camada intermediária com camadas ocultas, a rede neural se torna não linear. Um exemplo de redes perceptron multicamadas é mostrado na Figura (2).

Figura 2 – Redes Neurais *feed-forward*.



A Figura (2) é conhecida também como redes *feed-forward*, onde cada camada de neurônios recebe entradas das camadas anteriores. As saídas dos neurônios em uma camada são entradas para a próxima camada. As entradas para cada neurônio são obtidas a partir de uma combinação linear ponderada. O resultado é então modificado por uma função não linear antes da saída. Por exemplo, as entradas no neurônio oculto  $h_j^{(1)}$ , na Figura (2), são combinadas linearmente para fornecer

$$h_j^{(1)} = b_j + \sum_{i=1}^4 w_i x_i.$$

Na camada oculta, isso é modificado usando uma função não linear para fornecer a entrada para a próxima camada. Os parâmetros  $b_j$  e  $w_i$  são estimados a partir dos dados. Os valores dos pesos são frequentemente restritos para evitar que eles se tornem muito grandes. Os pesos usam valores aleatórios para inicializar, e eles são atualizados usando os dados observados. Consequentemente, há um elemento de aleatoriedade nas

previsões produzidas por uma rede neural. A rede geralmente é treinada várias vezes usando diferentes pontos de partida aleatório, e os resultados são calculados. Ver com mais detalhes na seção (3.3).

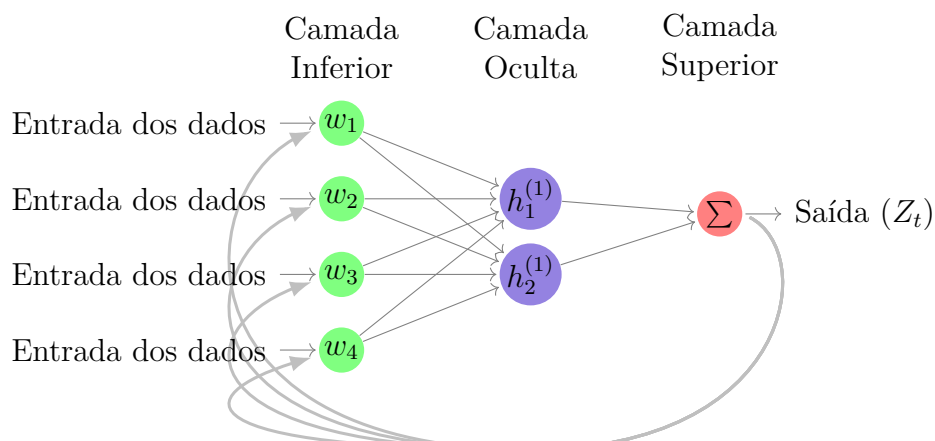
### 2.2.3 Redes Recorrentes *Feed-backward*

A estrutura de uma rede neural recorrente tem por base uma rede *feed-forward* com algumas modificações, notoriamente a introdução da realimentação, o que amplia sua potencialidade de modelagem de dados temporais ou espaciais (GOMES, 2005).

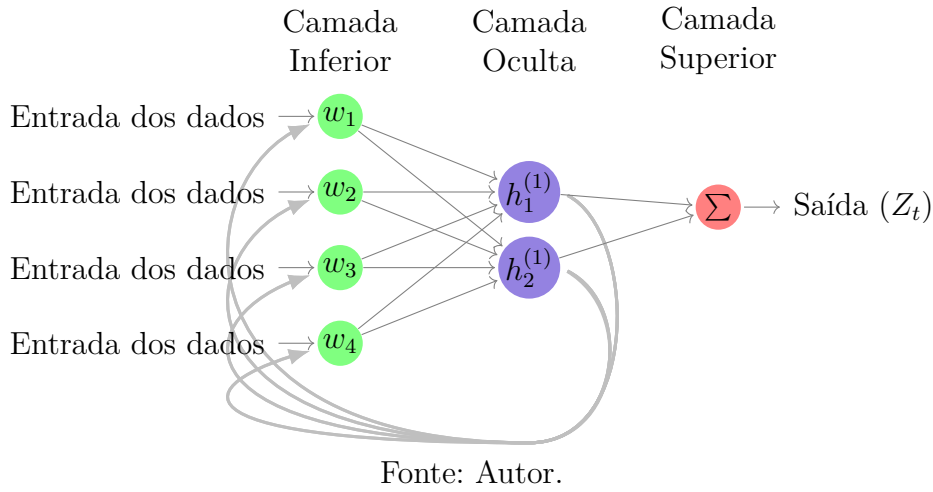
Dessa forma, uma rede *feed-forward* pode ser vista como um caso particular de uma rede recorrente. As realimentações consistem em saídas de neurônios de uma determinada camada para serem reintroduzidas como entradas de neurônios de camadas anteriores ou da própria camada. Essas possibilidades fazem com que a arquitetura de uma rede recorrente possa tomar diversas formas. Com esses tipos de rede, ampliam-se as possibilidades de modelagem de estruturas de auto-dependência de dados, no caso das séries temporais, não se limita apenas às estruturas autorregressivas.

Para o caso de predição de séries temporais, duas das redes recorrentes mais utilizadas são as do tipo **Elman** e do tipo **Jordan**. As saídas da camada intermediária (Elman) ou da camada de saída (Jordan) são defasadas e reintroduzidas como entradas da rede. (GOMES, 2005)

Figura 3 – Redes Neurais Recorrentes: **Jordan**.



Fonte: Autor.

Figura 4 – Redes Neurais Recorrentes: **Elman**.

### 2.2.4 Modelo Matemático para as Redes Neurais

Figuras associadas às redes neurais podem ser extremamente confusas, por isso é interessante pensar as mesmas como aninhamentos sucessivos de diversas transformações lineares seguidas por alguma função diferenciável, que é aplicada elemento a elemento da matriz de entrada. Para melhor entendê-las, vamos partir de uma rede neural bem simples: Um modelo de regressão linear, que pode ser entendido como uma rede neural. Essa dedução é vista na Equação (2.6). Uma rede neural simples sem nenhum neurônio oculto é dada pela equação

$$\mathbf{Z} = \mathbf{X}\mathbf{w} + \boldsymbol{\varepsilon},$$

$$\begin{bmatrix} Z_0 \\ Z_1 \\ \vdots \\ Z_n \end{bmatrix} = \begin{bmatrix} 1 & x_{11} & \dots & x_{1n} \\ 1 & x_{21} & \vdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{d1} & \dots & x_{dn} \end{bmatrix} \times \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_n \end{bmatrix} + \begin{bmatrix} \varepsilon_0 \\ \varepsilon_1 \\ \vdots \\ \varepsilon_n \end{bmatrix}.$$

onde  $\mathbf{X}$  é a matriz de entrada,  $\mathbf{w}$  a matriz coluna de pesos e  $\boldsymbol{\varepsilon}$  a parte aleatória.

Para adicionar neurônios a essa rede neural, deve-se multiplicar a matriz de entrada por uma matriz de neurônios, e essa multiplicação de matrizes, por mais um vetor de pesos, mantendo a consistência da saída. Temos assim, um modelo de uma rede neural com uma camada e  $n$  neurônios como

$$\mathbf{Z}_t = (\mathbf{X}\mathbf{W}_1)\mathbf{w} + \boldsymbol{\varepsilon},$$

em que

$$\begin{bmatrix} Z_0 \\ Z_1 \\ \vdots \\ Z_n \end{bmatrix} = \begin{bmatrix} 1 & x_{11} & \dots & x_{1d} \\ 1 & x_{21} & \vdots & x_{2d} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & \dots & x_{nd} \end{bmatrix} \times \begin{bmatrix} W_{01} & W_{02} & \dots & W_{0n} \\ W_{11} & W_{12} & \vdots & W_{1n} \\ \vdots & \vdots & \ddots & \vdots \\ W_{d1} & W_{d2} & \dots & W_{dn} \end{bmatrix} \times \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_n \end{bmatrix} + \begin{bmatrix} \varepsilon_0 \\ \varepsilon_1 \\ \vdots \\ \varepsilon_n \end{bmatrix}.$$

É importante perceber que a matriz  $\mathbf{W}_1$  representa a camada oculta da rede neural, e cada coluna dessa matriz é um neurônio da camada oculta. Podemos pensar no vetor  $\mathbf{w}$  como uma camada de saída com um único neurônio, que recebe o sinal dos neurônios anteriores, ponderando-os e produzindo a saída final da rede.

A rede neural acima não é muito interessante do ponto de vista prático, pois só consegue representar funções lineares. Esse problema pode ser contornado introduzindo uma das funções de ligação que veremos mais adiante. Com isso, alteramos o modelo da seguinte forma:

$$F(\mathbf{X}\mathbf{W}_1)\mathbf{w} = \mathbf{y},$$

em que  $F$  é alguma função não linear diferenciável denominado de Função de ativação.

As funções de ativação são essenciais para dar capacidade representativa às redes neurais artificiais, introduzindo um componente de não linearidade.

Considerando o modelo matemático de uma rede neural com duas camadas ocultas:

$$\mathbf{Z} = F(G(\mathbf{X}\mathbf{W}_1)\mathbf{W}_2)\mathbf{w} + \boldsymbol{\varepsilon}, \quad (2.5)$$

em que  $\mathbf{X}$  é a matriz de dados,  $\mathbf{W}_i$  são as camadas ocultas,  $\mathbf{w}$  são os pesos da camada de saída, e  $F$  e  $G$  são quaisquer funções de ativação. Se considerarmos  $F$  e  $G$  como identidades, a Equação (2.5) se transforma em uma regressão simples.

$$\begin{aligned} \mathbf{Z} &= \mathbf{X}\mathbf{W}_1\mathbf{W}_2\mathbf{w} + \boldsymbol{\varepsilon} \\ &= \mathbf{X}\mathbf{v} + \boldsymbol{\varepsilon}, \end{aligned} \quad (2.6)$$

em que  $\mathbf{v}$  será o resultado das multiplicações  $\mathbf{W}_1\mathbf{W}_2\mathbf{w}$ . Mais ainda,  $\mathbf{v}$  será um vetor coluna, com o mesmo número de linhas que variáveis em  $\mathbf{X}$ . Agora, note como uma rede neural sem função ativação é simplesmente a componente determinística de um modelo de regressão linear. Assim, essa rede neural sem as funções de ativação estão sujeitas às mesmas restrições que os modelos lineares.

### 2.2.5 Função de Ativação

Antes de abordarmos as funções de ativação individualmente, é importante entender porque precisamos delas. Intuitivamente, as funções de ativação introduzem um componente

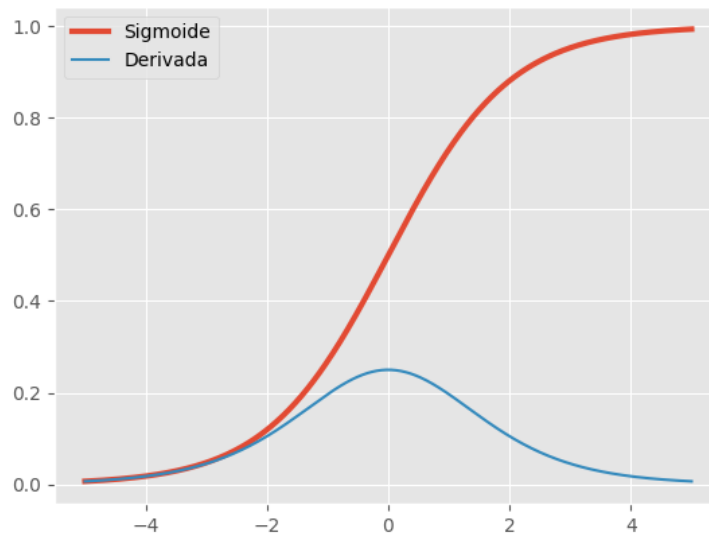
não linear nas redes neurais, fazendo com que elas possam aprender mais do que relações lineares entre as variáveis dependentes e independentes. O modelo de cada camada da rede pode incluir uma não linearidade na sua saída. É relevante enfatizar que a função não linear deve ser suave, ou seja, diferenciável. As funções de ativação são essenciais para dar capacidade representativa às redes neurais artificiais. Por outro lado, com esse poder a mais, surgem algumas dificuldades. A seguir serão descritos alguns tipos de funções de ativação.

### 2.2.5.1 Função Logística

A seguir, vê-se a função logística e sua derivada, assim como seu gráfico (Figura 5).

$$\Psi_{logistic}(\cdot) = \frac{1}{1 + e^{-(\cdot)}} \quad \text{e} \quad \Psi'_{logistic}(\cdot) = \Psi_{logistic}(\cdot)(1 - \Psi_{logistic}(\cdot)).$$

Figura 5 – Função de Ativação Logística.



Fonte: Repositório Digital do GitHub.

A função logística é suave e continuamente diferenciável. A função é não linear, isto significa essencialmente que quando se tem vários neurônios com a função logística como função de ativação a saída também é não linear.

A função varia de 0 a 1 tendo o gráfico em formato de *S*. Como neurônios biológicos funcionam de forma binária (ativando *versus* não ativando), a função logística é uma boa forma de modelar dados binários, já que assume valores apenas entre 0 (não ativação) e 1 (ativação). No entanto, se olharmos sua derivada podemos ver que ela satura (tende a 0) para valores abaixo de -5 e acima de 5, o que é um problema. Com essas derivadas

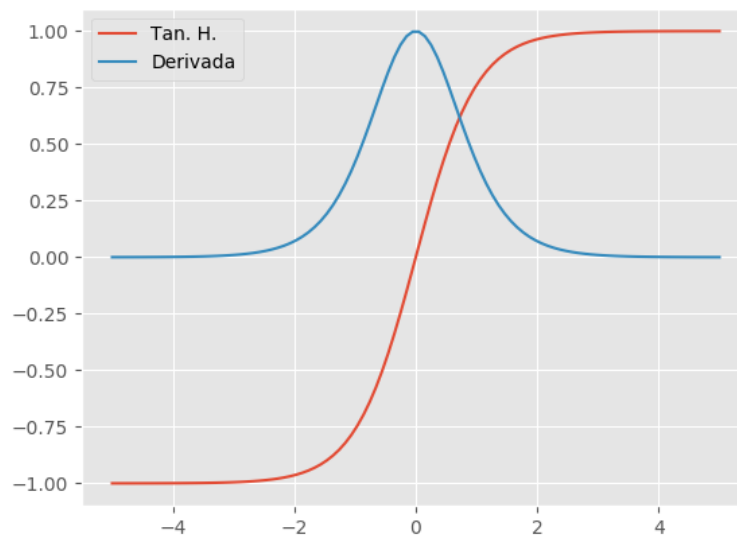
tendendo a zero, a propagação do gradiente desvanece nessas regiões, causando dificuldades no treinamento, isso significa que o gradiente está se aproximando de zero e a rede não está realmente aprendendo.

### 2.2.5.2 Função Tangente Hiperbólica

Similar à função logística, a função tangente hiperbólica ( $\tanh$ ) também tem o gráfico em formato de  $S$ , mas varia de -1 a 1. A função  $\tanh$  e sua derivada são dadas por:

$$\Psi_{\tanh}(\cdot) = 2\Psi_{\text{logistic}}(2(\cdot)) - 1 \quad \text{e} \quad \Psi'_{\tanh}(\cdot) = 1 - \tanh^2(\cdot).$$

Figura 6 – Função de Ativação Tangente Hiperbólica.



Fonte: Repositório Digital da GitHub.

Podemos ver na Figura (6) que as saturações ainda estão presentes, mas o valor da derivada é maior, chegando ao máximo de 1 quando a abscissa vai para zero ( $x = 0$ ).

### 2.2.5.3 Função ReLU

ReLU é uma função de ativação não linear, cuja a principal vantagem sobre outras funções de ativação, é que ela não ativa todos os neurônios ao mesmo tempo. O que significa que podemos olhar para a função ReLU e se a entrada for negativa, ela será convertida em zero e o neurônio não será ativado. Isso significa que ao mesmo tempo, apenas alguns neurônios são ativados.

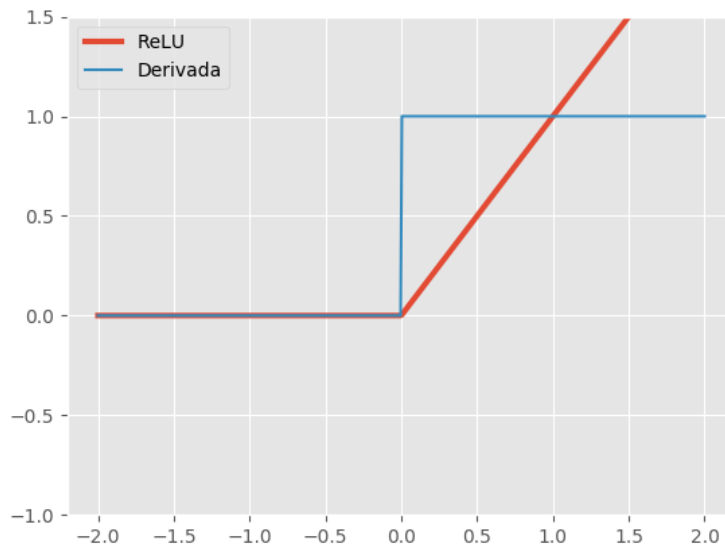
Redes com a função ReLU são fáceis de otimizar, já que a mesma é extremamente parecida com a função identidade. Como a ReLU produz uma grande quantidade de zeros

no seu domínio, temos que o gradiente se mantém no máximo enquanto a unidade estiver ativa.

A seguir, vê-se a função ReLU e sua derivada, assim como seu gráfico Figura (7).

$$\Psi_{ReLU}(x) = \max\{0, x\} \quad \text{e} \quad \Psi'_{ReLU}(x) = \begin{cases} 1, & \text{se } x \geq 0 \\ 0, & \text{c.c.} \end{cases}$$

Figura 7 – Função de Ativação ReLU.



Fonte: Repositório Digital da GitHub.

Teoricamente, a derivada não está definida em 0, mas podemos implementá-la como sendo 0 ou 1 sem maiores preocupações. Note que as derivadas são estáveis, sendo 1, quando  $x > 0$  e 0 quando  $x < 0$ . Note também que a segunda derivada é zero em todo o domínio, exceto para  $x = 0$ .

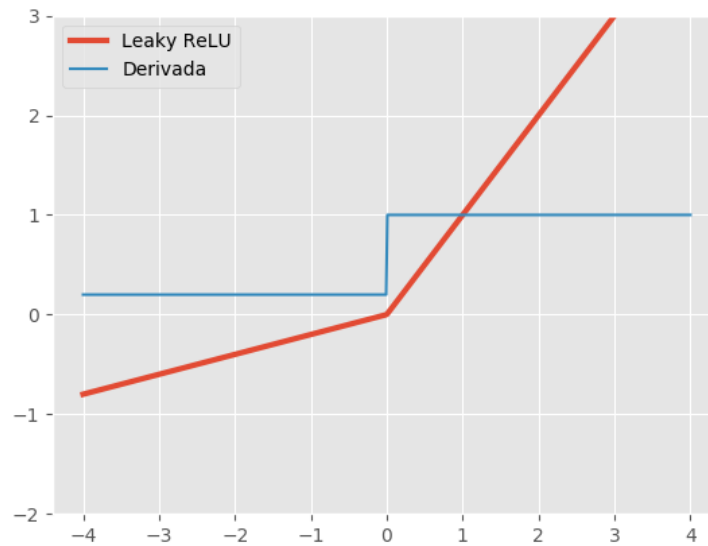
#### 2.2.5.4 Função *Leaky ReLU*

Para solucionar o problema dos zeros na função ReLU, uma proposta é introduzir uma pequena inclinação  $\alpha$  para a função na parte negativa do seu domínio. A seguir, vê-se a função *Leaky ReLU* e sua derivada, assim como seu gráfico Figura (8).

$$\Psi_{LeakyReLU}(x, \alpha) = \max\{\alpha x, x\}, \quad \alpha > 0 \quad \text{e} \quad \Psi'_{LeakyReLU}(x, \alpha) = \begin{cases} 1, & \text{se } x \geq 0 \\ \alpha, & \text{c.c.} \end{cases}$$



Figura 8 – Função de Ativação Leaky ReLU.



Fonte: Repositório Digital da GitHub.

Novamente, a função *Leaky* ReLU é bastante parecida com a função identidade e tem as derivadas estáveis. A novidade aqui é que a derivada na região negativa ainda é positiva, determinada por um hiper-parâmetro  $\alpha$ , chamado de vazamento. Normalmente,  $\alpha$  é algum valor pequeno, como 0,01. Na função ReLU, o gradiente é 0 para  $x < 0$ , o que faz os neurônios morrerem por ativações nessa região. *Leaky* ReLU ajuda a resolver esse problema. Em vez de definir a função ReLU como 0 para  $x < 0$ , definimos como um pequeno componente linear de  $x$ .

## 2.3 Redes Neurais Autorregressivas AR-NN( $p$ )







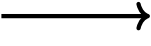
Como destacado na seção (2.2), redes neurais costumam conter uma parte linear e uma não linear. Para facilitar a construção do modelo AR-NN( $p$ ), utilizaremos representações gráficas e, em seguida, a equação matemática do modelo.

Aqui, explicaremos os componentes básicos do teorema da aproximação universal, na versão de [Hornik \(1993\)](#). As funções de ativação são importantes para o modelo, em geral essas funções permitem a análise da estacionariedade, usando métodos lineares. Depois de escolher a função de ativação e a arquitetura da rede, o modelo AR-NN( $p$ ) torna-se uma função paramétrica, e este será o ponto de partida para a construção de modelos, de acordo com o esquema de [Box e Jenkins \(1976\)](#).

### 2.3.1 Forma Gráfica AR-NN( $p$ )

O primeiro passo para entender o modelo AR-NN( $p$ ) é a visualização gráfica. Os gráficos usados aqui são semelhantes aos apresentados por [Anders \(1997\)](#) e [Haykin \(2009\)](#). Eles servem como inspiração para outros modelos, e dão uma visão mais profunda para redes complicadas. Inicialmente, começamos com um gráfico de uma AR linear. Na análise de séries temporais lineares, o termo camada é desconhecido. Logo, a Tabela (1) mostra os termos nos contextos estatísticos e de redes neurais.

Tabela 1 – Termos em Diferentes Áreas.

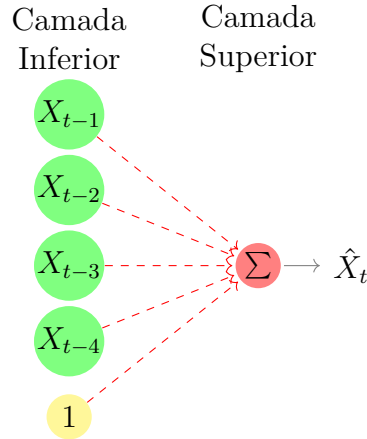
Símbolo	Termo na Estatística	Termo em Redes Neurais	Função Equivalente
	Variáveis independentes	Camada de entrada/ neurônios	$X_{t-i}$
	Variáveis dependentes	Camada de saída/ saída	$\hat{X}_t$
	Função não linear	Camada oculta/ neurônio oculto	$\Psi \left( \gamma_{0j} + \sum_{i=1}^p \gamma_{ij} X_{t-i} \right)$
	Constante/Viés	<i>Bias</i>	1, Para ser multiplicado por $\alpha_0$
	Parâmetros	Pesos	$\alpha$
	Parâmetros	Peso entre a entrada e a camada oculta	$\gamma_{ij}$
	Parâmetros	Peso entre a camada oculta e saída	$\beta_j$

Fonte: Autor.

Para o gráfico de um modelo AR(4) linear, utilizamos as seguintes camadas: A camada de entrada, que contém todas as variáveis independentes e a camada de saída, que contém a variável dependente. Note também que a constante é decomposta em um neurônio de valor 1, conhecido como *bias*. O mesmo serve para representar o parâmetro  $\alpha_0$  em um modelo AR. Por exemplo, considerando um modelo AR(4) linear e sua forma gráfica, Figura(9), temos

$$\text{AR}(4) = \alpha_0 + \alpha_1 X_{t-1} + \alpha_2 X_{t-2} + \alpha_3 X_{t-3} + \alpha_4 X_{t-4}.$$

Figura 9 – Representação gráfica de um modelo AR(4).



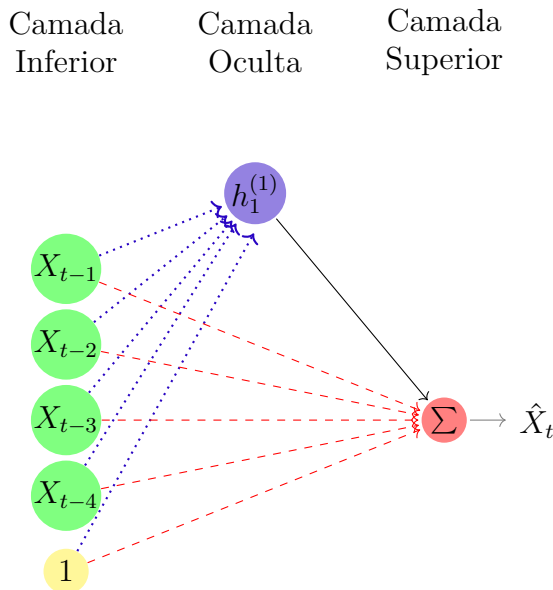
Fonte: Autor.

Como mencionado na seção (2.1.2), um AR linear às vezes não é suficiente e, portanto, precisa ser aumentado para uma parte não linear. A totalidade dessa parte não linear é chamada de camada oculta. Dentro da camada oculta, as variáveis são transformadas por uma função não linear. O resultado desta transformação não linear é adicionado ao resultado da parte linear. Seja  $F(\cdot)$  uma função não linear, então a extensão não linear de um AR(4) é dada por

$$\text{AR-NN}(4) = \alpha_0 + \alpha_1 X_{t-1} + \alpha_2 X_{t-2} + \alpha_3 X_{t-3} + \alpha_4 X_{t-4} + F(X_{t-1}, X_{t-2}, X_{t-3}, X_{t-4}),$$

e sua forma gráfica pela Figura (10).

Figura 10 – Representação gráfica do AR-NN(4).



Fonte: Autor.

Como definido na seção (2.2), a parte não linear contém os chamados neurônios ocultos que transformam as variáveis de entrada, ponderando-as pelos parâmetros  $\gamma_{ij}$  mais um viés  $\gamma_{0j}$ , através de uma função de ativação não linear  $\Psi(\cdot)$ . Dessa forma na Equação (2.7), o índice  $i$  indica o número de defasagens e o índice  $j$  o número de camadas ocultas. Uma camada oculta é denotada por

$$\Psi \left( \gamma_{0j} + \sum_{i=1}^n \gamma_{ij} X_{t-i} \right). \quad (2.7)$$

Cada neurônio oculto é ponderado por  $\beta_j$ , antes de ir para camada de saída. Por exemplo, se assumir um AR-NN(2),

$$\text{AR-NN}(2) = \alpha_0 + \alpha_1 X_{t-1} + \alpha_2 X_{t-2} + F(X_{t-1}, X_{t-2}), \quad (2.8)$$

e considerando dois neurônios ocultos, a função não linear fica definida da forma

$$\begin{aligned} F(X_{t-1}, X_{t-2}) = & \Psi(\gamma_{01} + \gamma_{11}X_{t-1} + \gamma_{21}X_{t-2})\beta_1 + \\ & \Psi(\gamma_{02} + \gamma_{12}X_{t-1} + \gamma_{22}X_{t-2})\beta_2. \end{aligned}$$

Na maioria dos casos,  $\Psi(\cdot)$  é a mesma para todos os neurônios ocultos, mas também pode ser escolhida para ser diferente para cada neurônio oculto. No entanto, isso não é uma prática comum e leva a complicações nos procedimentos de estimação.

De maneira geral, todos os AR-NN( $p$ ) são construídos com uma única camada oculta, isso será o suficiente para garantir a propriedade de aproximação universal das redes. Os detalhes serão abordados na seção (2.3.3).

### 2.3.2 Equação AR-NN( $p$ )

Na seção (2.3.1) foi conhecida a estrutura dos modelos AR-NN( $p$ ) na forma gráfica. A partir disso, entraremos na elaboração das equações dos mesmos. Por exemplo, para um AR-NN(2), a equação da rede fica dada por

$$\begin{aligned} X_t = & \alpha_0 + \alpha_1 X_{t-1} + \alpha_2 X_{t-2} + \\ & \Psi(\gamma_{01} + \gamma_{11}X_{t-1} + \gamma_{21}X_{t-2})\beta_1 + \\ & \Psi(\gamma_{02} + \gamma_{12}X_{t-1} + \gamma_{22}X_{t-2})\beta_2, \end{aligned}$$

e se a parte estocástica (componente aleatória) estiver incluída, podemos reescrever de maneira geral como

$$X_t = \alpha_0 + \sum_{i=1}^p \alpha_i X_{t-i} + \sum_{j=1}^h \Psi \left( \gamma_{0j} + \sum_{i=1}^p \gamma_{ij} X_{t-i} \right) \beta_j + \varepsilon_t. \quad (2.9)$$

Em algumas literaturas, podemos encontrar as redes neurais sem a parte linear, porém nessa monografia a mesma será incluída. Como mencionamos na introdução, nosso objetivo é melhorar os modelos lineares, aumentando-os para uma parte não linear.

Uma outra forma de representar os modelos AR-NN(p) é utilizando vetores, assim podemos rescrever a Equação (2.9) na representação vetorial

$$X_t = \alpha_0 + \mathbf{A}^\top \mathbf{x}_{t-p} + \sum_{j=1}^h \Psi(\gamma_{0j} + \mathbf{\Gamma}_j^\top \mathbf{x}_{t-p}) \beta_j + \varepsilon_t, \quad (2.10)$$

em que

$$\mathbf{A} = (\alpha_1, \alpha_2, \dots, \alpha_p)^\top, \quad \mathbf{\Gamma}_j = (\gamma_{1j}, \gamma_{2j}, \dots, \gamma_{pj})^\top, \quad \mathbf{x}_{t-p} = (X_{t-1}, \dots, X_{t-h})^\top \text{ e } \varepsilon_t = (\varepsilon_1, \varepsilon_2, \dots, \varepsilon_p)^\top.$$

Também podemos rescrever a Equação (2.10) usando  $\Theta$ , onde o mesmo é definido como

$$\Theta = (\alpha_0, \mathbf{A}^\top, \gamma_{01}, \dots, \gamma_{0h}, \beta_1, \dots, \beta_j, \mathbf{\Gamma}_1^\top, \dots, \mathbf{\Gamma}_h^\top)^\top.$$

Logo, rescrevemos da forma

$$X_t = G(\Theta, \mathbf{x}_{t-p}) + \varepsilon_t,$$

onde a dimensão de  $\Theta$  é  $(r \times 1)$  com  $r = h \cdot (p + 2) + p + 1$ .

Existem algumas considerações sobre a seleção do número de camadas ocultas, uma abordagem comum é discernir a rede para um número arbitrário de neurônios ocultos e, posteriormente, testar a significância de cada neurônio oculto. Os detalhes serão abordados na seção (3.4). Outra abordagem, proposta por Anders (1997) para definir o número de neurônios ocultos, é considerar a mediana das variáveis de entrada e saída. É claro que este método não leva em conta nenhuma necessidade técnica como a estrutura específica dos dados e da reação da função de ativação nas entradas. Portanto, não é realmente uma ferramenta prática.

Um método consistente com o procedimento de aumentar um AR linear para uma parte não linear, se os dados são não lineares, é estender o número de neurônios ocultos passo a passo: no início, apenas um neurônio oculto é adicionado e, em seguida, é testado de baixo para cima (*Bottom-up*), para ver se um neurônio oculto adicional melhoraria o modelo. Os detalhes serão abordados na seção (3.4.1).

### 2.3.3 O Teorema da Aproximação Universal

Na teoria matemática de redes neurais artificiais, o teorema da aproximação universal declara que uma rede neural pré-alimentada com uma única camada oculta que contém um número finito de neurônios pode aproximar funções contínuas em subconjuntos compactos de  $\mathbb{R}^m$ , com pressupostos mínimos de função de ativação. O teorema afirma que redes neurais simples podem representar uma grande variedade de funções interessantes quando há os parâmetros adequados.

Uma das primeiras versões do teorema foi provado por Cybenko (1989) para função de ativação sigmóide. Cybenko (1989) mostrou pela primeira vez que uma rede com uma única camada intermediária é suficiente para aproximar uniformemente qualquer função contínua definida num hipercubo unitário.

O teorema da aproximação universal afirma que as redes *feed-forward* com pelo menos uma camada oculta que contém um número finito de neurônios ocultos e com função de ativação arbitrária são aproximadores universais do espaço das funções contínuas  $C(I^m)$ .

Hornik (1993) mostrou que não é a escolha específica da função de ativação, mas a própria arquitetura de *feed-forward* que dá às redes neurais o potencial de serem aproximadores universais.

**Teorema 2.7.** *Suponha  $\Psi(\cdot)$  uma função de ativação qualquer e considere que  $I^m$  represente um hipercubo unitário  $[0, 1]^m$  de dimensão  $m$ . O espaço das funções contínuas em  $I^m$  é representado por  $C(I^m)$ . Então, dada qualquer função  $f \in C(I^m)$  e  $\varepsilon > 0$ , existe um inteiro  $M$  e existe um conjunto de constantes reais  $a_{ij}$ ,  $b_i$ ,  $w_i \in \mathbb{R}$ , em que  $i = 1, \dots, n$  e  $j = 1, \dots, m$  tal que podemos definir*

$$(A_n f)(x_1, \dots, x_m) = \sum_{i=1}^n w_i \Psi \left( \sum_{j=1}^m a_{ij} x_j + b_i \right),$$

como uma realização aproximada da função  $f(\cdot)$ , isto é,

$$\|f(x_1, \dots, x_m) - A_n f(x_1, \dots, x_m)\| < \varepsilon, \quad \forall (x_1, \dots, x_m) \in I^m.$$

Como medir a precisão da aproximação depende de como medir a proximidade entre as funções, que por sua vez varia conforme o problema específico a ser tratado. Em muitas aplicações, é necessário ter a rede simultaneamente bem em todas as amostras de entrada. Nesse caso, a proximidade é medida pela distância uniforme entre as funções, ou seja:

$$\|f - A_n f\|_\infty = \sup_{x \in I^m} |f(x) - A_n f(x)|.$$

Aplicando o Teorema (2.7) em termos do AR-NN( $p$ ), temos que algumas notações devem ser introduzidas: Seja  $\mathcal{W} \subseteq \mathbb{R}^m$  o espaço de pesos de modo que todos os  $\Gamma_j \in \mathcal{W}$  e  $\mathcal{B} \subseteq \mathbb{R}$  o espaço de viés de modo que todos os  $\gamma_{0j} \in \mathcal{B}$ . Então  $\mathcal{G}(\Psi; \mathcal{B}, \mathcal{W})$  é conjunto de todas as funções definidas como,

$$G(\Theta, \mathbf{x}_{t-p}) = \sum_{j=1}^h \Psi(\gamma_{0j} + \Gamma_j^\top \mathbf{x}_{t-p}) \beta_j$$

que se aproximam do verdadeiro  $F(\mathbf{x}_{t-p})$ . Em outras palavras,  $\mathcal{G}(\Psi; \mathcal{B}, \mathcal{W})$  é o conjunto de todas as funções que podem ser implementadas por uma rede neural com vieses em  $\mathcal{B}$  e pesos em  $\mathcal{W}$ .

Considerando a Equação (2.3) como uma função não linear não especificada, não importa que tipo de forma possa ter, os neurônios ocultos no AR-NN( $p$ ) podem aproximá-la. Também é preciso ter cuidado com o número de neurônios. Por exemplo, Lütkepohl e Tschernig (1996) geram dados de um AR(3) e estimam o processo por um AR-NN( $p$ ) com o número variável de neurônios ocultos,  $h = 0, \dots, 5$ . Ao fazerem essa simulação, um modelo com  $h = 1$  é escolhido, e portanto, o modelo linear não é identificado. Consequentemente, o trabalho da rede neural apenas se aproximou, não identificando a verdadeira equação. Esse fato diz intuitivamente que o AR-NN( $p$ ) pode ser um modelo mal especificado, capaz de fornecer uma boa aproximação.

## 2.4 Estacionariedade do Modelo

Antes que o esquema clássico de Box e Jenkins (1976), que consiste em seleção de variáveis, estimativa de parâmetros e validação de modelo possa ser aplicado às séries temporais, deve-se testar à estacionariedade e, eventualmente, pré-processado para uma representação estacionária (geralmente por diferenciação). Esta seção começa com uma definição geral de estacionariedade. Além disso, são apresentados os achados importantes de Trapletti, Leisch e Hornik (2000) sobre testes de estacionariedade em modelos AR-NN( $p$ ). Uma breve introdução ao princípio dos testes de raiz unitária é feita com ênfase na modificação do teste Dickey-Fuller (ADF) para ambientes não lineares, o teste *Rank*-ADF (RADF) de Granger e Hallman (1991). Este teste pode ser usado como um teste de estacionariedade, especialmente para séries temporais não lineares.

### 2.4.1 Estacionariedade dos modelos AR-NN

Como visto na seção (2.1.2), temos que a equação de modelo AR( $p$ ) é definido como

$$X_t = \alpha_1 X_{t-1} + \alpha_2 X_{t-2} + \dots + \alpha_p X_{t-p} + \varepsilon_t. \quad (2.11)$$

Com isso, podemos reescrever a Equação (2.11), deixando o componente aleatório como a variável dependente

$$\varepsilon_t = X_t - \alpha_1 X_{t-1} - \alpha_2 X_{t-2} - \dots - \alpha_p X_{t-p}.$$

Então, a equação do polinômio característico do processo AR( $p$ ) é definida por

$$1 - \alpha_1 B - \alpha_2 B^2 - \dots - \alpha_p B^p = 0, \quad (2.12)$$

em que as soluções da Equação (2.12) são denominadas raízes. O processo é fracamente estacionário se as raízes estiverem fora do círculo unitário, ou seja  $|B| > 1$ . Uma condição equivalente à condição de que as raízes estejam fora do círculo unitário é  $|\alpha_i| < 1 \quad \forall i = 1, \dots, p$  (KREIB; NEUHAUS, 2006);(HATANAKA, 1996).

Se o processo tiver suas raízes fora do círculo unitário, ele poderá ser invertido para uma representação de médias moveis (MA) infinita, com base nos resíduos. Nesse caso, pode-se facilmente mostrar que o processo  $X_t$  é estacionário, porque depende apenas de ruídos brancos. Portanto, reescrevemos a Equação (2.11) usando o operador de atraso  $B$

$$\begin{aligned} X_t &= (1 - \alpha_1 B - \dots - \alpha_n B^p) X_t + \varepsilon_t \\ &= \alpha(B) X_t + \varepsilon_t. \end{aligned} \quad (2.13)$$

O processo possui uma representação de médias móveis (MA) infinita se a inversa  $\alpha^{-1}(B)$  existir. Logo a Equação (2.13) é reescrita como

$$X_t = \alpha^{-1}(B) \varepsilon_t = \sum_{i=1}^{\infty} \alpha^i \varepsilon_{t-i},$$

em que a inversa só existe se o  $|B| < 1$  (KREIB; NEUHAUS, 2006).

Quando o processo tem uma raiz unitária, o polinômio  $\alpha(B)$  pode ser dividido pelo filtro  $(1 - B)$ . Ou seja, o filtro estacionariza o processo com raiz unitária. Sem estacionarização, o processo não possui representação de médias móveis (MA) infinita e não é estacionário. Um teorema importante referente à estacionariedade dos modelos AR-NN( $p$ ) foi formulado por Trapletti, Leisch e Hornik (2000).

**Teorema 2.8.** *Seja um processo AR-NN( $p$ ), em que  $\varepsilon_t$  é um processo gaussiano e  $\Psi$  uma função de ativação limitada. O polinômio característico da parte linear é denotado como*

$$\alpha(B) = 1 - \sum_{i=1}^p \alpha_i B^i.$$

*A condição*

$$\alpha(B) \neq 0 \quad \forall B, \quad |B| \leq 1,$$

*é suficiente, mas não necessária para que o processo AR-NN( $p$ ) seja geometricamente ergódico e assintoticamente estacionário.*

Será brevemente explicado por que a estacionariedade fraca da parte linear não é uma condição necessária. Se uma raiz estiver no círculo unitário, esperamos um processo de passeio aleatório com ou sem tendência temporal (desvio). Mas é possível que a parte não linear do processo cause um desvio em direção a uma solução estacionária. Isso significa



que a afirmação de que a estacionariedade da parte linear é suficiente, mas não é necessária no Teorema (2.8) (TRAPLETTI; LEISCH; HORNIK, 2000). Vemos que a estacionariedade do processo depende da parte linear e podemos usar a teoria usual das raízes unitárias e testar a estacionariedade. Se não temos parte linear, o AR-NN( $p$ ) sempre leva a uma representação estacionária.

### 2.4.2 O Teste de Classificação Aumentada de Dickey-Fuller *Rank*

Na literatura de Box e Jenkins (1976), o número de diferenças necessárias para que uma série se torne estacionária é conhecido como ordem de integração da série. Os testes de raízes unitárias são capazes, em geral, de detectar se a série foi suficientemente diferenciada, para se tornar estacionária. Para tanto, testa-se a hipótese nula de que a série não é estacionária, ou seja, possui raiz unitária, contra a alternativa de que a série é estacionária.

Testes de raízes unitárias possuem um papel muito importante na análise de séries temporais, mas o desempenho desses testes depende de um conjunto de pressupostos que muitas vezes são questionáveis em aplicações reais. A suposição de que o processo gerado é linear parece ser bastante restrito em algumas circunstâncias. Muitas vezes, a série é transformada em logaritmo antes do teste de raiz unitária ser aplicado. Uma alternativa de trabalhar com séries que não sejam lineares, é considerar versões robustas dos testes de raízes unitárias, tais como testes baseados nos postos (*ranks*) das observações.

Nesse contexto, Granger e Hallman (1991) propuseram o teste de Dickey-Fuller *rank* (RADF), onde as observações da série são substituídas pelos seus respectivos *ranks*. Com isso, as hipóteses do teste RADF é definida como

$$\begin{cases} H_0 : \alpha^{(r)} = 1, & \text{Tem raiz unitária;} \\ H_1 : \alpha^{(r)} < 1, & \text{Não tem raiz unitária} \end{cases}$$

(GRANGER; HALLMAN, 1991).

Seja o *rank* da série definido como

$$R_{n,t} = \text{Rank de } X_t \text{ entre } X_1, X_2, X_3, \dots, X_n,$$

e o estimador de mínimos quadrados para  $\alpha$  como

$$\hat{\alpha}^{(r)} = \frac{\sum_{t=2}^n R_{n,t-1} R_{n,t}}{\sum_{t=2}^n R_{n,t-1}^2}.$$

A estatística proposta por Granger e Hallman (1991) fica definida como

$$\hat{\tau}^{(r)} = \frac{\hat{\alpha}^{(r)} - 1}{s(\hat{\alpha}^{(r)})},$$

onde

$$s(\hat{\alpha}^{(r)}) = \frac{S_{(r)}}{\left(\sum_{t=2}^n R_{n,t-1}^2\right)^{\frac{1}{2}}} \quad \text{e} \quad S_{(r)}^2 = \frac{1}{n-2} \sum_{t=2}^n (R_{n,t} - \hat{\alpha}^{(r)} R_{n,t-1})^2,$$

em que  $s(\hat{\alpha}^{(r)})$  é o erro padrão de  $\hat{\alpha}^{(r)}$  e  $S_{(r)}^2$  é o estimador de  $\sigma^2$ .

Neste capítulo nos familiarizamos com os modelos autorregressivos de redes neurais, começando com definições sobre processos estocásticos, redes neurais e redes neurais autorregressivas. A partir de agora devemos entender como funciona a estimação dos parâmetros dos modelos autorregressivos de redes neurais, o próximo capítulo mostra como funciona toda modelagem univariada dos modelos autorregressivos de redes neurais.

### 3 Modelagem Univariada dos Modelos Autor-regressivos de Redes Neurais AR-NN( $p$ )

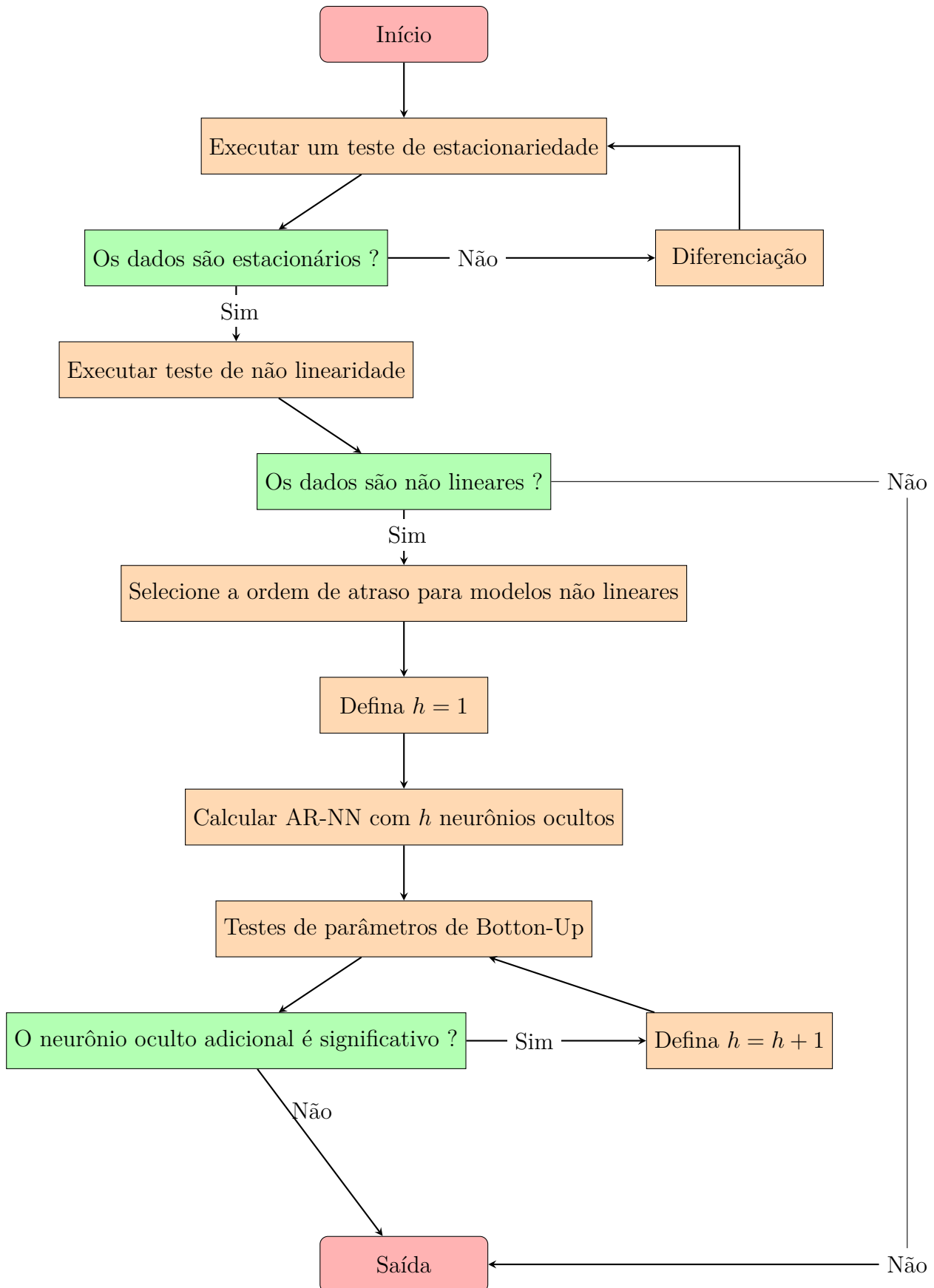
Neste capítulo, iremos definir um modelo AR-NN( $p$ ) univariado para uma determinada série temporal. Apenas estimar os parâmetros certamente não é suficiente para receber um modelo apropriado. Seguimos [Box e Jenkins \(1976\)](#), que propõem proceder em três etapas: Seleção de variáveis, estimação de parâmetros e validação do modelo (testes de parâmetros).

Antes de começarmos com o primeiro passo, é necessário garantir que os dados sejam estacionários. Se não são estacionários, a estacionarização é necessária. Embora esse problema seja analisado apenas para séries temporais lineares, vários autores afirmam que também é relevante para as redes neurais ([LEE; WHITE; GRANGER, 1993](#));([ANDERS, 1997](#));([TRAPLETTI; LEISCH; HORNIK, 2000](#)).

[Steurer \(1996\)](#) mostra por investigação empírica que as redes neurais funcionam com precisão para dados estacionários. Portanto, os métodos apresentados abaixo são aplicáveis apenas em séries temporais estacionárias.

Antes de ajustar um modelo AR-NN( $p$ ), é necessário usar o teste de não linearidade para série temporal por dois motivos: Primeiro, o esforço adicional necessário para ajustar um modelo não linear comparado a um modelo linear. Em segundo lugar, como sabemos na seção anterior, um modelo AR-NN( $p$ ) tem desempenho aproximado ou igual a um modelo linear se a série temporal investigada for determinada por um processo linear.

Na Figura (11), um fluxograma mostra as etapas para construir um modelo AR-NN( $p$ ) para uma determinada série temporal. Esta figura pode servir como um plano geral para construir um modelo AR-NN( $p$ ) de qualquer série temporal, pois o algoritmo construído faz todos os passos necessários, aumentando o número de neurônios ocultos passo a passo, começando com  $h = 1$ .

Figura 11 – Fluxograma da construção de modelo AR-NN ( $p$ ).

Fonte: Autor.

## 3.1 Teste de não Linearidade

Nas seções anteriores, nos familiarizamos com a estrutura dos modelos AR-NN( $p$ ). Se uma série é linear, o esforço adicional de usar um AR-NN( $p$ ) é desnecessário. Para evitar isso, a série deve ser testada para saber se existe uma não linearidade. O teste de não linearidade de [White \(1989\)](#) descrito nesta seção é um método simples e eficiente. Como ainda não especificamos o número de defasagens, o teste de não linearidade oculta deve ser executado em todas as defasagens, começando de 1 a um número máximo pré-especificado de defasagens. A aplicação empírica mostra que uma série temporal pode ser não linear para uma ordem de atraso e linear para outras ordens.

### 3.1.1 O teste de White

A não linearidade negligenciada no sentido de [White \(1989\)](#) significa que existe alguma não linearidade que não é coberta pelo processo AR linear. Se a não linearidade negligenciada existe, e o processo é determinado por uma função não linear, o modelo linear AR é mal especificado. Então, o teste deve examinar se o modelo linear está mal especificado ou não. Portanto, a hipótese nula no teste é que um modelo linear estimado é capaz de explicar a verdadeira função  $F(\mathbf{x}_{t-p})$ . Formalmente

$$H_0 : Prob(F(\mathbf{x}_{t-p}) = \alpha_0 + A^\top \mathbf{x}_{t-p}) = 1, \quad (3.1)$$

contra a hipótese alternativa de que o AR linear não explica a  $F(\mathbf{x}_{t-p})$ .

$$H_1 : Prob(F(\mathbf{x}_{t-p}) = \alpha_0 + A^\top \mathbf{x}_{t-p}) < 1. \quad (3.2)$$

O teste de  $H_0$  contra  $H_1$  é construído com base no pressuposto de resíduos gaussianos. Se  $H_0$  não se aplica, ou seja, nem toda função  $F(\mathbf{x}_{t-p})$  é explicada por  $\alpha_0 + A^\top \mathbf{x}_{t-p}$ , alguma não linearidade negligenciada é contida na parte estocástica. Para separar a não linearidade negligenciada da parte estocástica, devemos reescrever a equação do modelo AR linear como

$$X_t = \alpha_0 + A^\top \mathbf{x}_{t-p} + u_t, \quad (3.3)$$

em que

$$u_t = [F(\mathbf{x}_{t-p}) - \alpha_0 - A^\top \mathbf{x}_{t-p}] + \varepsilon_t, \quad (3.4)$$

em que  $\varepsilon_t$  é i.i.d.  $N(0, \sigma^2)$ .

O primeiro termo da Equação (3.4) observa a parte  $F(\mathbf{x}_{t-p})$  que não é coberta pelo processo linear, a não linearidade negligenciada. Se os resíduos  $u_t$  não são iguais a

parte estocástica  $\varepsilon_t$ , os mesmos contêm a não linearidade negligenciada. Se  $H_0$  se aplica, o primeiro termo da Equação (3.4) desaparece e o termo residual  $u_t$  consiste apenas pelos  $\varepsilon_t$ .

Se  $H_0$  for verdadeira, não há correlação entre o termo residual e o  $\mathbf{x}_{t-p}$ , o que significa que a esperança condicional  $E(u_t|\mathbf{x}_{t-p}) = E(\varepsilon_t|\mathbf{x}_{t-p}) = 0$ . Portanto, mesmo que  $\mathbf{x}_{t-p}$  seja transformado por qualquer função  $\mathcal{H}(\mathbf{x}_{t-p})$ , o termo residual  $u_t$  não está correlacionado com essa transformação, porque

$$E(\mathcal{H}(\mathbf{x}_{t-p})u_t) = E(E(\mathcal{H}(\mathbf{x}_{t-p})u_t|\mathbf{x}_{t-p})) = E(\mathcal{H}(\mathbf{x}_{t-p})) E(u_t|\mathbf{x}_{t-p}) = 0.$$

Definindo  $\mathcal{H}(\mathbf{x}_{t-p})$  como um neurônio oculto adicional, o mesmo pode ser construído usando uma função de ativação  $\Psi(\cdot)$ , com pesos aleatórios  $\gamma_0, \gamma_1, \dots, \gamma_n$ . No artigo original de [White \(1989\)](#), ele mostra que mais de uma unidade oculta adicional pode ser usada. Para manter o teste gerenciável, [Dietz \(2011\)](#) usou apenas uma unidade oculta. A equação do modelo AR-NN( $p$ ) com um neurônio oculto é definida como

$$\mathbf{x}_t = \alpha_0 + A^\top \mathbf{x}_{t-p} + \Psi(\gamma_0 + \Gamma^\top \mathbf{x}_{t-p})\beta + \varepsilon_t. \quad (3.5)$$

O teste de [White \(1989\)](#) é baseado apenas nos pesos  $\gamma_i$ . Nesse sentido, uma consequência da hipótese nula  $H_0$  (3.1) é dado por

$$H_0^* : E(\Psi(\gamma_0 + \Gamma^\top \mathbf{x}_{t-p}) \cdot u_t | \gamma_i) = 0 \quad \forall i = 0, \dots, n.$$

contra a hipótese alternativa

$$H_1^* : E(\Psi(\gamma_0 + \Gamma^\top \mathbf{x}_{t-p}) \cdot u_t | \gamma_i) \neq 0 \quad \forall i = 0, \dots, n.$$

Assim, a rejeição de  $H_0^*$  significa rejeitar  $H_0$ . No entanto, não rejeitar  $H_0^*$  não significa não rejeitar  $H_0$ . Consequentemente, testar  $H_0^*$  contra  $H_1^*$  não é consistente para testar  $H_0$  contra  $H_1$ .

Para chegar à estatística do teste, primeiro a esperança  $E(\Psi(\gamma_0 + \Gamma^\top \mathbf{x}_{t-p}) \cdot u_t | \gamma_i)$  deve ser estimada. A mesma é calculada da forma

$$E(\Psi(\gamma_0 + \Gamma^\top \mathbf{x}_{t-p}) \cdot u_t | \gamma_i) = \frac{1}{T} \sum_{t=1}^T \Psi(\gamma_0 + \Gamma^\top \mathbf{x}_{t-p}) \cdot u_t. \quad (3.6)$$

Se a hipótese nula não for rejeitada, o valor da Equação (3.6) deve estar em torno de zero. Por outro lado, se a hipótese nula é rejeitada, o valor da Equação (3.6) fica distante de zero ([WHITE, 1989](#)).

Um procedimento equivalente, que leva assintoticamente à estatística de teste é definida por uma regressão linear artificial

$$u_t = \phi_1 \nabla(\alpha_0 + A^\top \mathbf{x}_{t-p}) + \phi_2(\Psi(\gamma_0 + \Gamma^\top \mathbf{x}_{t-p})) + u_t^*, \quad (3.7)$$

em que  $\phi_1$  e  $\phi_2$  são parâmetros com dimensões  $(1 \times (1 + p))$  e  $(1 \times 1)$ , respectivamente,  $u_t$  são os resíduos da Equação (3.4),  $u_t^*$  é o termo residual da regressão linear artificial e  $\nabla$  denota o vetor de derivadas parciais (gradiente) das entradas (constante e defasagens de  $\mathbf{x}_t$ ). Para um melhor entendimento das seguintes estatísticas de teste multiplicador de Lagrange (LM), ver [Anders \(1997\)](#). Usando os resíduos  $u_t$  da Equação (3.4), podemos calcular a primeira estatística do teste multiplicador de Lagrange (LM)

$$T_{LM1} = T \cdot \frac{\sum_{t=1}^T \hat{u}_t}{\sum_{t=1}^T u_t^2}, \quad (3.8)$$

onde a mesma tem distribuição  $\chi^2$  com 1 grau de liberdade. Esta estatística de teste é o coeficiente de determinação não centrado da regressão linear artificial (3.7) multiplicada por  $T$ . De acordo com [Davidson e MacKinnon \(1993\)](#), a Equação (3.8) pode ser estabilizada pela multiplicação de um fator  $(T - r)$  com  $r$  igual ao número de parâmetros na Equação (3.7). Uma outra alternativa é a estatística definida como

$$T_{LM2} = \frac{\sum_{t=1}^T u_t^2 - \sum_{t=1}^T (u_t^*)^2}{\sum_{t=1}^T (u_t^*)^2 / (T - r)}. \quad (3.9)$$

A mesma tem distribuição  $F$  com  $(p + 1)$  e  $(T - r)$  graus de liberdade ([WHITE; GALLANT, 1992](#)).

## 3.2 Seleção de Variáveis

Agora, para uma série temporal com não linearidade oculta em pelo menos algum atraso, um AR-NN( $p$ ) pode ser ajustado. Além de estimar os parâmetros, ainda há duas coisas a decidir: Selecionar as defasagens e detectar o número de unidades ocultas [Medeiros, Teräsvirta e Rech \(2006\)](#). Para manter o esforço computacional simples, o primeiro problema é resolvido executando cada uma das defasagens, começando de 1 a um número máximo pré-especificado. O segundo problema é resolvido pela estratégia *Bottom-Up*, começando com um AR-NN( $p$ ) com  $h = 1$  e aumentando  $h$  gradativamente. Em geral, o procedimento de seleção de defasagens e parâmetros é realizado de acordo com o princípio de preferir o modelo mais simples de um conjunto de modelos com o mesmo desempenho.

Na análise de séries temporais lineares, a ordem de atraso é geralmente detectada calculando-se o critério de informação (IC) para vários atrasos, e escolhendo a ordem de atraso correspondente ao menor IC.

O IC mais comum é o critério de informação de Akaike (AIC), que é definido como

$$AIC = T \log(\sigma^2) + 2r$$

([BURNHAM; ANDERSON, 2004](#));([AKAIKE, 1974](#)).

Uma alternativa bem conhecida é o critério de informação Schwarz-Bayesiano (BIC), proposto por [Schwarz \(1978\)](#) e definido como

$$BIC = T \log(\sigma^2) + T \log(r).$$

Nas seções seguintes, discutimos alguns procedimentos de seleção de defasagens, que têm a propriedade comum de não se restringir apenas às redes neurais, mas ser aplicável a todos os tipos de processos não lineares, pois eles usam métodos não paramétricos (polinômios de Taylor e regressão de *kernel*), métodos para aproximar a função não linear desconhecida. Eles são capazes de fornecer uma visão aproximada de como os modelos não lineares se comportariam e qual estrutura de defasagem pode ser ideal para eles. Deve-se mencionar que todos os procedimentos mostrados a seguir têm a limitação de funcionar apenas se os dados estiverem estacionários.

### 3.2.1 O Coeficiente de Autocorrelação

O procedimento mais simples do ponto de vista computacional é baseado em coeficientes de autocorrelação (AC). Essa medida não se restringe a séries temporais lineares, ela também pode ser aplicada a séries não lineares ([FARAWAY; CHATFIELD, 1998](#)).

O coeficiente de correlação de Pearson entre a série original  $X_t$  e um atraso arbitrário  $X_{t-i}$ , em geral é definido como

$$AC_i = AC(X_t, X_{t-i}) = \frac{\text{Cov}(X_t, X_{t-i})}{\sigma_{X_t} \sigma_{X_{t-i}}}.$$

[Evans \(2003\)](#) propõe aplicar esta fórmula apenas em séries estacionárias (para evitar regressão espúria), embora seja frequentemente aplicada em séries não estacionárias. Na aplicação prática, particularmente para séries de dados econômicos e financeiros não estacionários do mundo real, [Dietz \(2011\)](#) observou que os dados têm uma estrutura altamente autocorrelacionada, o que significa que o coeficiente de autocorrelação (AC) é significativo para uma ordem de atraso alta. Portanto, o coeficiente de autocorrelação (AC) não é uma boa ferramenta para identificação da ordem de atraso, porque certamente inclui muitos atrasos, se aplicado a uma série não estacionária.

Os coeficientes de autocorrelação parcial (PAC) é uma modificação do coeficiente de autocorrelação (AC). Eles qualificam a correlação parcial entre as variáveis  $X_t$  e  $X_{t-i}$ , enquanto as variáveis entre elas são mantidas constantes ([METZ, 2010](#)).



Em outras palavras, a correlação entre as duas variáveis é corrigida pela influência das variáveis entre elas. O coeficiente de autocorrelação parcial (PAC) para o primeiro atraso é igual ao coeficiente de autocorrelação (AC) do primeiro atraso

$$AC_1 = PAC_1,$$

e os coeficientes de autocorrelação parcial (PAC) para defasagens maiores que 1,  $i > 1$ , são calculados por

$$PAC_i = \frac{AC_i - \sum_{j=1}^{i-1} PAC_{i-j} \cdot AC_{i-j}}{1 - \sum_{j=1}^{i-1} PAC_{i-j} \cdot AC_{i-j}}$$

(EVANS, 2003).

### 3.2.2 Informações Mútuas

A informação mútua (MI) é semelhante aos coeficientes de autocorrelação (AC), uma medida não paramétrica para a dependência entre duas séries. No caso da análise de séries temporais, as duas séries são a série original  $X_t$  e uma série retardada arbitrária  $X_{t-i}$ . Segundo Hausser e Strimmer (2009), existe uma relação entre os coeficientes de autocorrelação ( $AC^*$ ) e a informação mútua (MI). Observe que, neste caso, o  $AC^*$  não significa o coeficiente de correlação de Pearson da subseção (3.2.1), mas uma versão geral do AC que também é responsável principalmente pela não linearidade. O coeficiente de correlação de Pearson é considerado apenas um bom estimador do verdadeiro AC. Essa relação é definida como

$$MI_i = MI(X_t, X_{t-i}) = -\frac{1}{2} \log(1 - AC_i^{*2}).$$

Com isso, temos algumas propriedades do MI. O intervalo do MI é  $\mathbb{R}^+$ . O mesmo é simétrico,  $(MI(X_t; X_{t-i}) = MI(X_{t-i}; X_t))$ , e se a variável  $X_t$  e seu atraso  $X_{t-i}$  forem independentes, ele é zero (HAUSSER; STRIMMER, 2009).

Granger e Lin (1994) usaram essa relação entre o AC e o MI para formular o coeficiente de informação mútua, (MIC).

$$MIC_i = MIC(X_t, X_{t-i}) = |AC^*| = \sqrt{1 - \exp^{-2MI_i}}.$$

onde o intervalo do MIC é entre 0 e 1.

O coeficiente de informação mútua (MIC) de Granger e Lin (1994) é consequentemente um estimador alternativo para o valor absoluto do coeficiente de autocorrelação. Granger e Lin (1994) mostram, por simulação, que o coeficiente de informação mútua (MIC) faz um trabalho melhor na identificação da verdadeira ordem de atraso do que o coeficiente de correlação de Pearson.

### 3.3 Estimação dos parâmetros

O passo mais importante para concretizar o AR-NN( $p$ ), é a estimação dos pesos. Isso equivale à estimação dos parâmetros na análise de séries temporais lineares e é chamado de aprendizado ou treinamento na teoria das redes neurais. Existem muitos procedimentos para estimar os parâmetros das redes neurais. Assim, é preciso distinguir entre métodos de aprendizado supervisionado e métodos não supervisionados (HAYKIN, 2009).

Métodos supervisionados significam que a estimativa de saída é comparada com a saída desejada. Os métodos não supervisionados não usam critérios para controlar o processo de aprendizagem, portanto, eles não são aplicáveis às estatísticas e, principalmente, à análise de séries temporais. A seguir, nos concentramos apenas nos procedimentos de aprendizado supervisionado.

Em geral, pode-se dizer que existem duas classes diferentes de procedimentos de aprendizado supervisionado com relação à estimativa dos parâmetros. O aprendizado em lotes e o aprendizado *online* (WIDMANN, 2001);(HAYKIN, 2009).

O aprendizado em lotes é um procedimento iterativo em que os pesos são ajustados em cada iteração após a apresentação de todas as entradas,  $T$ . Enquanto o aprendizado *online*, às vezes chamado de aprendizado estocástico, os pesos são ajustados elemento a elemento. Isso significa que para cada conjunto de neurônios de entrada e saída, de 1 a  $T$ , os pesos são ajustados novamente. O processo AR-NN( $p$ ) é estimado para as entradas e saídas em um determinado momento  $t$  apenas, para o tempo  $t + 1$  os pesos são ajustados novamente. As principais vantagens do método de aprendizado *online* sobre os de lotes, são a menor complexidade computacional e a melhor adaptabilidade para integrar novos valores se os dados chegarem sequencialmente.

Alguns estudos mostraram que, sob certas condições, os procedimentos de aprendizado em lotes e *online* apresentam resultados semelhantes, principalmente se o conjunto de dados de entrada for grande (OPPER; WINTHER, 1999).

Em geral, pode-se dizer que o aprendizado *online* é mais rápido e menos complexo do que o aprendizado em lotes, mas no que diz respeito à precisão dos resultados, o desempenho é ruim. O aprendizado *online* pode ser mais útil em outras áreas, onde a complexidade é muito mais importante do que as estatísticas (BOTTOU, 2004).

Portanto, prosseguimos apenas com os procedimentos de aprendizagem em lotes, apesar de nos últimos anos, vários procedimentos terem sido feitos com relação à execução de procedimentos de aprendizagem *online*, no reconhecimento de padrões (SCHRAUDOLPH, 2002).

Fazemos isso porque nas estatísticas geralmente os conjuntos de dados são entregues completamente e contêm todas as informações necessárias para analisar as interdependên-

cias de longo prazo. Os procedimentos de aprendizado *online* seriam úteis se os dados de entrada fossem fornecidos continuamente, mesmo durante o processo de aprendizado e, portanto, o ajuste dos pesos seria mais preciso.

### 3.3.1 Função Desempenho

A qualidade do ajuste de um modelo AR-NN( $p$ ) pode ser determinada por

$$Q(\Theta) = \frac{1}{2} \sum_{t=1}^T (X_t - G(\Theta, \mathbf{x}_{t-p}))^2. \quad (3.10)$$

Se essa função de desempenho for usada, os procedimentos de estimação de parâmetros a seguir são referidos na literatura como método de mínimos quadrados não lineares (NLS) e não estão restritos apenas às redes neurais. Obviamente, é possível usar outras funções de desempenho, como a função de probabilidade, mas elas são menos comuns (ANDERS, 1997).

Como a função AR-NN( $p$ ) também deve ser válida para valores futuros da série temporal, a esperança da função (3.10) deve ser minimizada. Como um AR em geral é um processo estocástico, existe alguma incerteza, como sabemos da Definição (2.5).

Assumimos que a incerteza é determinada apenas pela parte estocástica  $\varepsilon_t$ . Essa incerteza deve ser minimizada, devido às nossas suposições sobre  $\varepsilon_t$ .

A relação entre a função de desempenho e a variação de  $\varepsilon_t$ , bem como o fato de que a minimização de um é igual à minimização do outro, é formalmente mostrada a seguir. As incertezas também causam problemas à propriedade estocástica da esperança, razão pela qual não se pode calcular diretamente  $\Theta$ , mas sim estimá-lo. Um estimador não linear ótimo de mínimos quadrados para  $\Theta$ , pode ser encontrado pela solução

$$\hat{\Theta} = \arg \min_{\Theta \in \Theta} E(Q(\Theta)),$$

em que  $\Theta$  denota o espaço de pesos da rede.

Pela transformação de  $Q(\Theta)$ , pode-se mostrar que  $\hat{\Theta}$  é um ótimo estimador,  $\hat{\Theta} \approx \Theta$ , e está minimizando o erro esperado entre uma função desconhecida  $F(\mathbf{x}_{t-p})$  e sua

aproximação  $G(\Theta, \mathbf{x}_{t-p})$  (WHITE, 1988).

$$\begin{aligned}
 E(Q(\Theta)) &= \frac{1}{2} E \left[ \sum_{t=1}^T (X_t - G(\Theta, \mathbf{x}_{t-p}))^2 \right] \\
 &= \frac{1}{2} E \left[ \sum_{t=1}^T (X_t - F(\mathbf{x}_{t-p}) + F(\mathbf{x}_{t-p}) - G(\Theta, \mathbf{x}_{t-p}))^2 \right] \\
 &= \frac{1}{2} E \left[ \sum_{t=1}^T (X_t - F(\mathbf{x}_{t-p}))^2 \right] + \\
 &\quad E \left[ \sum_{t=1}^T (X_t - F(\mathbf{x}_{t-p}))(F(\mathbf{x}_{t-p}) - G(\Theta, \mathbf{x}_{t-p})) \right] + \\
 &\quad \frac{1}{2} E \left[ \sum_{t=1}^T (F(\mathbf{x}_{t-p}) - G(\Theta, \mathbf{x}_{t-p}))^2 \right] \tag{3.11}
 \end{aligned}$$

$$\begin{aligned}
 &= \frac{1}{2} E \left[ \sum_{t=1}^T (X_t - F(\mathbf{x}_{t-p}))^2 \right] + \\
 &\quad \frac{1}{2} E \left[ \sum_{t=1}^T (F(\mathbf{x}_{t-p}) - G(\Theta, \mathbf{x}_{t-p}))^2 \right] \tag{3.12}
 \end{aligned}$$

$$= \frac{1}{2} E \left[ \sum_{t=1}^T \varepsilon_t^2 \right] + \frac{1}{2} E \left[ \sum_{t=1}^T (F(\mathbf{x}_{t-p}) - G(\Theta, \mathbf{x}_{t-p}))^2 \right] \tag{3.13}$$

(3.12) segue de (3.11) porque

$$\begin{aligned}
 &E \left[ \sum_{t=1}^T (X_t - F(\mathbf{x}_{t-p}))(F(\mathbf{x}_{t-p}) - G(\Theta, \mathbf{x}_{t-p})) \right] \\
 &= E \left[ \sum_{t=1}^T (X_t - F(\mathbf{x}_{t-p}))(F(\mathbf{x}_{t-p}))\varepsilon_t \right] \\
 &= E \left[ \sum_{t=1}^T (X_t - F(\mathbf{x}_{t-p}))(F(\mathbf{x}_{t-p}))\varepsilon_t | \mathbf{x}_{t-p} \right] \\
 &= E \left[ \sum_{t=1}^T (X_t - F(\mathbf{x}_{t-p}))(F(\mathbf{x}_{t-p})) E(\varepsilon_t | \mathbf{x}_{t-p}) \right] \\
 &= 0.
 \end{aligned}$$

Note que  $E(\varepsilon_t | \mathbf{x}_{t-p}) = 0 \quad \forall t$ . O primeiro termo de (3.13) afirma que  $\Theta$  minimiza os erros da parte estocástica. O segundo termo afirma que um mínimo de  $Q(\Theta)$  é atingido se  $G(\Theta, \mathbf{x}_{t-p}) = F(\mathbf{x}_{t-p})$ . Nesse caso (3.13) reduz-se a (com relação à suposição i.i.d)

$$Q(\Theta) = \frac{1}{2} \sum_{t=1}^T \varepsilon_t^2.$$

Por outro lado,

$$\sum_{t=1}^T \varepsilon_t = \sum_{t=1}^T (X_t - G(\Theta, \mathbf{x}_{t-p})),$$

e portanto

$$\varepsilon_t = \varepsilon_t(\Theta) = (X_t - G(\Theta, \mathbf{x}_{t-p})). \quad (3.14)$$

Na Equação (3.14), o residual no tempo  $t$  é descrito por uma função de  $\Theta$ . Esta função bem como a função de desempenho são importantes para as próximas seções.

### 3.3.2 Termos Matriciais Importantes

Nas seções seguintes, usaremos extensivamente o vetor gradiente, matriz Jacobiana e matriz hessiana. Portanto, eles devem ser introduzidos e explicados aqui.

A construção mais simples é o vetor gradiente, indicado por  $\nabla(\cdot)$ . O vetor gradiente é um vetor de coluna das derivadas parciais de uma função, em relação as suas respectivas variáveis. Considere, por exemplo, o vetor gradiente da função desempenho

$$\nabla Q(\Theta) = \begin{pmatrix} \frac{\partial Q(\Theta)}{\partial \theta_1} \\ \vdots \\ \frac{\partial Q(\Theta)}{\partial \theta_r} \end{pmatrix},$$

em que sua dimensão é  $(r \times 1)$ .

Se uma função depende adicionalmente do tempo  $t$ , por exemplo, o residual no tempo  $t$  na Equação (3.14), a matriz Jacobiana corresponde em certo sentido ao vetor gradiente, em que a matriz das derivadas parciais são, a variável (nas linhas) e o tempo (nas colunas). Denotamos isso por  $J(\cdot)$ . Para a equação (3.14), a matriz Jacobiana tem dimensão  $(T \times r)$  e é calculada por

$$J(\varepsilon_t(\Theta)) = \begin{pmatrix} \frac{\partial \varepsilon_1}{\partial \theta_1} & \frac{\partial \varepsilon_1}{\partial \theta_2} & \cdots & \frac{\partial \varepsilon_1}{\partial \theta_n} \\ \frac{\partial \varepsilon_2}{\partial \theta_1} & \frac{\partial \varepsilon_2}{\partial \theta_2} & \cdots & \frac{\partial \varepsilon_2}{\partial \theta_n} \\ \vdots & \vdots & \cdots & \vdots \\ \frac{\partial \varepsilon_T}{\partial \theta_1} & \frac{\partial \varepsilon_T}{\partial \theta_2} & \cdots & \frac{\partial \varepsilon_T}{\partial \theta_n} \end{pmatrix}.$$

Existe uma relação entre o vetor gradiente da função desempenho e a matriz Jacobiana que pode ser construída usando um vetor residual  $\boldsymbol{\varepsilon} = (\varepsilon_1, \varepsilon_2, \dots, \varepsilon_T)$ . A matriz

tem dimensão  $(r \times T)$  e é calculada por

$$\nabla Q(\Theta) = J(\varepsilon_t(\Theta))^T \varepsilon = \begin{pmatrix} \varepsilon_1 \frac{\partial \varepsilon_1}{\partial \theta_1} + \varepsilon_2 \frac{\partial \varepsilon_1}{\partial \theta_2} + \cdots + \varepsilon_T \frac{\partial \varepsilon_1}{\partial \theta_r} \\ \varepsilon_1 \frac{\partial \varepsilon_2}{\partial \theta_1} + \varepsilon_2 \frac{\partial \varepsilon_2}{\partial \theta_2} + \cdots + \varepsilon_T \frac{\partial \varepsilon_2}{\partial \theta_r} \\ \vdots \\ \varepsilon_1 \frac{\partial \varepsilon_T}{\partial \theta_1} + \varepsilon_2 \frac{\partial \varepsilon_T}{\partial \theta_2} + \cdots + \varepsilon_T \frac{\partial \varepsilon_T}{\partial \theta_r} \end{pmatrix}.$$

Observe que esse relacionamento é particularmente  $\forall i = 1, \dots, r$  porque

$$\frac{\partial Q(\Theta)}{\partial \theta_i} = \nabla \frac{1}{2} \left( \sum_{t=1}^T \varepsilon_t^2 \right) = 2 \cdot \frac{1}{2} \left( \sum_{t=1}^T \varepsilon_t \right) \frac{\partial \left( \sum_{t=1}^T \varepsilon_t \right)}{\partial \theta_i}.$$

Outra matriz importante, é a matriz hessiana. É a matriz de segunda ordem das derivadas de uma função com suas respectivas variáveis, e ela é denotada por  $\nabla^2(\cdot)$ .

Para a função de desempenho  $Q(\Theta)$ , a matriz hessiana tem dimensão  $(r \times r)$  e é representada por

$$\nabla^2 Q(\Theta) = \begin{pmatrix} \frac{\partial^2 Q(\Theta)}{\partial \theta_1 \partial \theta_1} & \frac{\partial^2 Q(\Theta)}{\partial \theta_1 \partial \theta_2} & \cdots & \frac{\partial^2 Q(\Theta)}{\partial \theta_1 \partial \theta_r} \\ \frac{\partial^2 Q(\Theta)}{\partial \theta_2 \partial \theta_1} & \frac{\partial^2 Q(\Theta)}{\partial \theta_2 \partial \theta_2} & \cdots & \frac{\partial^2 Q(\Theta)}{\partial \theta_2 \partial \theta_r} \\ \vdots & \vdots & \cdots & \vdots \\ \frac{\partial^2 Q(\Theta)}{\partial \theta_r \partial \theta_1} & \frac{\partial^2 Q(\Theta)}{\partial \theta_r \partial \theta_2} & \cdots & \frac{\partial^2 Q(\Theta)}{\partial \theta_r \partial \theta_r} \end{pmatrix}.$$

### 3.3.3 Características Básicas dos Algoritmos

Todos os algoritmos de estimação numérica de parâmetros para redes neurais funcionam da mesma maneira: Começando com um vetor de parâmetro inicial aleatório  $\Theta^0$ , e é pesquisado iterativamente até o vetor de parâmetro ideal.

O vetor de parâmetro ideal é alcançado se a função de desempenho for minimizada. A função possui vários extremos, mínimos e máximos, que satisfazem  $\nabla Q(\Theta) = 0$ , porque o gradiente  $\nabla Q(\Theta)$  é a inclinação da função (BISHOP, 1995).

No entanto, às vezes, apenas um mínimo local pode ser alcançado pelo algoritmo. Além disso, a escolha do vetor de peso inicial  $\Theta^0$  influencia o resultado do respectivo algoritmo, achando mínimos locais ou globais. Mas, não existem algoritmos alternativos que garantam um mínimo global.

Em geral, o algoritmo é realizado de acordo com a Figura (12). Os pesos são atualizados após cada iteração e a função de desempenho é calculada. Se um critério de parada for atingido, o algoritmo será encerrado. O critério de parada pode ser uma restrição referente à função desempenho, por exemplo, a distância entre a função desempenho em duas iterações, que deve estar abaixo de um determinado valor (ANDERS, 1997). Para nossa tentativa, esse critério de parada eventualmente contorna a detecção de um mínimo melhor, porque o algoritmo é parado imediatamente após o critério ser alcançado. Portanto, guardamos o resultado da função de desempenho e o vetor de parâmetro após cada iteração. Em seguida, o número máximo de iterações,  $i^{\max}$ , pode ser usado como critério de parada, e o vetor de parâmetro ideal é calculado usando as seguintes etapas

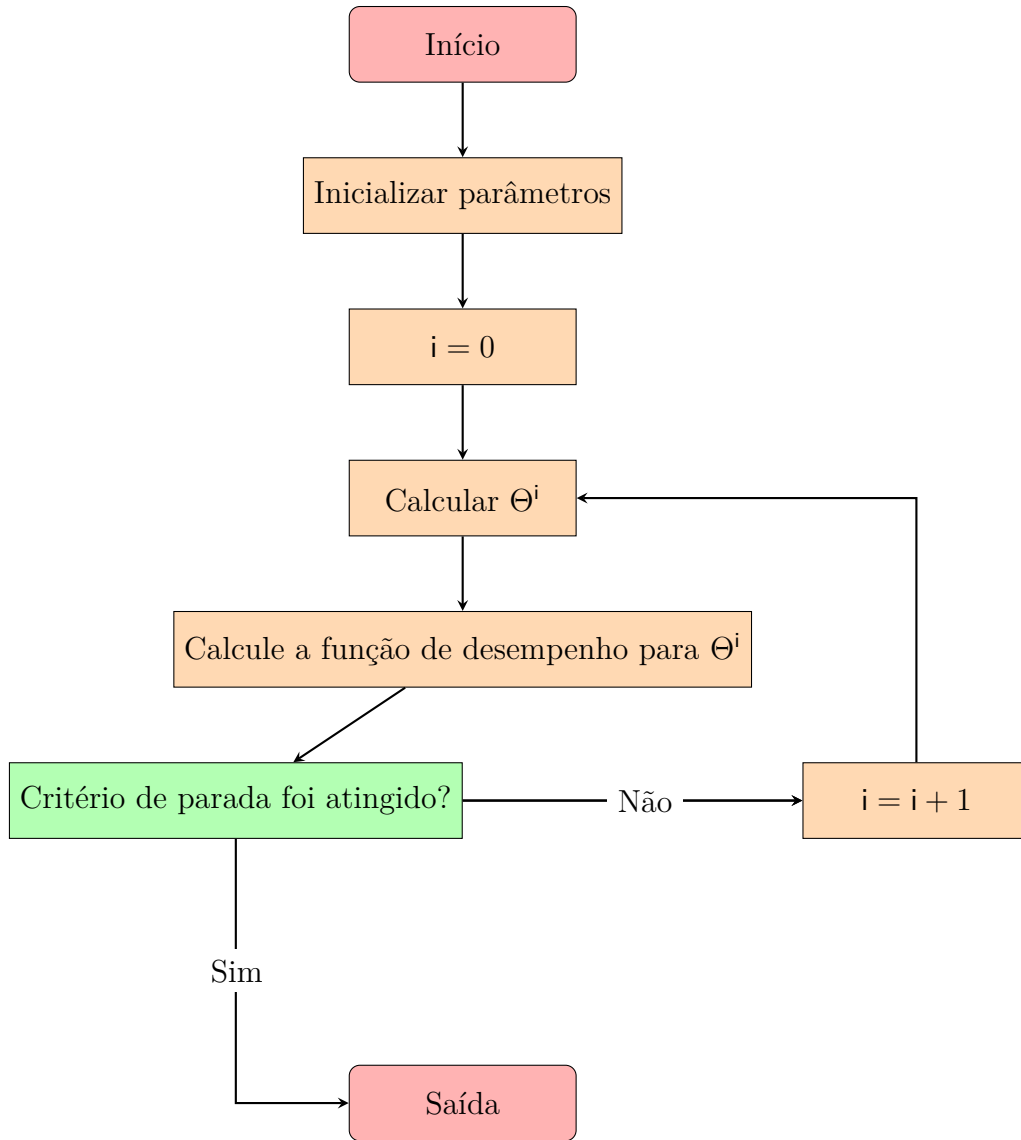
- 1 - Inicie o algoritmo com o vetor de peso inicial  $\Theta^0$ ;
- 2 - Após cada iteração, salve  $Q(\Theta^i)$  e  $\Theta^i$ ;
- 3 - Saia do algoritmo depois de  $i^{\max}$  iterações;
- 4 - Entre e pesquise nos valores salvos;

$$Q(\Theta^{i^*}) = \arg \min_{i \in [0; i^{\max}]} Q(\Theta^i)$$

- 5 -  $\Theta^{i^*}$  é o vetor de parâmetro ideal.

$i^*$  denota a iteração em que o vetor de parâmetro ideal é atingido. Esse procedimento pode ser interpretado como a busca de um mínimo global dentro de um horizonte finito de iterações. Frequentemente, a função desempenho converge para uma determinada constante dentro de um número limitado de iterações. Portanto, um bom mínimo local dentro de um número limitado de iterações,  $i^{\max}$ , geralmente é de fato um mínimo global.

Figura 12 – Estimação interativa dos parâmetros.



Fonte: Autor.

### 3.3.4 Métodos de Descida de Gradiente de Primeira Ordem

Nesta seção, são discutidos os algoritmos de estimativa de parâmetros iterativos mais antigos e computacionalmente mais simples para redes neurais. Eles são baseados nas derivadas parciais de primeira ordem, o vetor gradiente da função desempenho  $\nabla Q(\Theta)$ . Portanto, eles são chamados métodos de descida de gradiente de primeira ordem.

Anderson (1988) propõe calcular as alterações dos pesos proporcionais aos derivativos parciais acumulados. Esse algoritmo de aprendizado também é chamado de algoritmo de descida mais acentuada Bishop (1995). A mudança no peso individual  $\theta_i \in \Theta$  é

$$\Delta\theta_i^{i+1} = v \frac{\partial Q(\Theta^i)}{\partial \theta_i^i}, \quad (3.15)$$



em que o parâmetro  $v \in \mathbb{R}^+$  é chamado de taxa de aprendizado, e  $i$  o número de iterações (RUMELHART; MCCLELLAND; GROUP, 1987).

Começando com um  $\Theta^0$  inicial arbitrário, os pesos são atualizados após cada iteração. A Equação (3.15) pode ser escrita em representação vetorial (WIDMANN, 2001).

$$\Delta\Theta^{i+1} = -v\nabla Q(\Theta^i), \quad (3.16)$$

em que  $\nabla Q(\Theta^i)$  é o vetor de gradiente de dimensão  $(r \times 1)$ .

O principal problema com o algoritmo de descida é a escolha de uma taxa de aprendizado apropriada. Se for escolhida muito pequena, muitas etapas serão necessárias, porque as alterações após cada iteração são muito pequenas. Se, por outro lado, a taxa de aprendizado escolhida for muito grande, o perigo consiste em negligenciar um mínimo global, porque os resultados podem tender a uma forte oscilação. A variação da taxa de aprendizado é uma solução subjetiva e, portanto, não recomendável (BISHOP, 1995).

Várias extensões do algoritmo de descida foram desenvolvidas para sistematizar o método. A primeira a mencionar aqui, proposta por Rumelhart, McClelland e Group (1987), é incluir um termo de momento na Equação (3.16)

$$\Delta\Theta^{i+1} = -v\nabla Q(\Theta^i) + \alpha\Delta\Theta^i,$$

onde  $\alpha \in [0; 1]$  é o parâmetro de momento. A razão pela qual esse termo de momento é adicionado é que ele pode alterar variações de alta frequência na superfície do erro no espaço de peso. Em outras palavras, o termo momento suaviza as oscilações. O efeito é uma convergência mais rápida do algoritmo, porque pode-se usar um valor maior sem o perigo de perder qualquer mínimo global. (RUMELHART; MCCLELLAND; GROUP, 1987)

De acordo com Bishop (1995), essa extensão realmente não resolve os problemas do algoritmo de descida da Equação (3.16), porque depende de um segundo parâmetro,  $\alpha$ , que deve ser escolhido arbitrariamente como  $v$ .

Outra alternativa é o chamado método de negrito de Vogl et al. (1988), onde a taxa de aprendizado é atualizada de acordo com algumas regras após cada iteração. Por exemplo, considerando a Equação (3.16), se o valor da função de erro atual  $Q(\Theta^i)$  for menor que o valor da função de erro anterior  $Q(\Theta^{i-1})$ , a taxa de aprendizado poderá ser ligeiramente aumentada na próxima iteração.

Existem inúmeras alternativas para solucionar os problemas apresentados pelo método de descida de primeira ordem. Porém, não faz parte do objetivo dessa monografia entrar nesses métodos alternativos, e sim apenas explicar como é o procedimento geral de primeira ordem.

### 3.3.5 Métodos de Descida de Gradiente de Segunda Ordem

Os métodos de descida de gradiente de segunda ordem são algoritmos de aprendizado, que explicitamente fazem uso da matriz hessiana  $\nabla^2 Q(\Theta)$ . Considerando a Equação (3.15). A taxa de aprendizado é substituída pela matriz hessiana inversa, de modo que

$$\Theta^{i+1} = \Theta^i - (\nabla^2 Q(\Theta^i))^{-1} \nabla Q(\Theta^i). \quad (3.17)$$

O termo  $(\nabla^2 Q(\Theta^i))^{-1}$  é chamado de direção de Newton. Sua principal vantagem é que a direção de Newton, ou o passo de Newton, de uma função de erro quadrático aponta diretamente para um mínimo e, portanto, evita oscilações. Contudo, a determinação da matriz hessiana traz alguns problemas. Em primeiro lugar, é muito exigente, do ponto de vista computacional, calcular e inverter a matriz hessiana (BISHOP, 1995). Em segundo lugar, a direção de Newton pode apontar para um ponto máximo ou de sela, o que é o caso se a matriz hessiana não for definida positivamente. Como consequência, o erro não é necessariamente reduzido em cada iteração.

Bishop (1995) reduziu o segundo problema adicionando uma matriz simétrica definida positivamente à matriz hessiana, que inclui a unidade  $I$  e um parâmetro grande e suficiente,  $\lambda$ . Então a combinação

$$\nabla^2 Q(\Theta) + \lambda I, \quad (3.18)$$

é certamente positiva.

O primeiro problema, que é geralmente conhecido como a maior desvantagem do método de Newton e é a origem de vários procedimentos de aproximação, chamados métodos quase-Newton. Como eles não lidam com gradientes de segunda ordem diretamente, mas os aproximam por meio de gradientes de primeira ordem, eles são geralmente classificados como métodos de gradiente de primeira ordem (BISHOP, 1995);(WIDMANN, 2001); (HAYKIN, 2009).

O algoritmo de Levenberg-Marquardt na próxima seção, é mostrado como um tipo poderoso de um método quase-Newton.

### 3.3.6 O Algoritmo Levenberg-Marquardt

O algoritmo Levenberg-Marquardt desenvolvido por Levenberg (1944) e Marquardt (1963), é chamado de método quase-Newton, ele combina o algoritmo de descida de primeira ordem e o método de Newton. As vantagens desse método é que ele converge rapidamente como o método de Newton, mas não pode divergir devido à influência mais acentuada do algoritmo de descida de primeira ordem.

O algoritmo de Levenberg-Marquardt é comumente conhecido como um dos métodos de aprendizado mais poderosos para redes neurais. Segundo Bishop (1995), o algoritmo de

Levenberg-Marquardt é especialmente aplicável às funções de desempenho da soma dos quadrados dos erros.

Podemos mostrar que a matriz hessiana da função desempenho pode ser estimada pelo produto cruzado das matrizes jacobianas de  $\varepsilon_t(\Theta)$ .

$$\begin{aligned}\nabla^2 Q(\Theta) &= \nabla^2 \left( \frac{1}{2} \sum_{t=1}^T \varepsilon_t(\Theta)^2 \right) \\ &= \nabla \left( \nabla \left( \frac{1}{2} \sum_{t=1}^T \varepsilon_t(\Theta)^2 \right) \right) \\ &= \nabla (J(\varepsilon_t(\Theta))^\top E).\end{aligned}$$

Pela regra de diferenciação do produto

$$\frac{\partial \left( \sum_{t=1}^T \varepsilon_t \frac{\partial \varepsilon_t}{\partial \theta_i} \right)}{\partial \theta_j} = \sum_{t=1}^T \left( \frac{\partial \varepsilon_t}{\partial \theta_j} \frac{\partial \varepsilon_t}{\partial \theta_i} + \varepsilon_t \frac{\partial^2 \varepsilon_t}{\partial \theta_i \partial \theta_j} \right), \quad (3.19)$$

o segundo termo à direita da Equação (3.19) é aproximadamente zero

$$\varepsilon_t \frac{\partial^2 \varepsilon_t}{\partial \theta_i \partial \theta_j} = 0$$

(BISHOP, 1995).

Logo, o primeiro termo é igual ao produto cruzado das matrizes jacobianas. Com esse resultado, obtemos a Equação (3.17).

$$\Delta \Theta^{i+1} = - \left[ J(\varepsilon_t(\Theta^i))^\top J(\varepsilon_t(\Theta^i)) \right]^{-1} J(\varepsilon_t(\Theta^i))^\top E^i. \quad (3.20)$$

O produto cruzado puro das matrizes jacobianas às vezes leva à singularidades e isso pode ser problemático. Assim, é necessária uma modificação. A Equação (3.17), juntamente com a Equação (3.18), pode ser reescrita como

$$\Delta \Theta^{i+1} = - \left[ \nabla^2 Q(\Theta^i) + \lambda I \right]^{-1} \nabla Q(\Theta^i). \quad (3.21)$$

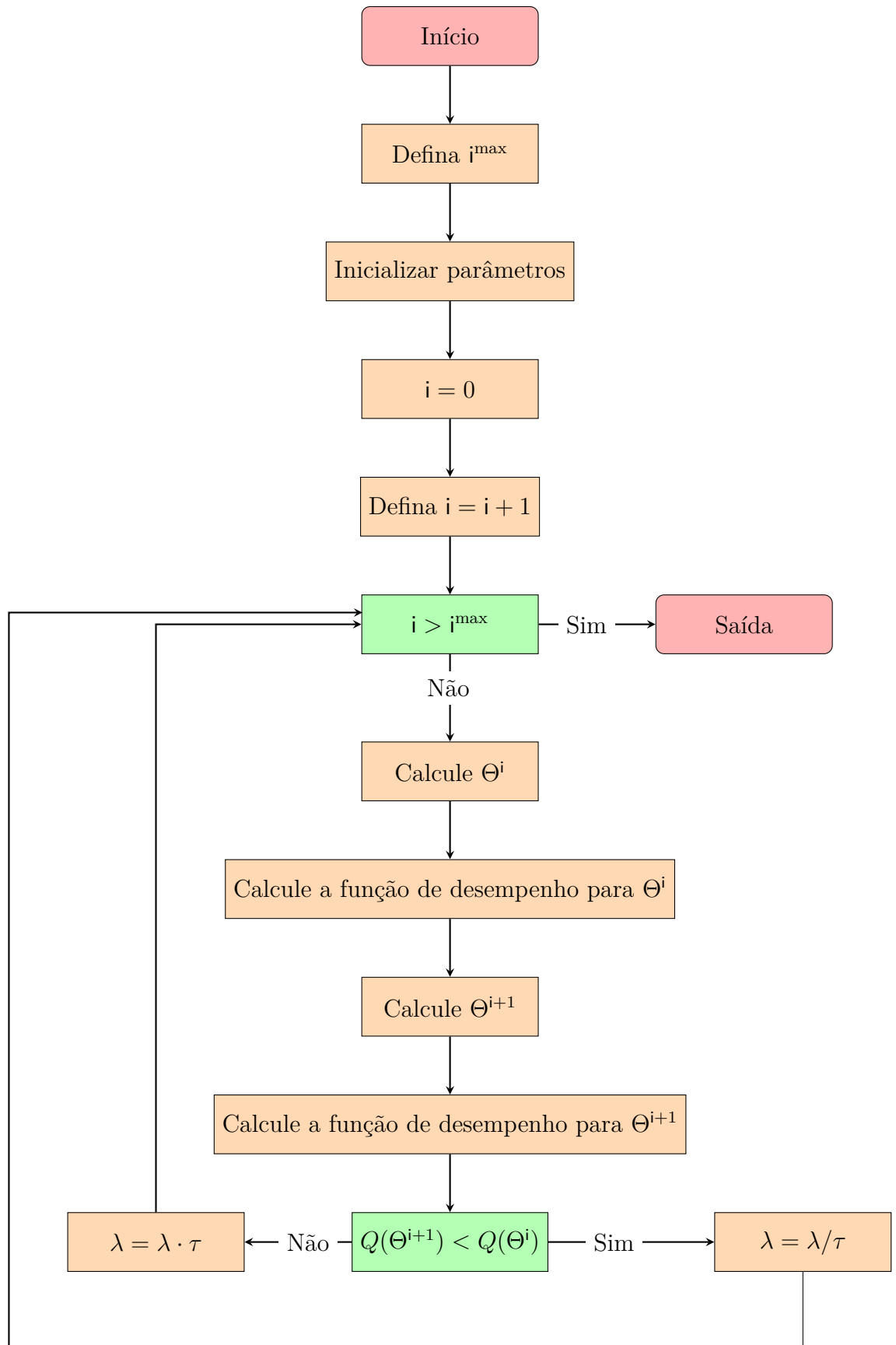
A Equação de Levenberg-Marquardt (3.21) agora contém a aproximação da Equação (3.20) para (3.18) resultando na Equação (3.22)

$$\Delta \Theta^{i+1} = - \left[ J(\varepsilon_t(\Theta^i))^\top J(\varepsilon_t(\Theta^i)) + \lambda I \right]^{-1} J(\varepsilon_t(\Theta^i))^\top E^i, \quad (3.22)$$

em que  $\lambda$  é multiplicado por um fator  $\tau$  se uma iteração resultar em um aumento na função desempenho  $Q(\Theta)$ . Se uma iteração reduzir a função desempenho  $Q(\theta)$ ,  $\lambda$  será dividido por  $\tau$  (HAYKIN, 2009).

A Figura (13) mostra como funciona o algoritmo de Levenberg-Marquardt.

Figura 13 – Fluxograma do algoritmo de Levenberg-Marquardt.



Fonte: Autor.

## 3.4 Testes de Parâmetros

A etapa final é examinar se o modelo estimado é apropriado. Isso é feito pelo teste *Bottom-up*. O mesmo significa começar com o modelo estimado, e examinar se uma unidade oculta adicional melhoraria o modelo.

Os testes de parâmetros são bem conhecidos a partir de estatísticas lineares. Eles consideram um parâmetro ou um conjunto de parâmetros dentro do modelo estimado e os testam quanto à significância

### 3.4.1 Testes de Parâmetro *Bottom-Up*

Antes de começar a explicar o teste, temos que ter uma breve noção da expansão de Taylor.

#### 3.4.1.1 Expansão de Taylor

A expansão de Taylor é um método para aproximar funções não lineares por uma cadeia de polinômios de ordem crescente. Suas principais vantagens são sua propriedade geral de aproximação, e a existência de uma distribuição para seus parâmetros. Baseado em [Siegmund-Schultze \(1988\)](#) e estendido por [Stone \(1948\)](#), o teorema de Stone-Weierstrass diz que um polinômio de Taylor de ordem suficientemente alta pode aproximar-se de qualquer função ([MEDEIROS; TERÄSVIRTA; RECH, 2006](#)). A expansão de Taylor de ordem  $k$  para uma função  $F(x)$  em torno de um ponto  $x_0$  é definido como

$$F(x) = F(x_0) + \frac{F'(x_0)}{1!}(x - x_0) + \dots + \frac{F^k(x_0)}{k!}(x - x_0)^k, \quad (3.23)$$

onde  $F'(x_0)$  é a primeira derivada para  $F(x_0)$  e  $F^k(x_0)$  é a  $k$ -ésima derivada para  $F(x_0)$  ([ANDERS, 1997](#)).

*Observação 3.1.* Se  $x_0 = 0$ , (expansão de Taylor em torno de 0) a série (3.23) também é chamada de série Maclaurin.

Se tentarmos aproximar uma função não linear desconhecida  $F(\mathbf{x}_{t-p})$ , não poderíamos determinar  $F(0)$  bem como suas derivadas. Em geral, as derivadas em relação a zero consistem apenas em partes constantes.

Todas as constantes de uma série de Maclaurin podem ser combinadas em parâmetros para que não seja mais necessário conhecer as derivadas.

Logo, a aproximação polinomial da Equação (2.3) torna-se

$$\begin{aligned}
 X_t = & \alpha_0 + \underbrace{\sum_{j_1=1}^n \alpha_{j_1} X_{t-j_1}}_{\text{Componente Linear}} + \underbrace{\sum_{j_1=1}^n \sum_{j_2=j_1}^n \alpha_{j_1, j_2} X_{t-j_1} X_{t-j_2}}_{\text{Componente Quadrática}} \\
 & + \underbrace{\sum_{j_1=1}^n \sum_{j_2=j_1}^n \sum_{j_3=j_2}^n \alpha_{j_1, j_2, j_3} X_{t-j_1} X_{t-j_2} X_{t-j_3} + \dots}_{\text{Componente Cúbica}}, \\
 & + \underbrace{\sum_{j_1=1}^n \dots \sum_{j_k=j_{k-1}}^n \alpha_{j_1, j_2, \dots, j_k} X_{t-j_1} \dots X_{t-j_k}}_{\text{Componente de grau k}} + u_t
 \end{aligned}$$

em que  $u_t$  é a parte residual que consiste  $\varepsilon_t$  mais o erro adicional causado pela aproximação.

### 3.4.1.2 Bottom-Up

O teste de proposto por Lee, White e Granger (1993), também pode ser usado como um teste de não linearidade oculta adicional. A Equação (3.3) do teste de White será substituído por um AR-NN  $G(\Theta; \mathbf{x}_{t-p})$ . Logo, sem perda de generalidade, o mesmo fica definido como

$$X_t = G(\Theta; \mathbf{x}_{t-p}) + u_t,$$

com

$$u_t = (F(\mathbf{x}_{t-p}) - G(\Theta, \mathbf{x}_{t-p})) + \varepsilon_t. \quad (3.24)$$

Se o primeiro termo na Equação (3.24) for zero, o valor estimado de AR-NN  $G(\Theta; \mathbf{x}_{t-p})$  explica completamente o processo e não há não-linearidade oculta adicional. Para testar isso como na Equação (3.5), um neurônio oculto adicional é adicionado à equação AR-NN e testado

$$X_t = G(\Theta; \mathbf{x}_{t-p}) + \Psi(\gamma_{0a} + \Gamma_a^\top \mathbf{x}_{t-p})\beta_a + \varepsilon_t,$$

O índice  $a$  indica o neurônio oculto adicional. O procedimento adicional agora é o mesmo que no teste de não linearidade. A regressão linear artificial da equação (3.7) torna-se

$$u_t = \phi_1 \nabla G(\Theta; \mathbf{x}_{t-p}) + \phi_2 (\Psi(\gamma_{0a} + \Gamma_a^\top \mathbf{x}_{t-p})\beta_a) + u_t^*. \quad (3.25)$$

O segundo termo desta equação é aproximado por um polinômio de Taylor, de modo que a Equação (3.25) fica dada por

$$u_t = \phi_1 \nabla G(\Theta, \mathbf{x}_{t-p}) - \frac{1}{3} \sum_{j_1=1}^n \sum_{j_2=j_1}^n \sum_{j_3=j_2}^n \phi_{2j_1, j_2, j_3} X_{j_1} X_{j_2} X_{j_3} + u_t^*.$$

Assim, a hipótese nula pode ser escrita como

$$H_0 : \phi_{2_{j_1, j_2, j_3}} = 0 \quad \forall(j_1, j_2, j_3),$$

com alternativa

$$H_1 : \phi_{2_{j_1, j_2, j_3}} \neq 0 \quad \forall(j_1, j_2, j_3).$$

O cálculo das estatísticas de teste é o mesmo da subseção (3.1.1) (Equações (3.8) e (3.9)). A única diferença é o segundo grau de liberdade para a estatística do teste  $F$ , que aqui é  $(T - r)$ .

### 3.5 Medidas de Erro

Primeiramente, deve-se entender o que é erro de previsão. Para isso, assuma que estimamos um modelo que descreva a variável  $Y_t$ . Dado esse modelo, temos que a variável de interesse passa ser descrita como o valor previsto pelo modelo mais um erro, isto é

$$Y_t = \hat{Y}_t + \varepsilon_t.$$

Logo, o erro de previsão é dado por

$$\varepsilon_t = Y_t - \hat{Y}_t,$$

e o erro percentual,  $p_t$ , como

$$p_t = 100 \left( \frac{\varepsilon_t}{Y_t} \right).$$

Hyndman e Athanasopoulos (2018) separam as medidas de erro em quatro categorias: medidas que dependem da escala, medidas baseadas em erros percentuais, medidas baseadas em erros relativos e medidas relativas. Aqui, iremos tratar apenas das medidas que dependem da escala e dos erros percentuais, por serem as usadas na aplicação.

As medidas que dependem da escala dos dados são

- ME (*Mean Error*),
- RMSE (*Root Mean Square Error*),
- MAE (*Mean Absolute Error*).

A primeira consiste simplesmente na média dos erros, enquanto que a segunda é a raiz quadrada da média dos erros quadráticos e a terceira medida é dada pela média



dos erros em valores absolutos. Essas três medidas fazem parte do grupo de medidas que dependem da escala na qual os dados estão expressos.

Essas medidas podem ser utilizadas para efeito de comparação de diferentes modelos aplicados à mesma amostra de dados. [Hyndman e Athanasopoulos \(2018\)](#) chamam a atenção para o fato de que, apesar da medida RMSE ser mais utilizada, ela é mais sensível a *outliers*, quando comparada com a MAE. Abaixo a fórmula para o cálculo das três medidas.

$$ME = \frac{\sum_{t=1}^T \varepsilon_t}{T}, \quad RMSE = \sqrt{\frac{\sum_{t=1}^T \varepsilon_t^2}{T}} \quad \text{e} \quad MAE = \frac{\sum_{t=1}^T |\varepsilon_t|}{T}.$$

As medidas que não dependem da escala dos dados são

- MPE (*Mean Percentage Error*),
- MAPE (*Mean Absolute Percentage Error*),

em que, MPE representa o erro percentual médio, enquanto que MAPE é dada pela média do erro percentual em valor absoluto. Consequentemente, essas medidas têm a vantagem de não serem dependentes de escala e podem assim serem utilizadas para comparar o poder de previsão utilizando dados com diferentes escalas.

Um problema óbvio dessas medidas baseadas em erros percentuais é que elas não são definidas para  $Y_t = 0$ , além disso, [Hyndman e Athanasopoulos \(2018\)](#) apontam para problemas que envolvem a distribuição dos erros percentuais quando  $Y_t \rightarrow 0$ . Abaixo a fórmula para o cálculo das medidas.

$$MPE = \frac{\sum_{t=1}^T p_t}{T} \quad \text{e} \quad MAPE = \frac{\sum_{t=1}^T |p_t|}{T}.$$

Com todo o estudo completado, deve-se fazer aplicações e ver como os modelos AR-NN( $p$ ) se ajusta com dados reais.



## 4 Aplicação

Para ilustrar a metodologia dos modelos  $AR-NN(p)$ , foram modeladas duas séries temporais, uma contendo indícios de tendência e sazonalidade, e a outra, uma série com indícios de estacionariedade e variância não constante. As escolhas dessas séries se deram pela preocupação do ajuste do modelo  $AR-NN(p)$  em diferentes casos.

As séries são sobre vendas mensais (em milhões de dólares) de medicamentos para diabéticos na Austrália, do ano de 1991 a 2008. A outra série trata das variações mensais (em %) das ações preferenciais da Petrobras (PETR4), do ano de 1995 a 2019.

Para fins experimentais, 80% dos dados foram utilizados no treinamento do modelo (obtenção das estimativas dos parâmetros). Os outros 20% serviram para validação (comparar o valor real com o previsto).

Com o ajuste das duas séries, será feito o diagnóstico do modelo, previsões e algumas medidas para erro.

Todas as análises e gráficos foram feitos no *Software R (2019)*. Em todos os modelos foram utilizados a função de ativação logística por ela já ser implementada no *Software R (2019)*.

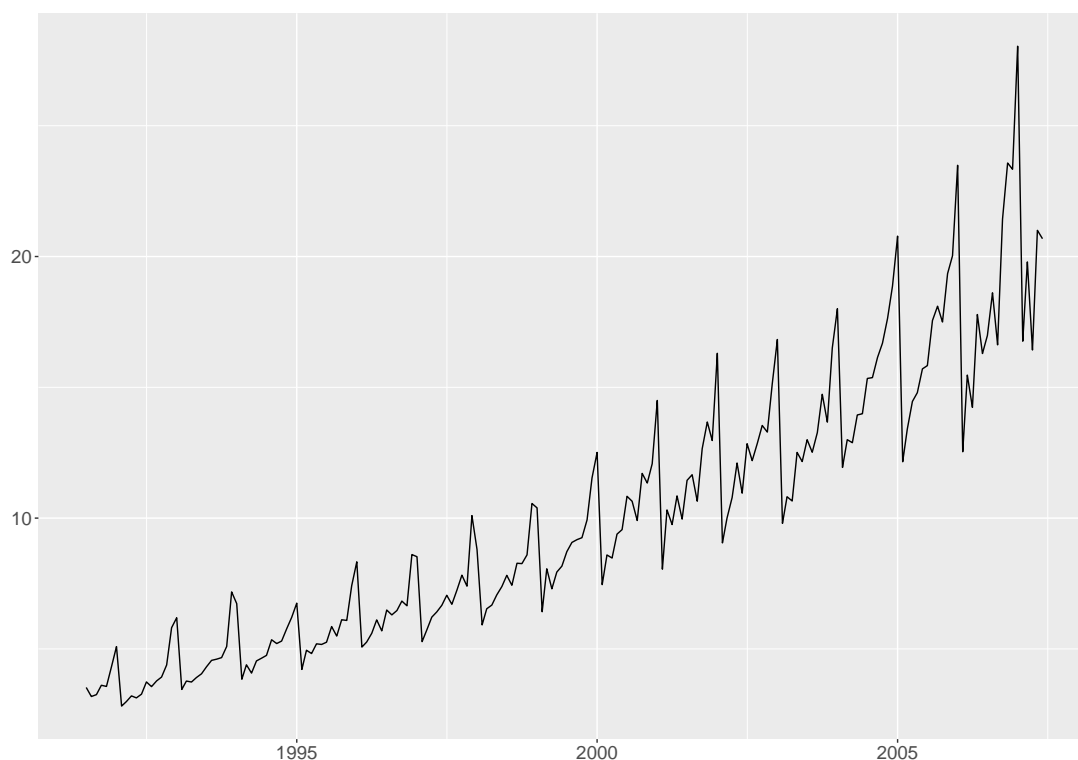
### 4.1 Vendas mensais de medicamentos para diabéticos na Austrália, de 1991 a 2008.

Diabetes é uma doença causada pela produção insuficiente ou má absorção de insulina, hormônio que regula a glicose no sangue e garante energia para o organismo. A insulina é um hormônio que tem a função de quebrar as moléculas de glicose (açúcar) transformando-as em energia para manutenção das células do nosso organismo.

O mercado global de medicamentos para diabéticos vem testemunhando um crescimento significativo no decorrer dos anos. Esse crescimento é atribuído à crescente prevalência de diabetes e demanda significativa de medicamentos orais para diabéticos. Além disso, sedentarismo, que está diretamente relacionado à obesidade e altos níveis de estresse é considerado um fator importante para o crescimento do mercado de medicamentos orais para diabéticos.

O conjunto de dados a ser analisado é sobre vendas mensais de medicamentos orais para diabéticos na Austrália, registrados pela *Australian Health Insurance Commission* do ano de 1991 a 2008. Os dados estão disponíveis no *software R* pelo pacote `fpp2`.

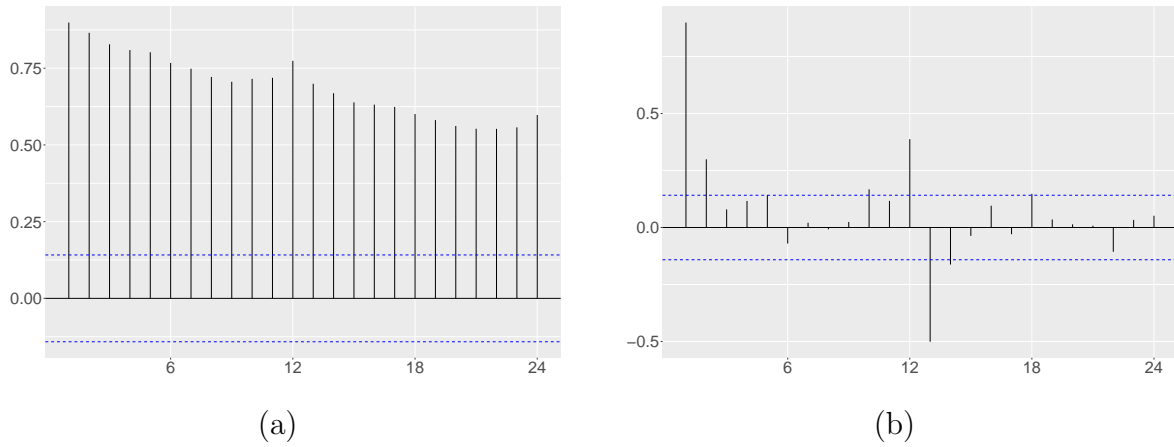
Figura 14 – Série sobre vendas mensais de medicamentos orais para diabéticos na Austrália em milhões de dólares ao longo dos meses.



Fonte: Autor.

A Figura (14), mostra o gráfico da série sobre vendas mensais de medicamentos orais para diabéticos na Austrália, onde percebemos indícios de tendência crescente e sazonalidade. Para evidenciar melhor esses indícios de tendência e sazonalidade, foram construídos os gráficos de autocorrelação e de autocorrelação parcial, os mesmos podem ser vistos na Figura (15).

Figura 15 – (a) Função de autocorrelação da série em relação as defasagens. (b) Função de autocorrelação parcial da série em relação as defasagens.



Fonte: Autor.

A Figura (15a) mostra as autocorrelações significativas, que persistem por dezenas de defasagens, o que dá indícios que a série precisa ser diferenciada. A Figura (15b) mostra com mais evidência indícios de sazonalidade, já que a mesma apresenta algumas oscilações ao longo das defasagens.

Para verificar todas as suspeitas feitas acima, será realizado um teste estatístico para cada uma das suposições.

#### 4.1.1 Tendência

Formalmente, foi realizado um teste de Wald-Wolfowitz para verificar a tendência da série.

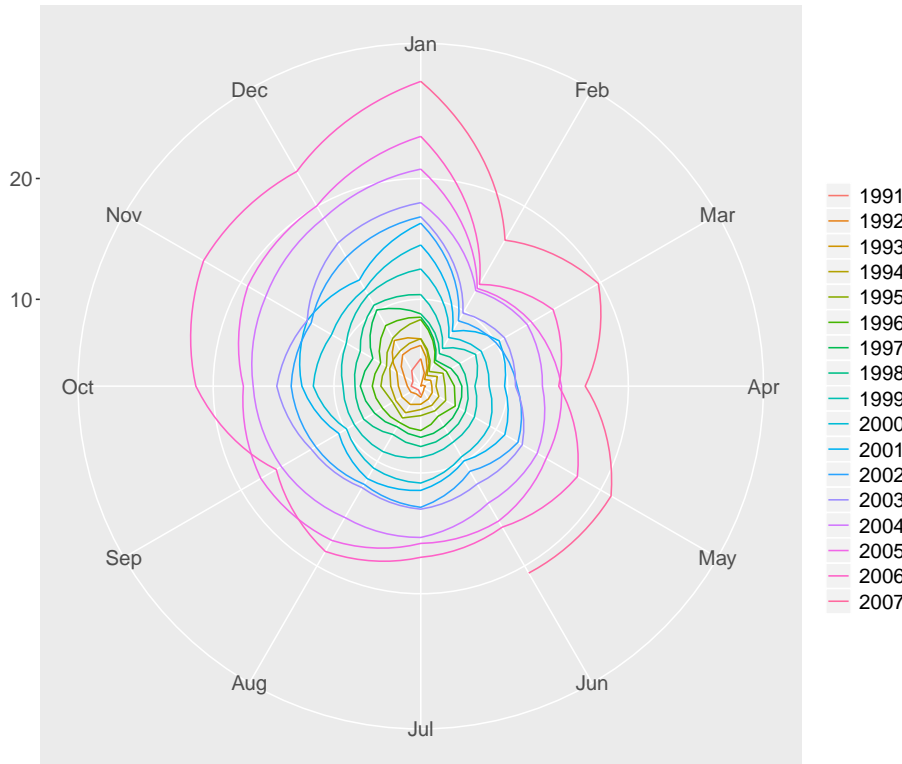
$$\begin{cases} H_0 : \text{Não Existe tendência,} \\ H_1 : \text{Existe tendência.} \end{cases}$$

Que conforme o esperado, foi rejeitada a hipótese nula de não tendência com um  $p$ -valor extremamente pequeno ( $p\text{-valor} = 2,2 \times 10^{-16}$ ).

#### 4.1.2 Sazonalidade

Pelas Figuras (15a) e (15b) é notório ver a sazonalidade. Para confirmar tal fato, foi realizado um teste estatístico formal. Porém, antes do teste, apresentaremos um gráfico sazonal polar, proposto por Hyndman e Athanasopoulos (2018), para tentar observar os períodos da sazonalidade. A Figura (16) mostra com detalhes o gráfico sazonal polar para vendas mensais de medicamentos orais para diabéticos na Austrália.

Figura 16 – Gráfico sazonal polar de todos os anos das vendas mensais de medicamentos ao longo dos meses.



Fonte: Autor.

Nota-se pela Figura (16), que ao longo dos anos, houve um aumento e um decaimento nos meses de Janeiro e Fevereiro, respectivamente. A partir disso, temos mais evidências de que a série contém sazonalidade. Formalmente, foi realizado um teste de Friedman para verificar se existe sazonalidade, em que suas hipóteses são

$$\begin{cases} H_0 : \text{Não há sazonalidade determinística} , \\ H_1 : \text{Há sazonalidade determinística} . \end{cases}$$

Que conforme o esperado foi rejeitado a hipótese nula de não sazonalidade, com um  $p$ -valor extremamente pequeno ( $p$ -valor=  $5,2 \times 10^{-4}$ ).

### 4.1.3 Raiz Unitária

Uma maneira de determinar mais objetivamente se a diferenciação é necessária, é usando um teste de raiz unitária. Como visto anteriormente, o teste de classificação aumentada Dickey-Fuller *rank* serve exatamente para identificar se a diferenciação é

necessária, ou seja, se a série possui raiz unitária. Temos como hipóteses,

$$\begin{cases} H_0 : \text{Tem raiz unitária,} \\ H_1 : \text{Não tem raiz unitária.} \end{cases}$$

Ao aplicar o teste, obtivemos um  $p$ -valor maior que o nível de 5%, o que nos diz que a série tem raiz unitária e que os dados precisam ser diferenciados ( $p$ -valor = 0,24).

#### 4.1.4 Teste de White

Para verificar se a série tem alguma não linearidade negligenciada, foi realizado o teste de White, visto na subseção (3.1.1).

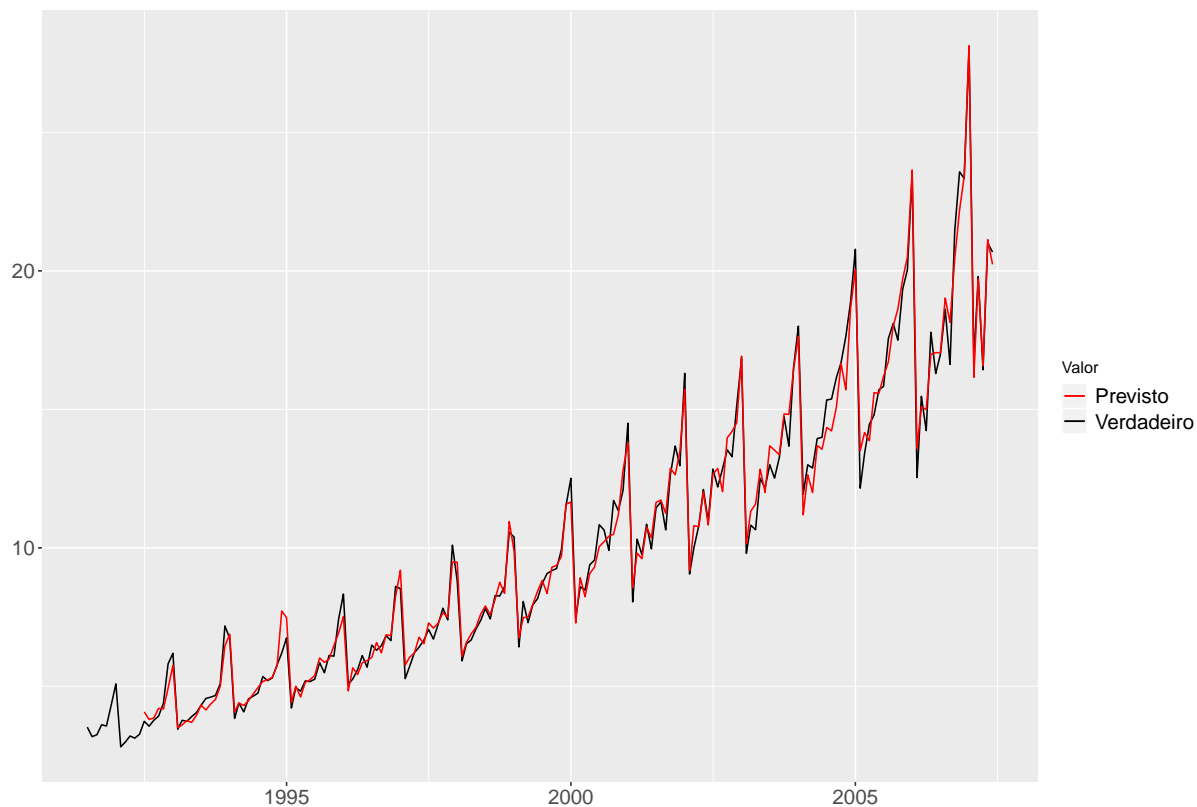
$$\begin{cases} H_0 : \text{Um modelo AR linear explica os dados,} \\ H_1 : \text{Um modelo AR linear não explica os dados.} \end{cases}$$

Com isso, obtivemos um  $p$ -valor extremamente pequeno, o que nos diz que um modelo AR linear não é capaz de explicar corretamente os dados ( $p$ -valor = 0,0003637).

#### 4.1.5 Modelo

A partir das análises acima, propusemos um modelo AR-NN(5) com uma camada e seis neurônios ocultos. Esse modelo foi escolhido, porque a parte linear contém o menor AIC, e parte não linear, por ser o número máximo de neurônios significativos no teste *bottom-up*. Com isso, a Figura (17) mostra um bom ajuste do modelo ao longo da série.

Figura 17 – Valores estimados com o Modelo AR-NN(5) com 1 camada e 6 neurônios ocultos ajustado sobre a série vendas mensais de medicamentos para diabéticos na Austrália.

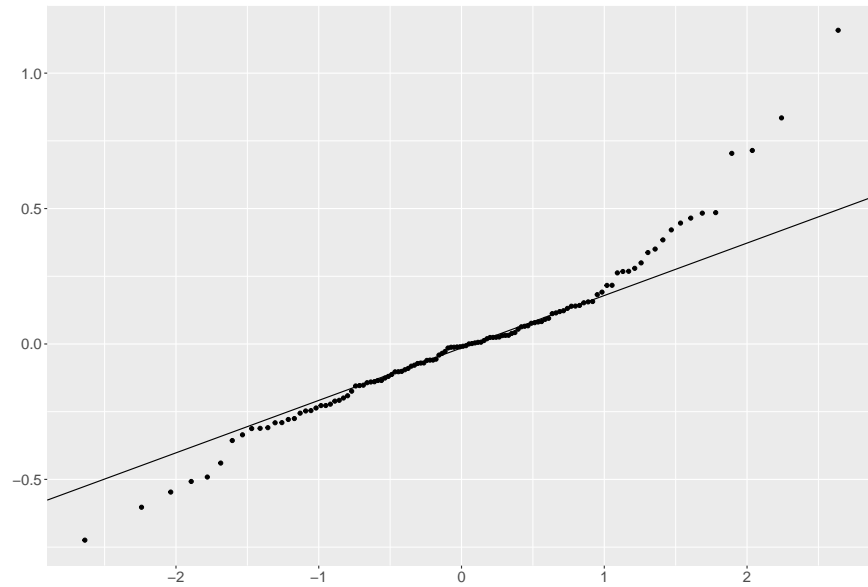


Fonte: Autor.

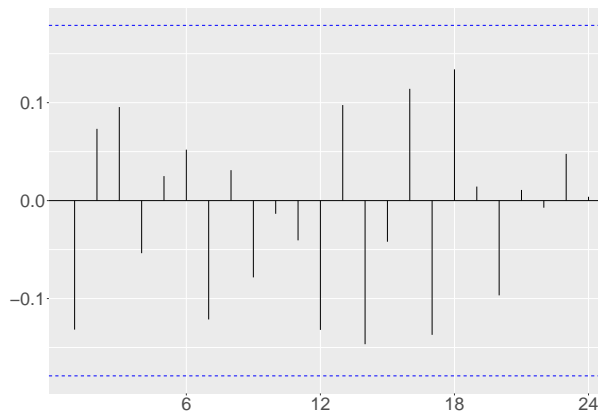
Ainda, apresentaremos um diagnóstico de modelo para ver se os resíduos estão de acordo com as propostas vistas anteriormente. Pela Figura (18), podemos observar que o `qqplot` dos resíduos foge da normalidade. Porém, não existe pontos significativos nas funções de autocorrelação e de autocorrelação parcial, o que nos diz que o modelo está bem ajustado.



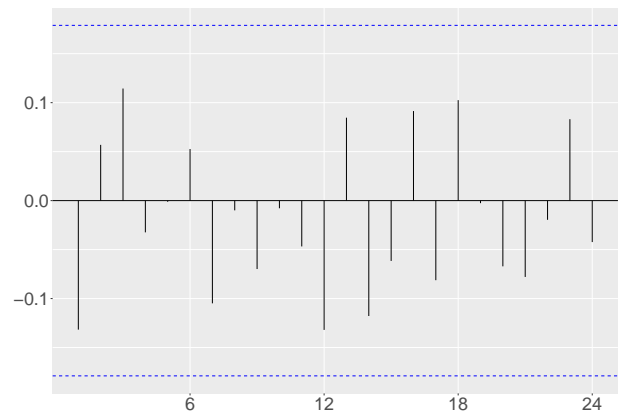
Figura 18 – (a) Quantis da distribuição dos resíduos contra os quantis da distribuição normal (QQ-plot). (b) Função de autocorrelação dos resíduos em relação as defasagens. (c) Função de autocorrelação parcial dos resíduos em relação as defasagens.



(a)



(b)

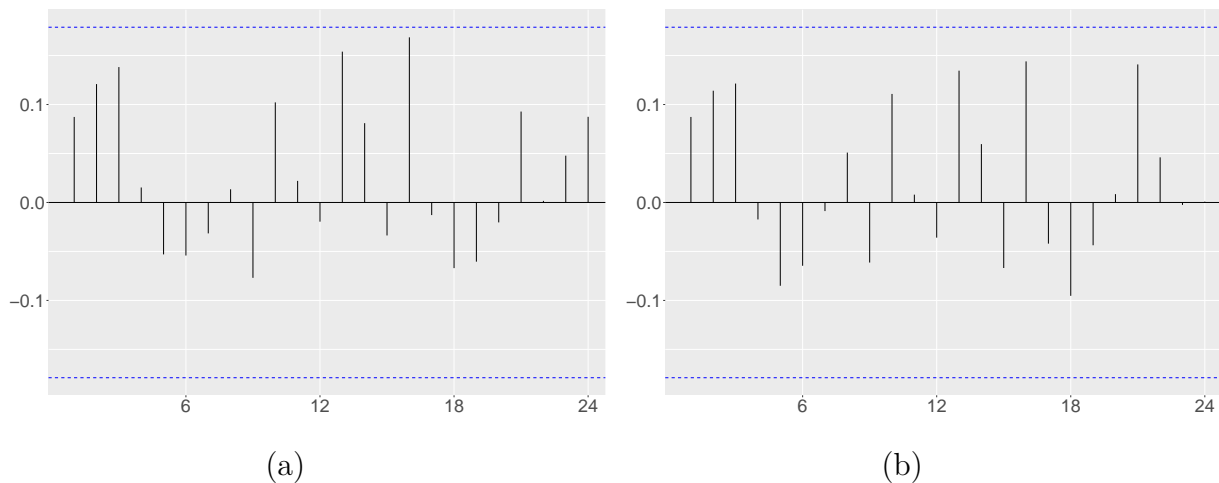


(c)

Fonte: Autor.

Realizando o mesmo diagnóstico para os resíduos ao quadrado, também não obtivemos pontos significativos nas funções de autocorrelação e de autocorrelação parcial, o que nos diz que o modelo tratou bem a volatilidade.

Figura 19 – (a) Função de autocorrelação dos resíduos ao quadrado em relação as defasagens. (b) Função de autocorrelação parcial dos resíduos ao quadrado em relação as defasagens.



Fonte: Autor.

#### 4.1.5.1 Previsões

Com o ajuste do modelo e a análise de diagnóstico feitas, chegou a hora de fazer as previsões e compará-las com os valores reais de série. Foram realizadas previsões para 12 meses, e as mesmas podem ser vistas na Tabela (2) e Figura (20).

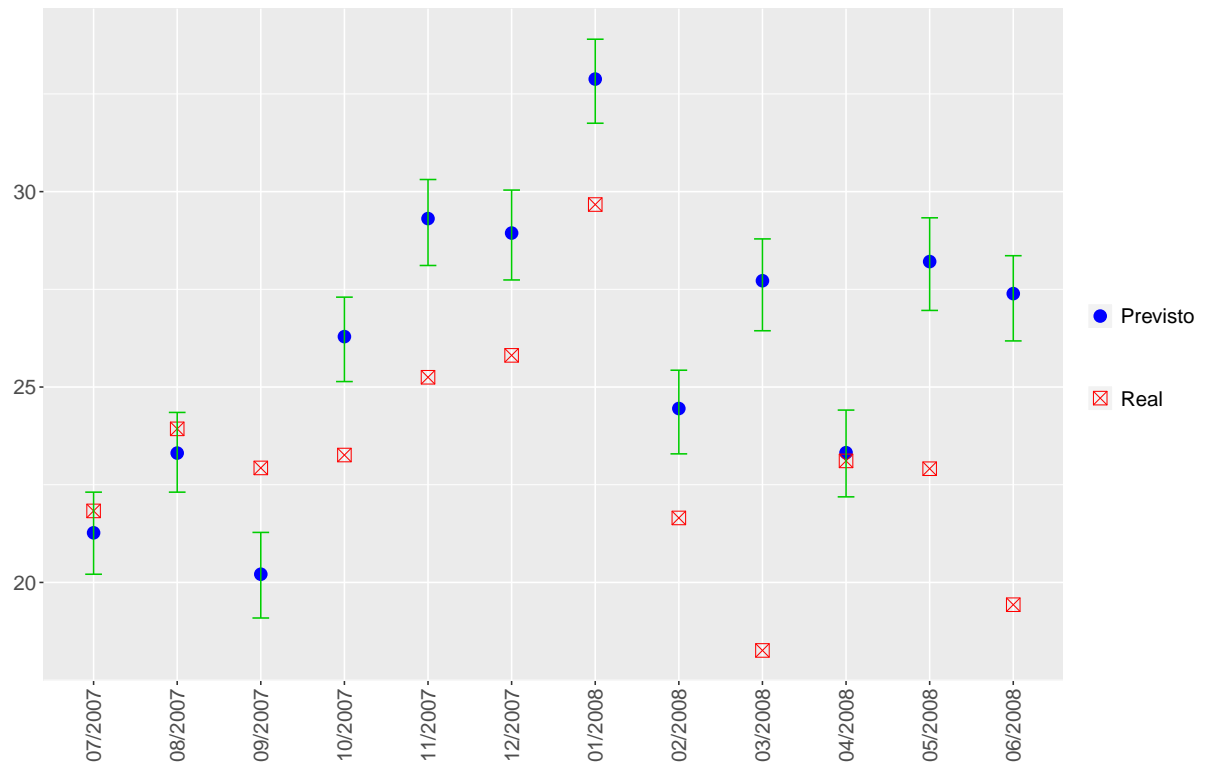
Tabela 2 – Previsões.

Mês	Previsão	Valor Real	LI 95%	LS 95%
07/2007	21,27	21,83	20,21	22,31
08/2007	23,31	23,93	22,31	24,35
09/2007	20,21	22,93	19,09	21,28
10/2007	26,29	23,26	25,14	27,30
11/2007	29,31	25,25	28,11	30,31
12/2007	28,94	25,81	27,74	30,04
01/2008	32,88	29,67	31,75	33,90
02/2008	24,45	21,65	23,29	25,43
03/2008	27,72	18,26	26,44	28,79
04/2008	23,32	23,11	22,19	24,41
05/2008	28,21	22,91	26,96	29,33
06/2008	27,39	19,43	26,18	28,36

LI: Limite inferior. LS: Limite superior

Fonte: Autor.

Figura 20 – Representação gráfica da Tabela (2).



Fonte: Autor.

Podemos perceber que existem vários meses onde o valor real encontra-se fora do intervalo de confiança, isso se deve por estarmos ajustando um modelo autorregressivo de redes neurais sem nenhuma componente sazonal, o que faz o modelo ter dificuldades nesses tipos de dados. Com isso, podemos dizer que o modelo é razoável para previsões a curto prazo já que o intervalo de confiança dos dois primeiros meses englobam os valores reais.

Foram calculadas algumas medidas de erro do modelo sobre o conjunto de treinamento e o conjunto de teste. Tais medidas podem ser vistas na Tabela (3).

Tabela 3 – Medidas de erro.

	ME	RMSE	MAE	MPE	MAPE
Conjunto de Treinamento	0,00	0,52	0,38	-0,34	3,93
Conjunto de Teste	-2,94	4,50	3,59	-13,72	16,57

Fonte: Autor.

O modelo se adequou bem aos dados, tendo um bom ajuste tanto nas autocorrelações como na variância. Porém, o modelo não teve bom desempenho nas previsões, ficando o

valor real várias vezes fora do intervalo de confiança. Mesmo assim, apresentou previsões bem próximas do valor real. Pelas medidas de erro, podemos ver que o modelo se adequou bem ao conjunto de treinamento, já que obteve erros quase sempre próximos de zero. Quando isso acontece, dizemos que o modelo é sobre-ajustado (*overfitting*). Um modelo sobre-ajustado apresenta alta precisão quando testado com seu conjunto de treinamento, porém tal modelo não é uma boa representação da realidade, ou seja, apresenta medidas de erro altos para as previsões.

## 4.2 Ações preferenciais da Petrobras - PETR4

A Petróleo Brasileiro S.A, popularmente conhecida como Petrobras, é uma empresa de capital aberto, ou seja, o seu capital social é formado por ações que podem ser adquiridas em leilão no mercado.

Ações representam uma fração do capital social de uma empresa.

Atualmente, ela opera em 25 países nas áreas de exploração, produção, refino, comercialização e transporte de petróleo, gás natural e seus derivados. A Petrobras oferece no mercado dois tipos de ações e são conhecidas por seus códigos diferentes, PETR3 e a PETR4.

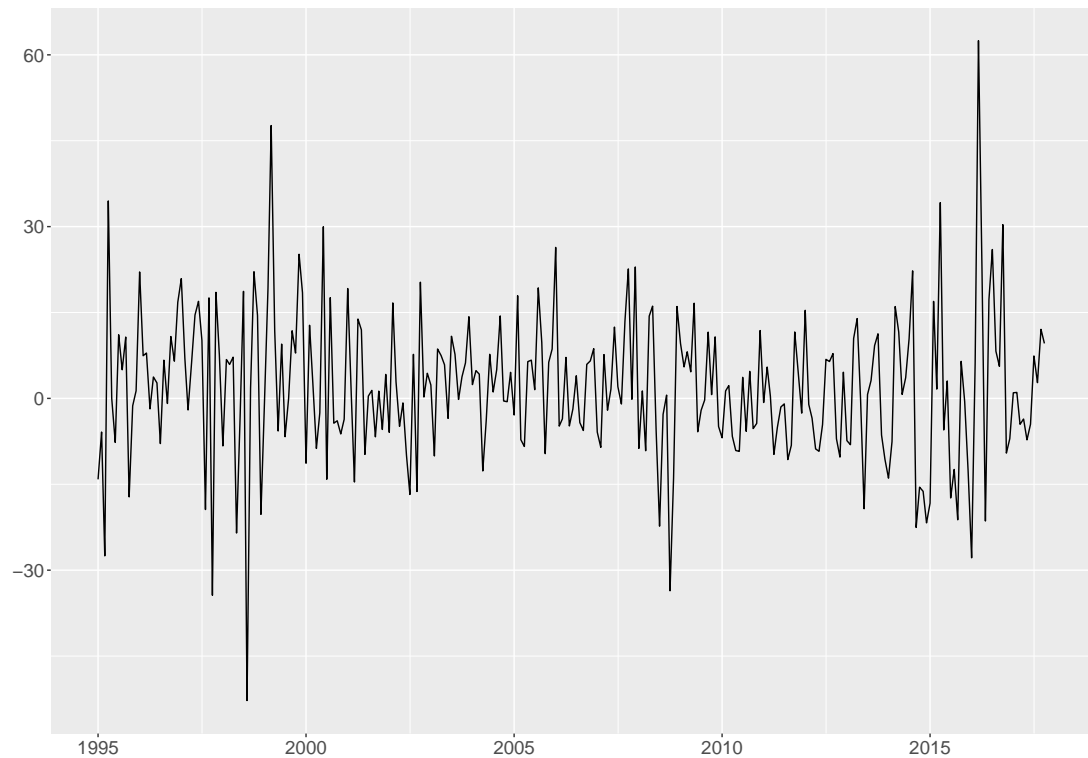
- PETR3: é uma ação ordinária (ON) e os acionistas que a possuem ganham direito de voto nas assembleias. Porém, eles recebem um menor percentual de dividendos.
- PETR4: é uma ação preferencial (PN) e muitos investidores a adquirirem porque possuem uma participação maior na divisão de lucros da empresa.

A maior parte dos estudos financeiros concentra-se na análise de séries de retornos ao invés do uso da série dos preços dos ativos. A razão de utilizarmos séries de retornos tem dois fatores, as informações de retornos atendem aos interesses de investidores e a série de retornos possui propriedades estatísticas mais interessantes do que série dos preços. Algumas dessas propriedades são

- estacionariedade,
- fraca dependência linear e não linear,
- caudas pesadas na distribuição e excesso de curtose.

O conjunto de dados é sobre os retornos das ações preferenciais da Petrobras (PETR4), em meses, do ano 1995 até 2019. Os dados estão disponíveis no site <<https://br.investing.com/equities/petrobras-pn-historical-data>>.

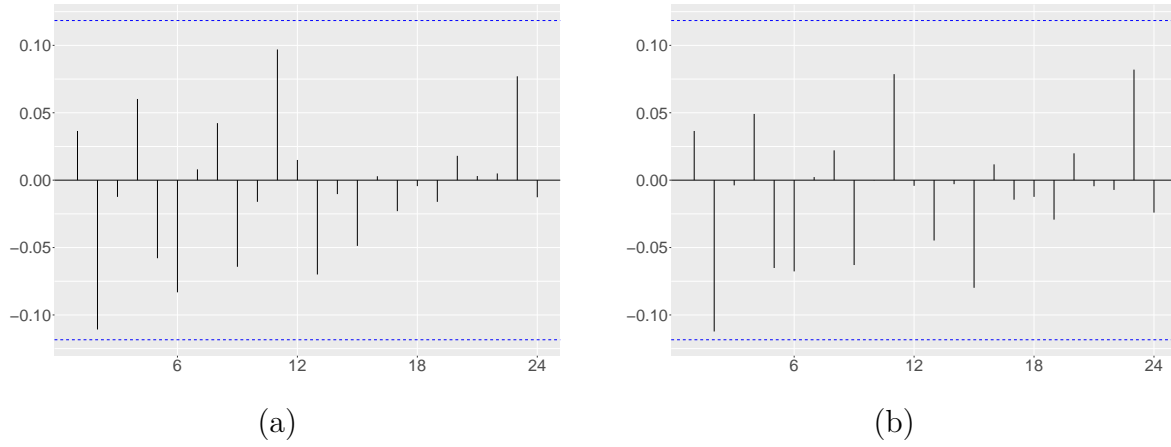
Figura 21 – Retornos das ações preferenciais da Petrobras (PETR4) ao longo dos meses.



Fonte: Autor.

A Figura (21), mostra o gráfico da série sobre os retornos mensais das ações preferenciais da Petrobras, e com ele, não temos indícios de tendência e/ou sazonalidade. Por se tratar de uma séries de retornos, temos indícios de estacionaridade. Para evidenciar melhor, foram construídos os gráficos de autocorrelação (ACF) e de autocorrelação parcial (PACF). Os mesmos podem ser vistos na Figura (22).

Figura 22 – (a) Função de autocorrelação da série em relação as defasagens. (b) Função de autocorrelação parcial da série em relação as defasagens.



Fonte: Autor.

As Figuras (22a) e (22b) não mostram autocorrelações significativas por dezenas de defasagens, como também, não mostram indícios de sazonalidade.

Para verificar todos os fatos, foi realizado um teste estatístico para cada uma das suposições.

#### 4.2.1 Tendência

Formalmente, foi realizado um teste de Wald-Wolfowitz para verificar a tendência da série.

$$\begin{cases} H_0 : \text{Não Existe tendência,} \\ H_1 : \text{Existe tendência.} \end{cases}$$

Que conforme o esperado, não foi rejeitado a hipótese nula de não tendência, ou seja, a série de fato é estacionária. ( $p$ -valor = 0,36)

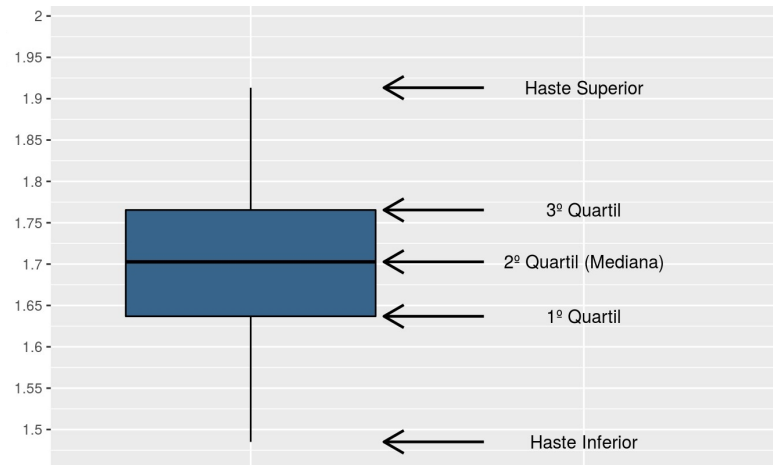
#### 4.2.2 Sazonalidade

Diferente da primeira análise, não é notório pelas Figuras (22b) e (22c) ver a sazonalidade. Foi feita uma breve análise do gráfico *box plot* dos meses, e em seguida foi realizado um teste de Friedman. Tal teste analisa se a série tem ou não sazonalidade determinística e as hipóteses de desses testes são

$$\begin{cases} H_0 : \text{Não há sazonalidade determinística ,} \\ H_1 : \text{Há sazonalidade determinística .} \end{cases}$$

A Figura (23) mostra uma breve explicação do gráfico *box plot*. Ela serve para se ter um melhor entendimento da Figura (24).

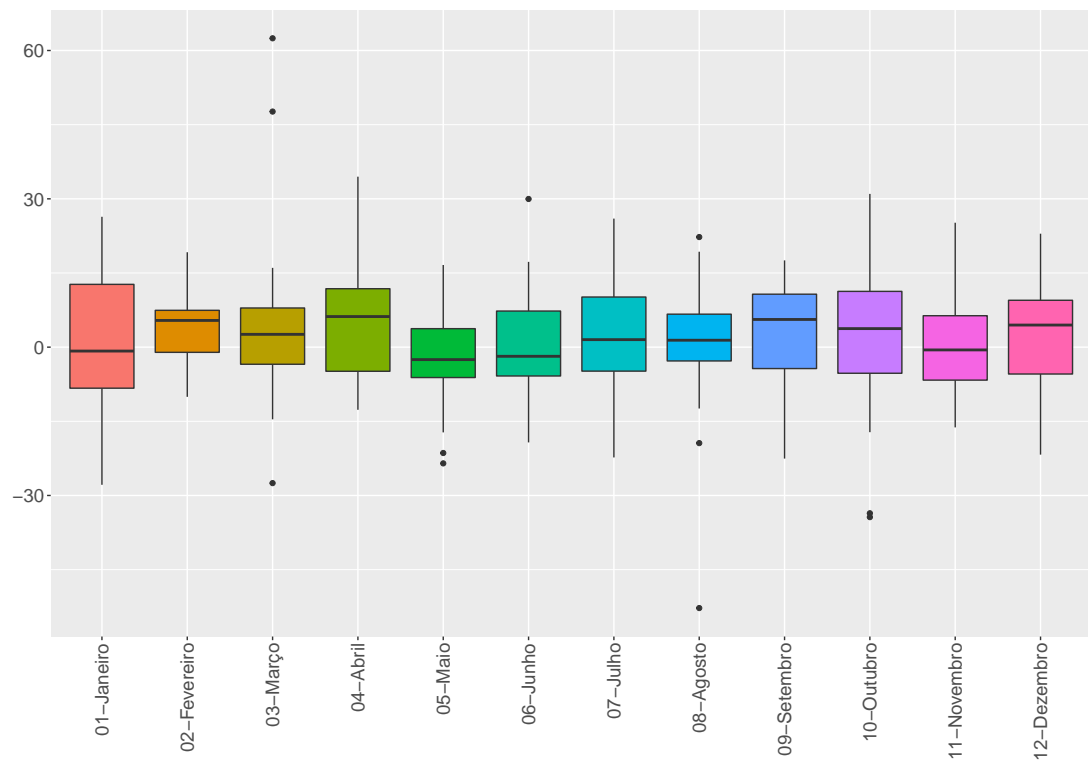
Figura 23 – Breve explicação do gráfico *box plot*.



Fonte: Autor.

A Figura (24) mostra com detalhes a variação da mediana de cada mês durante todos os anos.

Figura 24 – *Box plot* do retorno para cada mês.



Fonte: Autor.

Nota-se pela Figura (24), que ao longo dos anos, houve algumas variações nos meses. Com isso, devemos testar se essas variações foram significativas ao ponto de serem consideradas uma sazonalidade. Usando um teste estatístico formal, teste de Friedman, obtivemos um valor maior que 5%, ou seja, a série não tem sazonalidade determinística ( $p$ -valor = 0,24).

#### 4.2.2.1 Raiz Unitária

Uma maneira de determinar mais objetivamente se a diferenciação é necessária, é usando um teste de raiz unitária. Como visto anteriormente, o teste de classificação aumentada de Dickey-Fuller *rank* serve exatamente para identificar se a diferenciação é necessária, ou seja, se a série possui raiz unitária.

$$\begin{cases} H_0 : \text{Tem raiz unitária;} \\ H_1 : \text{Não tem raiz unitária.} \end{cases}$$

Ao aplicar o teste, obtivemos um  $p$ -valor menor que o nível de 5%, o que nos diz que a série não tem raiz unitária ( $p$ -valor = 0,004).

#### 4.2.2.2 Teste White

Para verificar se a série tem alguma não linearidade negligenciada, foi realizado o teste de White.

$$\begin{cases} H_0 : \text{Um modelo AR linear explica os dados;} \\ H_1 : \text{Um modelo AR linear não explica os dados.} \end{cases}$$

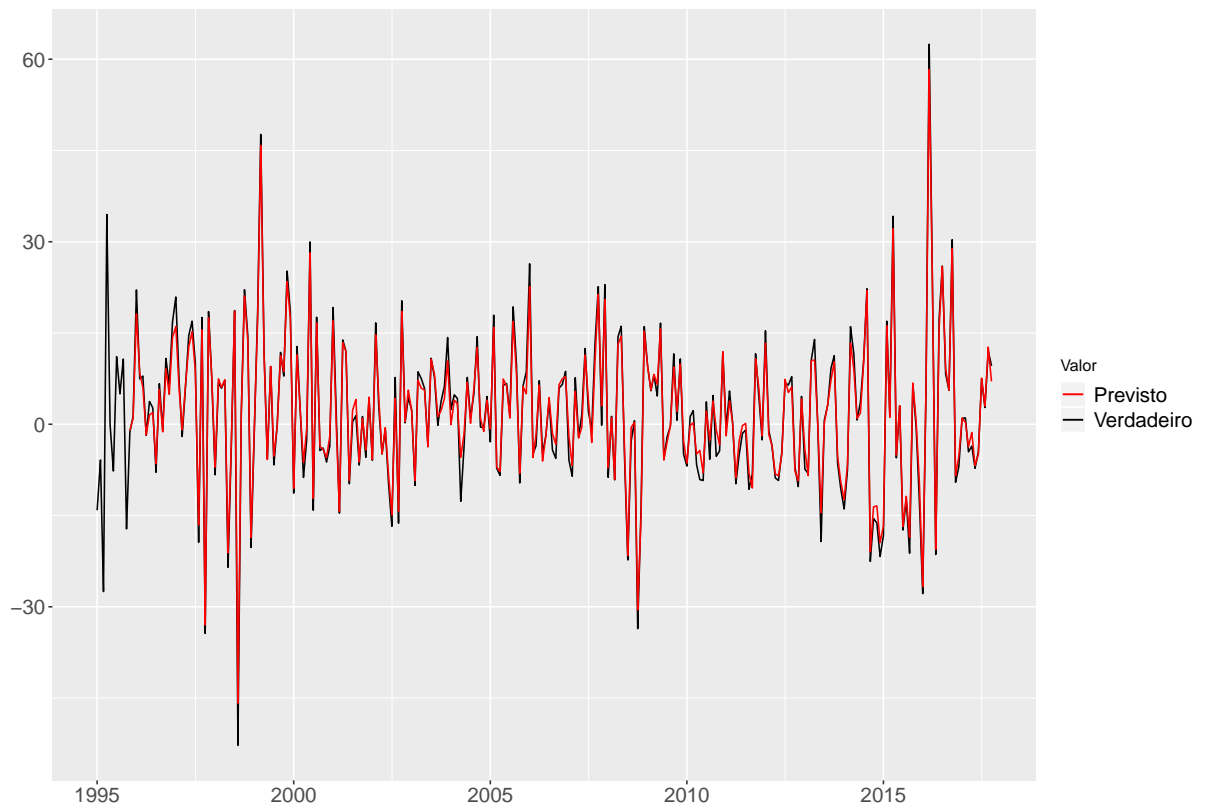
Com isso, obtivemos um  $p$ -valor extremamente alto, o que nos diz que um modelo AR linear é capaz de explicar corretamente os dados. Porém, em diversas aplicações vemos que isso não é possível por causa da alta volatilidade, e como nosso objetivo é ver como o modelo se adéqua a esses tipos de dados, será ajustado um modelo AR-NN( $p$ ) ( $p$ -valor = 0.96).

#### 4.2.3 Modelo

A partir das análises acima, propusemos um modelo AR-NN(17) com uma camada e dez neurônios ocultos. Esse modelo foi escolhido, porque a parte linear contém o menor AIC, e a parte não linear, por ser o número máximo de neurônios significativos no teste *bottom-up*. Com isso, a Figura (25) mostra um bom ajuste do modelo ao longo da série.



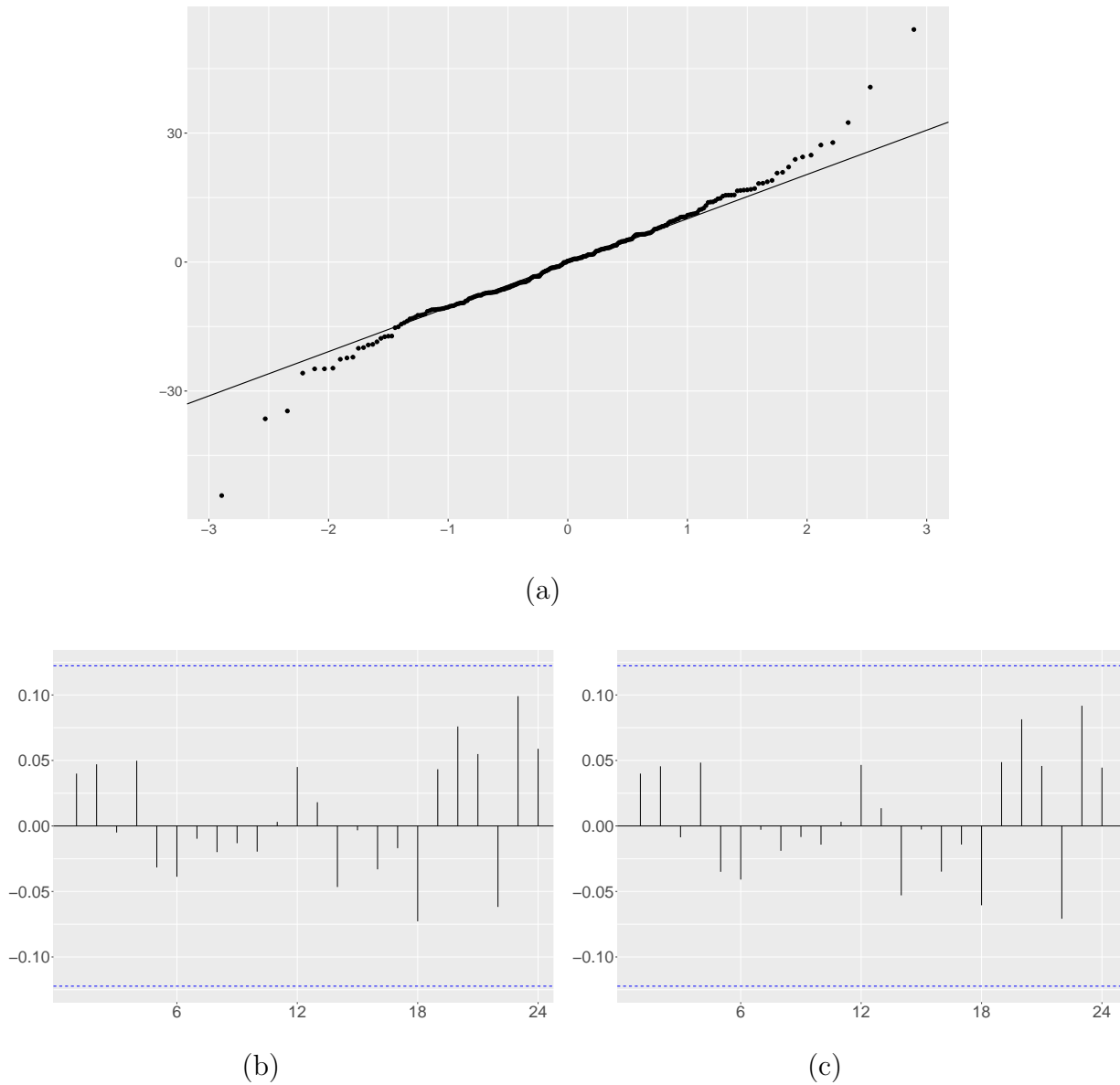
Figura 25 – Valores estimados com o Modelo AR-NN(17) com 1 camada e 10 neurônios ocultos ajustado sobre a série das ações preferenciais da Petrobras - PETR4.



Fonte: Autor.

Feito o diagnóstico do modelo, para ver se os resíduos estão de acordo com as propostas vistas anteriormente, podemos observar pela Figura (26) que o `qqplot` dos resíduos foge um pouco da normalidade, tendo caudas pesadas. Porém, não existe pontos significativos nas funções de autocorrelação e de autocorrelação parcial, o que nos diz que o modelo está bem ajustado.

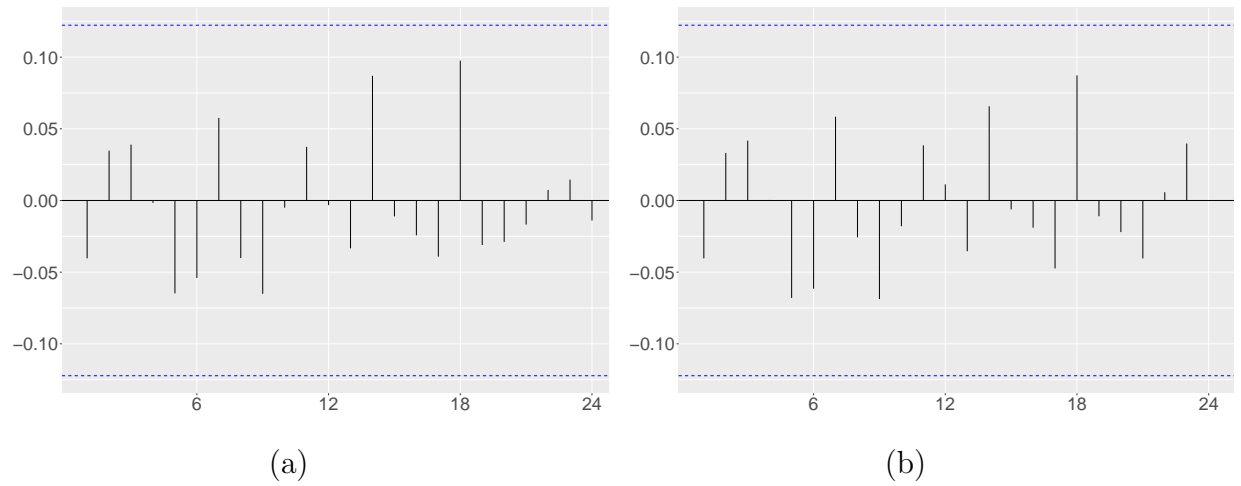
Figura 26 – (a) Quantis da distribuição dos resíduos contra os quantis da distribuição normal (QQ-plot). (b) Função de autocorrelação dos resíduos em relação as defasagens. (c) Função de autocorrelação parcial dos resíduos em relação as defasagens.



Fonte: Autor.

Realizando o mesmo diagnóstico para os resíduos ao quadrado, podemos ver pela Figura (27) que também não obtivemos pontos significativos nas funções de autocorrelação e de autocorrelação parcial, o que nos diz que o modelo tratou bem a volatilidade.

Figura 27 – (a) Função de autocorrelação dos resíduos ao quadrado em relação as defasagens. (b) Função de autocorrelação parcial dos resíduos ao quadrado em relação as defasagens.



Fonte: Autor.

#### 4.2.4 Previsões

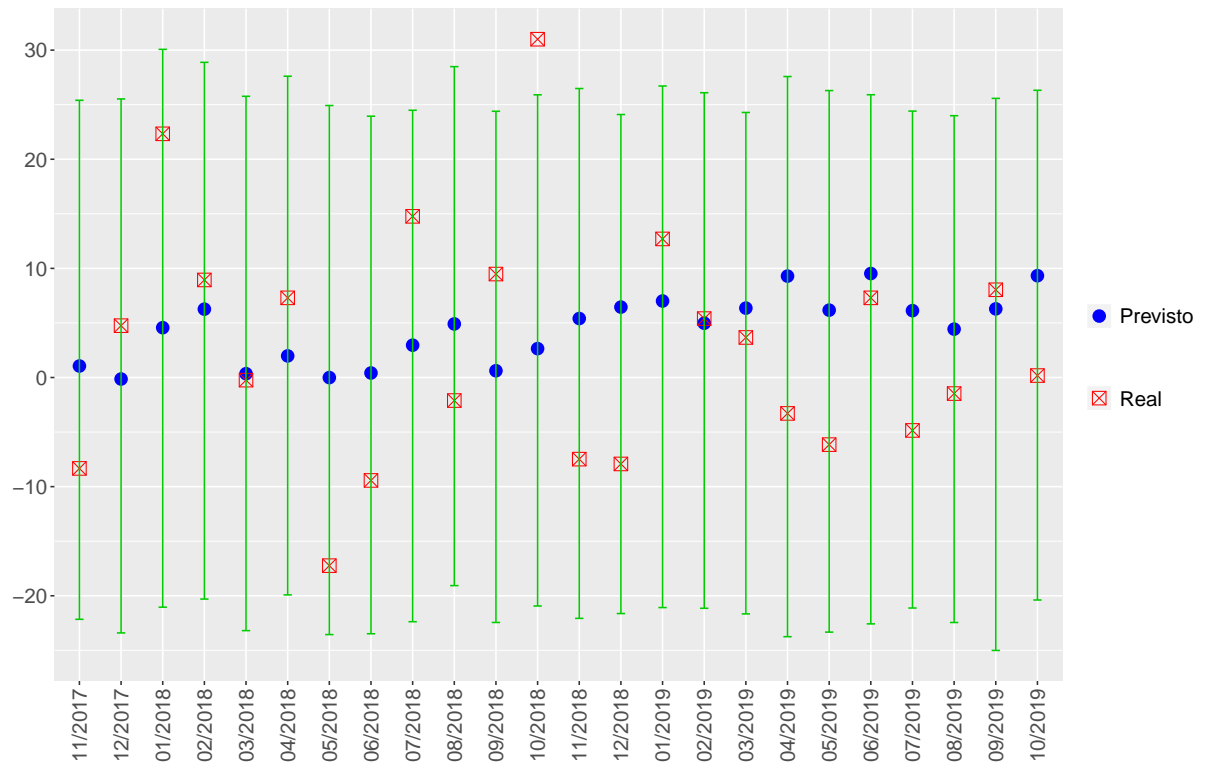
Com o ajuste do modelo e a análise de diagnóstico feitas, chegou a hora de fazer as previsões e compará-las com os valores reais de série. Foram realizadas previsões para 24 meses, e as mesmas podem ser vistas na Tabela (4) e na Figura (28).

Tabela 4 – Previsões dos retornos mensais das ações PETR4.

Mês	Previsão	Valor Real	LI 95%	LS 95%
11/2017	1,05	-8,33	-22,16	25,40
12/2017	-0,14	4,75	-23,41	25,53
01/2018	4,57	22,33	-21,04	30,07
02/2018	6,26	8,94	-20,30	28,88
03/2018	0,35	-0,24	-23,19	25,77
04/2018	1,98	7,30	-19,92	27,61
05/2018	-0,01	-17,24	-23,56	24,92
06/2018	0,41	-9,43	-23,48	23,95
07/2018	2,96	14,77	-22,38	24,49
08/2018	4,91	-2,11	-19,07	28,49
09/2018	0,62	9,48	-22,44	24,40
10/2018	2,65	31,00	-20,94	25,90
11/2018	5,40	-7,48	-22,07	26,48
12/2018	6,45	-7,92	-21,63	24,10
01/2019	7,01	12,70	-21,08	26,71
02/2019	4,97	5,40	-21,14	26,10
03/2019	6,36	3,67	-21,66	24,29
04/2019	9,29	-3,29	-23,75	27,58
05/2019	6,17	-6,15	-23,33	26,29
06/2019	9,53	7,30	-22,58	25,91
07/2019	6,13	-4,85	-21,12	24,41
08/2019	4,43	-1,47	-22,45	24,00
09/2019	6,29	8,04	-25,01	25,58
10/2019	9,33	0,18	-20,39	26,32

Fonte: Autor.

Figura 28 – Representação gráfica da Tabela (4).



Fonte: Autor.

Podemos perceber que existe apenas um mês fora do intervalo de confiança, o mesmo pode ser justificado por ser o mês correspondente às eleições presidenciais de 2018. Normalmente eleições presidenciais modificam bastante os valores das ações. Com isso, podemos dizer que é um modelo bem ajustado com boas previsões a longo prazo.

Foram calculadas algumas medidas de erro do modelo sobre o conjunto de treinamento e o conjunto de teste. Tais medidas podem ser vistas na Tabela (5).

Tabela 5 – Medidas de erro.

	ME	RMSE	MAE
Conjunto de Treino	0,01	1,86	1,43
Conjunto de Teste	-3,12	9,23	7,55

Fonte: Autor.

Nesse conjunto de dados foram calculadas apenas medidas de erro que dependem da escala dos dados, como o conjunto de dados tem valores próximos de zero, o cálculo das medidas de erro que não dependem da escalar conduziria aos problemas apresentados na seção (3.5).

O modelo se adequou bem aos dados, tendo um bom ajuste tanto nas autocorrelações como na variância. Além disso, o modelo teve ótimo desempenho nas previsões ficando o valor real sempre dentro do intervalo de confiança. Pelas medidas de erro, podemos ver que o modelo se adequou bem, tanto ao conjunto de treinamento como no conjunto de teste, os mesmos obtiveram erros considerados pequenos.

## 5 Conclusões

Os principais objetivos dessa monografia foram alcançados. O estudo do modelo AR-NN( $p$ ) foi realizado, compreendendo sua fundamentação teórica, arquiteturas e algoritmos de aprendizagem.

Modelos autorregressivos de redes neurais apresentaram bons resultados preditivos para a série econométrica (PETR4), com linearidade presente. Os modelos AR-NN( $p$ ) podem ser alternativas viáveis aos modelos existentes, já que apresentaram boas aproximações.

No estudo da série com tendência e sazonalidade (medicamentos para diabéticos na Austrália), o modelo encontrou dificuldades nos resultados preditivos, ficando o valor real várias vezes fora do intervalo de confiança. Mesmo assim, obteve valores bem próximos dos verdadeiros, dando a entender que não foi adequado a forma em que os intervalos de confiança foram construídos.

Dessa forma, conclui-se que os modelos AR-NN( $p$ ) apresentaram um bom desempenho de ajuste e predição em séries com e sem linearidade, volatilidade (variância condicionada não constante) e sazonalidade.

### 5.1 Recomendações para Trabalhos Futuros

- Estudo comparativo entre modelos AR-NN( $p$ ) e SARIMA( $p, d, q$ )  $\times$  ( $P, D, Q$ ): Ajustar os modelos AR-NN( $p$ ) e SARIMA( $p, d, q$ )  $\times$  ( $P, D, Q$ ) e verificar o desempenho dos modelos para um certo conjunto de dados;
- Desempenho dos modelos não lineares com diferentes funções de ativação: Ajustar os modelos AR-NN( $p$ ) com diferentes tipos de funções de ativação e verificar o desempenho dos modelos para um certo conjunto de dados;
- Modelos SARIMA-NN: Realizar um estudo sobre modelos SARIMA-NN, o mesmo pode ser obtido adicionando a parte sazonal e de médias móveis aos modelos AR-NN( $p$ );
- Desempenho dos modelos não lineares com diferentes tipos de redes neurais (redes neurais recorrentes ou *deep learning*): Fazer o estudo dos modelos AR-NN( $p$ ) com outros tipos de redes neurais e verificar seu desempenho.
- Modelos híbridos: Métodos lineares de previsão como o modelo autorregressivo integrado de médias móveis (ARIMA) e métodos não lineares como redes neurais

artificiais estão sujeitos à problemas de especificação de modelo. Sistemas híbridos propostos na literatura visam realizar uma correção das previsões originais através da previsão da série de erros. A previsão final pode ser obtida através da combinação das previsões das séries temporal e de resíduos através de operadores regressão linear e redes neurais.



# Referências

AKAIKE, H. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, Institute of Electrical and Electronics Engineers (IEEE), v. 19, n. 6, p. 716–723, dec 1974. Citado na página 54.

ANDERS, D. U. Statistische neuronale netze. *Handbuch*, 1997. Citado 7 vezes nas páginas 40, 43, 49, 53, 57, 61 e 68.

ANDERSON, J. A. Neurocomputing: Foundations of research (v. 1). Bradford Books, 1988. Disponível em: <<https://www.amazon.com/Neurocomputing-Foundations-Research-v-1/dp/0262010976?SubscriptionId=AKIAIOBINVZYXZQZ2U3A&tag=chimbori05-20&linkCode=xm2&camp=2025&creative=165953&creativeASIN=0262010976>>. Citado na página 62.

BISHOP, C. M. Neural networks for pattern recognition. Oxford University Press, 1995. Disponível em: <[https://www.ebook.de/de/product/3242184/christopher\\_m\\_bishop\\_neural\\_networks\\_for\\_pattern\\_recognition.html](https://www.ebook.de/de/product/3242184/christopher_m_bishop_neural_networks_for_pattern_recognition.html)>. Citado 5 vezes nas páginas 60, 62, 63, 64 e 65.

BOTTOU, L. Stochastic learning. Springer Berlin Heidelberg, p. 146–168, 2004. Citado na página 56.

BOX, G. E. P.; JENKINS, G. M. *Time Series Analysis: Forecasting and Control (Revised Edition)*. Holden-Day, 1976. ISBN 0816211043. Disponível em: <<https://www.amazon.com/Time-Analysis-Forecasting-Control-Revised/dp/0816211043?SubscriptionId=AKIAIOBINVZYXZQZ2U3A&tag=chimbori05-20&linkCode=xm2&camp=2025&creative=165953&creativeASIN=0816211043>>. Citado 5 vezes nas páginas 30, 39, 45, 47 e 49.

BURNHAM, K. P.; ANDERSON, D. R. Multimodel inference. *Sociological Methods & Research*, SAGE Publications, v. 33, n. 2, p. 261–304, nov 2004. Citado na página 54.

CYBENKO, G. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems*, Springer Science and Business Media LLC, v. 2, n. 4, p. 303–314, dec 1989. Citado na página 44.

DAVIDSON, R.; MACKINNON, J. G. Estimation and inference in econometrics. Oxford University Press, 1993. Disponível em: <<https://www.amazon.com/Estimation-Inference-Econometrics-Russell-Davidson/dp/0195060113?SubscriptionId=AKIAIOBINVZYXZQZ2U3A&tag=chimbori05-20&linkCode=xm2&camp=2025&creative=165953&creativeASIN=0195060113>>. Citado na página 53.

DIETZ, S. Autoregressive neural network processes - univariate, multivariate and cointegrated models with application to the german automobile industry. 06 2011. Citado 2 vezes nas páginas 52 e 54.

Practical business forecasting. Blackwell Publishing Ltd, jan 2003. Citado 2 vezes nas páginas 54 e 55.

FAN, J.; YAO, Q. Nonlinear time series: nonparametric and parametric methods. Springer Science & Business Media, 2008. Citado na página 30.

FARAWAY, J.; CHATFIELD, C. Time series forecasting with neural networks: A comparative study using the airline data. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, [Wiley, Royal Statistical Society], v. 47, n. 2, p. 231–250, 1998. ISSN 00359254, 14679876. Disponível em: <<http://www.jstor.org/stable/2988352>>. Citado na página 54.

GOMES, D. T. *Redes Neurais Recorrentes Para Previsão de Séries Temporais de Memórias Curta e Longa*. Dissertação (Mestrado) — Unicamp, 2005. Citado na página 33.

GRANGER, C.; LIN, J.-L. USING THE MUTUAL INFORMATION COEFFICIENT TO IDENTIFY LAGS IN NONLINEAR MODELS. *Journal of Time Series Analysis*, Wiley, v. 15, n. 4, p. 371–384, jul 1994. Citado na página 55.

GRANGER, C.; TERÄSVIRTA, T. Modelling nonlinear economic relationships. *Oxford University Press*, 1993. Citado na página 31.

GRANGER, C. W. J.; HALLMAN, J. NONLINEAR TRANSFORMATIONS OF INTEGRATED TIME SERIES. *Journal of Time Series Analysis*, Wiley, v. 12, n. 3, p. 207–224, may 1991. Citado 2 vezes nas páginas 45 e 47.

HATANAKA, M. *Time-Series-Based Econometrics 'Unit Roots and Cointegration'*. OUP Oxford, 1996. ISBN 0198773536. Disponível em: <[https://www.ebook.de/de/product/2761380/michio\\_hatanaka\\_time\\_series\\_based\\_econometrics\\_unit\\_roots\\_and\\_cointegration.html](https://www.ebook.de/de/product/2761380/michio_hatanaka_time_series_based_econometrics_unit_roots_and_cointegration.html)>. Citado na página 46.

HAUSSER, J.; STRIMMER, K. Entropy inference and the james-stein estimator, with application to nonlinear gene association networks. 2009. Citado na página 55.

HAYKIN, S. S. Neural networks and learning machines. *Pearson Education*, 2009. Citado 4 vezes nas páginas 40, 56, 64 e 65.

HORNIK, K. Some new results on neural network approximation. *Neural Networks*, Elsevier BV, v. 6, n. 8, p. 1069–1072, jan 1993. Citado 2 vezes nas páginas 39 e 44.

HYNDMAN, R.; ATHANASOPOULOS, G. *Forecasting: Principles and Practice*. 2nd. ed. Australia: OTexts, 2018. Citado 3 vezes nas páginas 70, 71 e 75.

KREIB, J.-P.; NEUHAUS, G. *Einführung in die Zeitreihenanalyse (Statistik und ihre Anwendungen)*. [S.l.]: Springer Berlin Heidelberg, 2006. ISBN 3540256288. Citado na página 46.

KUAN, C.-M. Artificial neural networks. Springer, jan. 2018. Citado na página 31.

LEE, T.-H.; WHITE, H.; GRANGER, C. W. Testing for neglected nonlinearity in time series models. *Journal of Econometrics*, Elsevier BV, v. 56, n. 3, p. 269–290, apr 1993. Citado 2 vezes nas páginas 49 e 69.

LEVENBERG, K. A method for the solution of certain non-linear problems in least squares. *Quarterly of Applied Mathematics*, American Mathematical Society (AMS), v. 2, n. 2, p. 164–168, jul 1944. Citado na página 64.

- LÜTKEPOHL, H.; TSCHERNIG, R. Nichtparametrische verfahren zur analyse und prognose von finanzmarktdaten. Springer, p. 145–171, 1996. Citado 2 vezes nas páginas 30 e 45.
- MARQUARDT, D. W. An algorithm for least-squares estimation of nonlinear parameters. *Journal of the Society for Industrial and Applied Mathematics*, Society for Industrial & Applied Mathematics (SIAM), v. 11, n. 2, p. 431–441, jun 1963. Citado na página 64.
- MEDEIROS, M. C.; TERÄSVIRTA, T.; RECH, G. Building neural network models for time series: a statistical approach. *Journal of Forecasting*, Wiley Online Library, v. 25, n. 1, p. 49–75, 2006. Citado 3 vezes nas páginas 30, 53 e 68.
- METZ, R. Zeitreihenanalyse. VS Verlag für Sozialwissenschaften, p. 1053–1090, 2010. Citado na página 54.
- MORETTIN, P. A. Econometria financeira: um curso em séries temporais financeiras. 2008. Citado 2 vezes nas páginas 25 e 29.
- OPPER, M.; WINTHER, O. A bayesian approach to on-line learning. Cambridge University Press, United Kingdom, p. 363–378, 1999. Citado na página 56.
- RUMELHART, D. E.; MCCLELLAND, J. L.; GROUP, P. R. Parallel distributed processing, vol. 1: Foundations. A Bradford Book, 1987. Disponível em: <<https://www.amazon.com/Parallel-Distributed-Processing-Vol-Foundations/dp/026268053X?SubscriptionId=AKIAIOBINVZYXZQZ2U3A&tag=chimbori05-20&linkCode=xm2&camp=2025&creative=165953&creativeASIN=026268053X>>. Citado na página 63.
- SCHRAUDOLPH, N. N. Fast curvature matrix-vector products for second-order gradient descent. *Neural Computation*, MIT Press - Journals, v. 14, n. 7, p. 1723–1738, jul 2002. Citado na página 56.
- SCHWARZ, G. Estimating the dimension of a model. *The Annals of Statistics*, Institute of Mathematical Statistics, v. 6, n. 2, p. 461–464, mar 1978. Citado na página 54.
- SIEGMUND-SCHULTZE, R. Der beweis des weierstraßschen approximationssatzes 1885 vor dem hintergrund der entwicklung der fourieranalysis. *Historia Mathematica*, Elsevier BV, v. 15, n. 4, p. 299–310, nov 1988. Citado na página 68.
- Software R. *R: A Language and Environment for Statistical Computing*. Vienna, Austria, 2019. Disponível em: <<https://www.R-project.org/>>. Citado na página 73.
- STEURER, E. Prognose von 15 zeitreihen der DGOR mit neuronalen netzen. *OR Spektrum*, Springer Science and Business Media LLC, v. 18, n. 2, p. 117–125, jun 1996. Citado na página 49.
- STONE, M. H. The generalized weierstrass approximation theorem. *Mathematics Magazine*, Informa UK Limited, v. 21, n. 4, p. 167, mar 1948. Citado na página 68.
- TRAPLETTI, A.; LEISCH, F.; HORNIK, K. Stationary and integrated autoregressive neural network processes. *Neural Computation*, MIT Press - Journals, v. 12, n. 10, p. 2427–2450, oct 2000. Citado 4 vezes nas páginas 45, 46, 47 e 49.

VOGL, T. P. et al. Accelerating the convergence of the back-propagation method. *Biological Cybernetics*, Springer Nature, v. 59, n. 4-5, p. 257–263, sep 1988. Citado na página 63.

WHITE. Economic prediction using neural networks: the case of IBM daily stock returns. In: *IEEE International Conference on Neural Networks*. [S.l.]: IEEE, 1988. Citado na página 58.

WHITE. An additional hidden unit test for neglected nonlinearity in multilayer feedforward networks. In: *International Joint Conference on Neural Networks*. [S.l.]: IEEE, 1989. Citado 2 vezes nas páginas 51 e 52.

WHITE, H.; GALLANT, A. Artificial neural networks: Approximation and learning theory. Blackwell, 1992. Disponível em: <<https://books.google.com.br/books?id=Xwd0QgAACAAJ>>. Citado na página 53.

WIDMANN, G. Künstliche neuronale netze und ihre beziehungen zur statistik. Lang, Peter GmbH, 2001. Disponível em: <[https://www.ebook.de/de/product/17231667/gabriele\\_widmann\\_kuenstliche\\_neuronale\\_netze\\_und\\_ihre\\_beziehungen\\_zur\\_statistik.html](https://www.ebook.de/de/product/17231667/gabriele_widmann_kuenstliche_neuronale_netze_und_ihre_beziehungen_zur_statistik.html)>. Citado 3 vezes nas páginas 56, 63 e 64.

WOLD, H. A study in the analysis of stationary time series. 1938. Citado na página 30.



# ANEXO A – Estimativas dos parâmetros dos modelos:

## A.1 AR-NN(5) com 1 camada e 6 neurônios ocultos.

Tabela 6 – Estimativas dos Parâmetros.

$\alpha_0 =$	10,8808	$\gamma_{01} =$	1,59	$\gamma_{02} =$	0,09	$\gamma_{03} =$	0,22	$\gamma_{04} =$	0,53	$\gamma_{05} =$	0,06	$\gamma_{06} =$	1,59
$\alpha_1 =$	0,5041	$\gamma_{11} =$	0,22	$\gamma_{12} =$	0,33	$\gamma_{13} =$	0,33	$\gamma_{14} =$	0,22	$\gamma_{15} =$	0,41	$\gamma_{16} =$	1,19
$\alpha_2 =$	0,2752	$\gamma_{21} =$	1,03	$\gamma_{22} =$	0,45	$\gamma_{23} =$	0,75	$\gamma_{24} =$	1,22	$\gamma_{25} =$	0,79	$\gamma_{26} =$	0,54
$\alpha_3 =$	-0,2599	$\gamma_{31} =$	1,62	$\gamma_{32} =$	0,10	$\gamma_{33} =$	1,62	$\gamma_{34} =$	0,05	$\gamma_{35} =$	0,58	$\gamma_{36} =$	1,08
$\alpha_4 =$	0,9693	$\gamma_{41} =$	0,16	$\gamma_{42} =$	0,15	$\gamma_{43} =$	0,54	$\gamma_{44} =$	0,45	$\gamma_{45} =$	0,67	$\gamma_{46} =$	1,62
$\alpha_5 =$	0,9906	$\gamma_{51} =$	1,71	$\gamma_{52} =$	0,20	$\gamma_{53} =$	0,14	$\gamma_{54} =$	1,46	$\gamma_{55} =$	0,13	$\gamma_{56} =$	1,04
		$\beta_1 =$	0,92	$\beta_2 =$	0,98	$\beta_3 =$	0,47	$\beta_4 =$	1,43	$\beta_5 =$	1,04	$\beta_6 =$	0,52

Fonte: Autor.

## A.2 AR-NN(17) com uma camada e dez neurônios ocultos.

Tabela 7 – Estimativas dos Parâmetros.

$\alpha_0 =$	0,61	$\gamma_{0,1} =$	1,38	$\gamma_{0,2} =$	1,27	$\gamma_{0,3} =$	1,67	$\gamma_{0,4} =$	1,68	$\gamma_{0,5} =$	0,14
$\alpha_1 =$	0,87	$\gamma_{1,1} =$	0,22	$\gamma_{1,2} =$	0,83	$\gamma_{1,3} =$	0,15	$\gamma_{1,4} =$	0,13	$\gamma_{1,5} =$	0,87
$\alpha_2 =$	-0,55	$\gamma_{2,1} =$	0,91	$\gamma_{2,2} =$	0,04	$\gamma_{2,3} =$	0,82	$\gamma_{2,4} =$	0,01	$\gamma_{2,5} =$	0,54
$\alpha_3 =$	0,73	$\gamma_{3,1} =$	0,50	$\gamma_{3,2} =$	0,91	$\gamma_{3,3} =$	0,73	$\gamma_{3,4} =$	0,62	$\gamma_{3,5} =$	0,57
$\alpha_4 =$	0,84	$\gamma_{4,1} =$	0,09	$\gamma_{4,2} =$	0,02	$\gamma_{4,3} =$	0,51	$\gamma_{4,4} =$	0,59	$\gamma_{4,5} =$	0,26
$\alpha_5 =$	0,73	$\gamma_{5,1} =$	0,22	$\gamma_{5,2} =$	0,70	$\gamma_{5,3} =$	0,16	$\gamma_{5,4} =$	0,28	$\gamma_{5,5} =$	0,22
$\alpha_6 =$	-0,42	$\gamma_{6,1} =$	0,32	$\gamma_{6,2} =$	0,30	$\gamma_{6,3} =$	0,62	$\gamma_{6,4} =$	0,27	$\gamma_{6,5} =$	0,84
$\alpha_7 =$	0,87	$\gamma_{7,1} =$	0,54	$\gamma_{7,2} =$	0,03	$\gamma_{7,3} =$	0,67	$\gamma_{7,4} =$	0,82	$\gamma_{7,5} =$	0,37
$\alpha_8 =$	0,15	$\gamma_{8,1} =$	0,20	$\gamma_{8,2} =$	0,06	$\gamma_{8,3} =$	0,58	$\gamma_{8,4} =$	0,19	$\gamma_{8,5} =$	0,69
$\alpha_9 =$	0,05	$\gamma_{9,1} =$	0,70	$\gamma_{9,2} =$	0,56	$\gamma_{9,3} =$	0,32	$\gamma_{9,4} =$	0,99	$\gamma_{9,5} =$	0,35
$\alpha_{10} =$	-0,73	$\gamma_{10,1} =$	0,12	$\gamma_{10,2} =$	0,26	$\gamma_{10,3} =$	0,81	$\gamma_{10,4} =$	0,42	$\gamma_{10,5} =$	0,07
$\alpha_{11} =$	0,92	$\gamma_{11,1} =$	0,82	$\gamma_{11,2} =$	0,58	$\gamma_{11,3} =$	0,43	$\gamma_{11,4} =$	0,15	$\gamma_{11,5} =$	0,53
$\alpha_{12} =$	0,12	$\gamma_{12,1} =$	0,78	$\gamma_{12,2} =$	0,49	$\gamma_{12,3} =$	0,78	$\gamma_{12,4} =$	0,59	$\gamma_{12,5} =$	0,04
$\alpha_{13} =$	0,63	$\gamma_{13,1} =$	0,24	$\gamma_{13,2} =$	0,84	$\gamma_{13,3} =$	0,22	$\gamma_{13,4} =$	0,98	$\gamma_{13,5} =$	0,23
$\alpha_{14} =$	-0,64	$\gamma_{14,1} =$	0,09	$\gamma_{14,2} =$	0,47	$\gamma_{14,3} =$	0,08	$\gamma_{14,4} =$	0,29	$\gamma_{14,5} =$	0,19
$\alpha_{15} =$	0,15	$\gamma_{15,1} =$	0,69	$\gamma_{15,2} =$	0,89	$\gamma_{15,3} =$	0,33	$\gamma_{15,4} =$	0,46	$\gamma_{15,5} =$	0,56
$\alpha_{16} =$	0,64	$\gamma_{16,1} =$	0,86	$\gamma_{16,2} =$	0,19	$\gamma_{16,3} =$	0,21	$\gamma_{16,4} =$	0,20	$\gamma_{16,5} =$	0,16
$\alpha_{17} =$	0,35	$\gamma_{17,1} =$	0,54	$\gamma_{17,2} =$	0,67	$\gamma_{17,3} =$	0,21	$\gamma_{17,4} =$	0,50	$\gamma_{17,5} =$	0,23
		$\beta_1 =$	0,43	$\beta_2 =$	0,33	$\beta_3 =$	0,72	$\beta_4 =$	0,56	$\beta_5 =$	0,09

Fonte: Autor.

Tabela 8 – Estimativas dos Parâmetros.

$\gamma_{0;6} =$	2,38	$\gamma_{0;7} =$	1,05	$\gamma_{0;8} =$	-0,45	$\gamma_{0;9} =$	0,29	$\gamma_{0;10} =$	1,55
$\gamma_{1;6} =$	0,44	$\gamma_{1;7} =$	0,85	$\gamma_{1;8} =$	0,30	$\gamma_{1;9} =$	0,74	$\gamma_{1;10} =$	0,88
$\gamma_{2;6} =$	0,28	$\gamma_{2;7} =$	0,20	$\gamma_{2;8} =$	0,32	$\gamma_{2;9} =$	0,29	$\gamma_{2;10} =$	0,10
$\gamma_{3;6} =$	0,20	$\gamma_{3;7} =$	0,42	$\gamma_{3;8} =$	0,67	$\gamma_{3;9} =$	0,60	$\gamma_{3;10} =$	0,56
$\gamma_{4;6} =$	0,63	$\gamma_{4;7} =$	0,50	$\gamma_{4;8} =$	0,69	$\gamma_{4;9} =$	0,05	$\gamma_{4;10} =$	0,40
$\gamma_{5;6} =$	0,04	$\gamma_{5;7} =$	0,10	$\gamma_{5;8} =$	0,17	$\gamma_{5;9} =$	0,19	$\gamma_{5;10} =$	0,70
$\gamma_{6;6} =$	0,32	$\gamma_{6;7} =$	0,13	$\gamma_{6;8} =$	0,04	$\gamma_{6;9} =$	0,79	$\gamma_{6;10} =$	0,18
$\gamma_{7;6} =$	0,42	$\gamma_{7;7} =$	0,61	$\gamma_{7;8} =$	0,68	$\gamma_{7;9} =$	0,65	$\gamma_{7;10} =$	0,44
$\gamma_{8;6} =$	0,55	$\gamma_{8;7} =$	0,33	$\gamma_{8;8} =$	0,31	$\gamma_{8;9} =$	0,61	$\gamma_{8;10} =$	0,70
$\gamma_{9;6} =$	0,17	$\gamma_{9;7} =$	0,70	$\gamma_{9;8} =$	0,44	$\gamma_{9;9} =$	0,44	$\gamma_{9;10} =$	0,74
$\gamma_{10;6} =$	0,22	$\gamma_{10;7} =$	0,15	$\gamma_{10;8} =$	0,64	$\gamma_{10;9} =$	0,22	$\gamma_{10;10} =$	0,20
$\gamma_{11;6} =$	0,09	$\gamma_{11;7} =$	0,41	$\gamma_{11;8} =$	0,14	$\gamma_{11;9} =$	0,30	$\gamma_{11;10} =$	0,28
$\gamma_{12;6} =$	0,14	$\gamma_{12;7} =$	0,72	$\gamma_{12;8} =$	0,19	$\gamma_{12;9} =$	0,84	$\gamma_{12;10} =$	0,21
$\gamma_{13;6} =$	0,89	$\gamma_{13;7} =$	0,33	$\gamma_{13;8} =$	0,49	$\gamma_{13;9} =$	0,07	$\gamma_{13;10} =$	0,19
$\gamma_{14;6} =$	0,02	$\gamma_{14;7} =$	0,92	$\gamma_{14;8} =$	0,50	$\gamma_{14;9} =$	0,31	$\gamma_{14;10} =$	0,22
$\gamma_{15;6} =$	0,16	$\gamma_{15;7} =$	0,77	$\gamma_{15;8} =$	0,91	$\gamma_{15;9} =$	0,16	$\gamma_{15;10} =$	0,24
$\gamma_{16;6} =$	0,57	$\gamma_{16;7} =$	0,45	$\gamma_{16;8} =$	0,31	$\gamma_{16;9} =$	0,15	$\gamma_{16;10} =$	0,41
$\gamma_{17;6} =$	0,51	$\gamma_{17;7} =$	0,80	$\gamma_{17;8} =$	0,31	$\gamma_{17;9} =$	0,63	$\gamma_{17;10} =$	0,31
$\beta_6 =$	0,53	$\beta_7 =$	0,66	$\beta_8 =$	0,64	$\beta_9 =$	0,44	$\beta_{10} =$	0,62

Fonte: Autor.