2
Fundamentos Teóricos

Este capítulo fornece um breve resumo dos fundamentos teóricos necessários ao entendimento do modelo proposto neste trabalho. O capítulo inicia com a Seção 2.1, onde comenta-se sobre processos estocásticos, incluindo a descrição do modelo estocástico periódico auto-regressivo de ordem p - PAR(p) [30], usado atualmente para gerar cenários de afluências no sistema de planejamento e operação do Sistema Interligado Nacional (SIN) [37]. A Seção 2.2 apresenta o método de amostragem Quase Monte Carlo, utilizado na geração de amostras de ruídos adicionados às saídas das redes neurais artificiais (RNAs) para compor as séries sintéticas geradas pelo Processo Estocástico Neural (PEN) - modelo proposto nesse trabalho. Alguns testes de hipóteses, utilizados na avaliação da aderência dos cenários gerados com a série histórica são apresentados na Seção 2.3. Encontra-se na Seção 2.4 os conceitos principais de Redes Neurais Artificiais, base do modelo desta tese. Para finalizar, a Seção 2.5 fornece os conceitos básicos sobre Algoritmos Genéticos, base do modelo desenvolvido em [60], usado para calcular o custo do planejamento da operação energética de médio prazo usando os cenários gerados pelo modelo proposto nessa tese - o Processo Estocástico Neural.

2.1
Processos Estocásticos

Um processo estocástico (PE) é uma coleção de variáveis aleatórias definidas num mesmo espaço de probabilidades, indexadas por elementos t pertencentes a determinado intervalo temporal, que descreve a evolução, no tempo ou no espaço, de algum processo físico [51].

Portanto, um PE é um modelo matemático caracterizado por uma coleção de variáveis aleatórias ordenadas, no tempo ou no espaço, e definidas em um conjunto, contínuo ou discreto, que descreve a evolução de algum fenômeno natural com características aleatórias [47].

Geralmente representa-se um PE por $\{Z(t) : t \in T\}$ onde t representa o instante de tempo, $Z(t)$ é uma variável aleatória chamada de estado do processo

no instante t e T é o conjunto de índices denominado espaço paramétrico do PE. Se o conjunto T é um intervalo (finito ou infinito) de números reais, diz-se que o PE $\{Z(t) : t \in T\}$ é um processo contínuo. Ao contrário, se T é um conjunto finito ou contável, como por exemplo $T = \{1, 2, 3, \ldots\}$ ou $T = \{1, 9, 43, 279\}$, diz-se que o PE é um processo discreto [7].

O espaço de estados de um PE é o conjunto de todos os possíveis valores da variável $Z(t)$, que também pode ser discreto ou contínuo. Portanto, a combinação dos tipos de espaço paramétrico e de espaço de estados conduz a quatro classes de processos estocásticos [7]:

1. T discreto e $Z(t)$ discreto: processo discreto com espaço de estados discreto;

2. T contínuo e $Z(t)$ discreto: processo contínuo com espaço de estados discreto;

3. T discreto e $Z(t)$ contínuo: processo discreto com espaço de estados contínuo;

4. T contínuo e $Z(t)$ contínuo: processo contínuo com espaço de estados contínuo;

O conceito de processo estocástico proporciona a análise probabilística de séries temporais. Tem-se que uma série temporal é um conjunto de observações de uma dada variável, ordenado segundo o parâmetro tempo, geralmente em intervalos eqüidistantes [62]. Assim, uma série temporal pode ser considerada uma realização de um PE, isto é, uma possível trajetória do processo. Portanto, PE é um processo gerador de dados cuja série temporal é uma realização amostral dentre todas as séries possíveis de serem geradas por este modelo.

A Figura 2.1 ilustra a seqüência do estudo de uma série temporal, onde dado um PE (fenômeno real) retira-se uma série temporal (amostra finita de observações equiespaçadas no tempo) e através da análise de séries temporais (estudo da amostra) identifica-se um modelo cujo objetivo é inferir sobre o comportamento da realidade.

A partir da análise de séries temporais é possível obter recursos para a especificação de um modelo adequado para a série. A partir da expressão matemática do modelo, pode-se obter as fórmulas para os seus momentos como média, variância, entre outros. Portanto, uma maneira de descrever um PE é através dos momentos das variáveis aleatórias, em especial, a média, a variância e autocovariância do processo [7]. A média e a variância de um PE discreto são

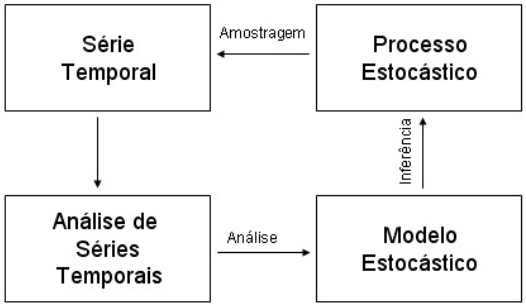

Figura 2.1: Processo Estocástico e Série Temporal [62]

funções do instante de tempo t, definidas, respectivamente, pelas equações 2-1 e 2-2.

$$\mu(t) = E[Z(t)], \tag{2-1}$$

$$\sigma^2(t) = Var[Z(t)] = E\{[Z(t) - \mu(t)]^2\}. \tag{2-2}$$

sendo $E[\cdot]$ o valor esperado e $Z(t)$ o estado do processo no instante t.

A autocovariância de um PE discreto é uma função definida, entre os instantes t_1 e t_2, por:

$$\gamma(t_1, t_2) = Cov[Z(t_1), Z(t_2)] = E\{[Z(t_1) - \mu(t_1)] \cdot [Z(t_2) - \mu(t_2)]\}.$$

Isto é, a autocovarância de um PE discreto é apenas a covariância entre instantes de tempo diferentes. Logo, a variância do PE é apenas um caso particular da autocovariância, onde $t_1 = t_2$.

Momentos de ordem mais alta podem ser definidos de maneira semelhante, contudo são pouco usados na prática. As definições dos momentos para um PE contínuo são análogas às de um PE discreto.

Estas características de processos estocásticos estão intimamente ligadas à noção de estacionariedade de um processo [7]. Diz-se que um processo é estacionário se não existem mudanças nas suas características, isto é, se ele for invariante em relação ao tempo. Segundo a estacionariedade, um processo pode ser classificado em:

Processo Estritamente Estacionário: é quando suas estatísticas não são afetadas por variações devido à escolha da origem dos tempos, ou seja, a distribuição de probabilidade conjunta [1] não muda ao se deslocar no tempo ou no espaço. Dessa forma, a distribuição de probabilidade

[1]Sejam X e Y duas variáveis aleatórias definidas no mesmo espaço de probabilidade. Tem-se que a Distribuição de Probabilidade Conjunta de X e Y é a distribuição de probabilidades de sua ocorrência simultânea que pode ser representada pela função $F(x, y) = P(X \leq x, Y \leq y)$ para qualquer par de valores (x, y) [40].

conjunta $P\{Z(t_1) = z1, Z(t_2) = z_2, \ldots, Z(t_n) = z_n\}$ é a mesma que $P\{Z(t_1 + k) = z_1, Z(t_2 + k) = z_2, \ldots, Z(t_n + k) = z_n\}$, para qualquer t_i, k e n. A média e a variância do processo são constantes para todo instante de tempo $t \in T$ e a função de autocovariância só depende da defasagem (ou *lag*) $t_{i+k} - t_i$. A média do processo é dada pela equação 2-3.

$$\mu = E[Z(t)] = E[Z(t + k)]. \tag{2-3}$$

A função de autocovariância é chamada de autocovariância de *lag* k e pode ser escrita como a equação 2-4.

$$\gamma(k) = E\{[Z(t) - \mu] \cdot [Z(t + k) - \mu]\} \tag{2-4}$$

Portanto, quando $k = 0$, tem-se a variância constante do processo:

$$\sigma^2 = \gamma(0) = E\{[Z(t) - \mu]^2\}.$$

Processo Fracamente Estacionário: a condição de estacionariedade é mais fraca porque impõe-se condições apenas sobre os dois primeiros momentos, que não garantem condições sobre a estacionariedade da função de probabilidade. Assim, a média do processo é constante e sua autocovariância depende apenas do *lag* k (diferença em valor absoluto de $t_{i+k} - t_i$).

O conceito de estacionariedade é muito utilizado na análise de séries temporais. Na prática, as séries temporais podem ser classificadas em 3 tipos: aquelas que exibem propriedades de estacionaridade em longos períodos (estritamente estacionárias); aquelas que possuem uma razoável estacionaridade em períodos curtos (fracamente estacionárias) e aquelas que são obviamente não estacionárias, no sentido que suas propriedades estão continuamente mudando com o tempo. Alguns métodos estatísticos usados em séries temporais, restringem o uso à séries temporais (fracamente) estacionárias. Para isso, eles tratam a não-estacionaridade de séries temporais, através de técnicas utilizadas para remover ou filtrar a parte não-estacionária, trabalhando apenas com a parte estacionária [47].

Existem alguns processos estocásticos que são muito utilizados na especificação de modelos para séries temporais. Um exemplo de PE básico, que pode ser usado na construção de processos mais complicados é o ruído branco ou seqüência aleatória.

Ruído Branco

Um PE discreto é chamado de ruído branco se ele for um processo puramente aleatório, ou seja, se os $Z(t)$ constituem uma seqüência de variáveis aleatórias independentes e identicamente distribuídas.

Um ruído branco tem média e variância constantes e função de autocorrelação nula em todos os lags, ou seja, são descorrelatados. Usualmente apresentam distribuição normal de média zero e de variância um, $Z(t) \sim N(0,1)$.

Os processos ruído branco aparecem na construção de outros processos mais complexos, como, por exemplo, os modelos de Box e Jenkins [9].

2.1.1
Modelos de Box e Jenkins

Os modelos ARIMA de Box e Jenkins [9] são modelos estatísticos lineares, muito utilizados na análise de séries temporais, que usam as correlações entre as observações em diversos instantes. A abreviação ARIMA em inglês representa *Auto-Regressive Integrated Moving Average*, que significa: modelo auto-regressivo, integrado de médias móveis.

O fundamento teórico da técnica é gerar um processo (fracamente) estacionário passando um ruído branco por um filtro linear de memória infinita. Porém, na prática, o problema resolvido é o inverso, ou seja, através de uma realização $Z(t)$ de tamanho n do processo gerador da série, tenta-se chegar ao ruído branco $a(t)$ pela passagem sucessiva de $Z(t)$ pelo filtro linear, como ilustra a Figura 2.2.

Figura 2.2: Diagrama Operacional do modelo Box e Jenkins

A modelagem ARIMA consiste em encontrar a estrutura de autocorrelação presente na série temporal $Z(t)$ e a autocorrelação existente entre os termos do erro aleatório $a(t)$, através da equação 2-5:

$$Z(t) = \Psi \cdot a(t) \tag{2-5}$$

onde Ψ é a representação de infinitos parâmetros, ou seja,

$$Z(t) = a(t) - \psi_1 \cdot a(t-1) - \psi_2 \cdot a(t-2) - ...$$

Entretanto, esses parâmetros infinitos podem ser substituídos pela razão de dois parâmetros finitos: $\Psi = \frac{\Theta}{\Phi}$, onde Φ são os parâmetros da parte auto-regressiva (AR) da modelagem da série e Θ são os parâmetros da parte da modelagem de autocorrelação entre os termos do erro do modelo, chamado de médias móveis (MA).

O operador de atraso (*backward shift operator*), representado por B, é definido por:

$$B \cdot Z(t) = Z(t-1)$$

e pode apresentar potências, onde $B^k \cdot Z(t) = Z(t-k)$.

Assim, a equação 2-5 pode ser representada pela equação 2-6:

$$\Phi(B) \cdot Z(t) = \Theta(B) \cdot a(t) \tag{2-6}$$

Seja p o número de termos auto-regressivos (AR) e q o número de termos médias móveis (MA), então, dizemos que os termos p e q são os graus dos polinômios Φ e Θ, respectivamente. De acordo com Box e Jenkins [9], o modelo dado pela equação 2-6 é denominado ARMA(p, q).

A condição essencial para a aplicação deste modelo é que a série a ser modelada seja (fracamente) estacionária. Caso contrário, cria-se uma nova série realizando sucessivas diferenças no conjunto original de dados.

Por exemplo, seja a seqüência $X_1, X_2, ..., X_t$ uma série temporal não estacionária. Considere as sucessivas diferenças: $Z_1 = X_2 - X_1$, $Z_2 = X_3 - X_2$, $Z_3 = X_4 - X_3$. Se Z_t for uma série estacionária, ela pode ser modelada por um modelo ARMA(p, q).

Matematicamente, essas diferenças são representadas pelo operador diferença ∇^d, onde o índice d diz respeito ao número de diferenças que tornam a série original estacionária. Neste caso, o modelo de Box & Jenkins é referenciado como ARIMA(p, d, q).

As ordens p e q do modelo são determinadas pelas funções: função de autocorrelação (FAC) e função de autocorrelação parcial (FACP). Para isso, basta observar a forma de decréscimo da série e em que *lag* acontece um corte brusco, isto é, onde a correlação é mais significativa.

Identificado o número de diferenças d e as ordens p e q do modelo, é preciso obter as estimativas dos seus parâmetros Φ e Θ, que pode ser feita utilizando-se o método dos mínimos quadrados ou máxima verosimilhança [7]. Depois da etapa de estimação é necessário realizar a verificação ou diagnóstico que compreende: a verificação dos parâmetros estimados e a análise dos resíduos, que pela definição do modelo devem representar ruídos brancos.

O melhor modelo a ser usado para modelar uma série temporal deve ser

parcimonioso, isto é, utilizar o menor conjunto de parâmetros possível para ajustar à série de dados observados (série histórica).

Para facilitar a compreensão, esses modelos podem ser separados em dois modelos complementares: os modelos de médias móveis (MA) e os modelos auto-regressivos (AR). Existe um princípio de dualidade entre os modelos do tipo MA e AR: um MA(q) de ordem finita corresponde ao um AR(p) de ordem infinita e vice-versa [9].

Modelos Auto-regressivos - AR(p)

Os modelos auto-regressivos AR(p) modelam a variável dependente $Z(t)$ usando os p últimos valores da série estacionária ($Z(t-1), Z(t-2), \ldots Z(t-p)$), isto é, a série $Z(t)$ é escrita a partir dos seus valores passados. O parâmetro p é chamado de ordem do modelo.

Os modelos AR(p) são representados pela equação 2-7:

$$Z(t) = \phi_1 \cdot Z(t-1) + \phi_2 \cdot Z(t-2) + ... + \phi_p \cdot Z(t-p) + a(t) \qquad (2\text{-}7)$$

cujos elementos são os mesmos descritos na equação 2-5.

Neste modelo, a FAC apresenta um comportamento de exponencial amortecida ou de senóide amortecida, enquanto a FACP se anula após o *lag* p. Portanto, a autocorrelação parcial se mantém significativa nos p períodos de defasagem da variável dependente $Z(t)$.

Esse modelo pode ser genericamente representado por modelos ARIMA(p, 0, 0).

Modelos Médias Móveis - MA(q)

Os modelos médias móveis MA(q) são construídos a partir da constatação de que cada observação $Z(t)$ é gerada por uma média ponderada dos erros aleatórios de q períodos passados. O parâmetro q indica quantos valores passados do erro são usados como regressores.

Os modelos MA(q) são representados pela equação 2-8:

$$Z(t) = a(t) + \theta_1 \cdot a(t-1) + \theta_2 \cdot a(t-2) + ... + \theta_q \cdot a(t-q) \qquad (2\text{-}8)$$

cujos elementos são os mesmos descritos na equação 2-5.

Neste modelo, a FAC se anula bruscamente após o *lag* q, enquanto a FACP apresenta um comportamento de exponencial amortecida ou de senóide amortecida.

Esse modelo pode ser genericamente representado por modelos ARIMA(0, 0, q).

Modelos ARMA(p, q)

Um modelo ARMA(p, q) é um processo misto que combina num mesmo modelo a parte AR e a parte MA e são representados pela equação 2-9:

$$\begin{aligned} Z(t) = \phi_1 \cdot Z(t-1) + \phi_2 \cdot Z(t-2) + \ldots + \phi_p \cdot Z(t-p) \\ +a(t) + \theta_1 \cdot a(t-1) + \theta_2 \cdot a(t-2) + \ldots + \theta_q \cdot a(t-q) \end{aligned} \tag{2-9}$$

cujos elementos são os mesmos descritos na equação 2-5.

Neste modelo, a FAC apresenta um comportamento que é a forma da FAC de um modelo AR, enquanto a FACP apresenta um comportamento que é a forma da FACP de um modelo MA. Portanto, se uma série temporal estacionária produz FAC e FACP fora do padrão de processos puros, ela será considerada gerada por um processo ARMA(p, q) misto.

A identificação das ordens p e q de processos mistos é realizada através de critérios de informação, que selecionam entre os modelos ARMA(p, q) aquele, de menor complexidade, que consegue minimizar, simultaneamente, a discrepância entre os modelos e entre os dados. Utiliza-se sempre o menor número de parâmetros possíveis - princípio da parcimônia. Os critérios mais conhecidos são critério de Akaike (AIC) e *Bayesian Information Criterio*(BIC) [7].

Esse modelo pode ser genericamente representado por modelos ARIMA(p, 0, q).

Modelos SARIMA(p, d, q) (P, D, Q)

Um modelo SARIMA(p, d, q) (P, D, Q) é um processo que inclui componentes de sazonalidade no modelo. Sazonalidades são flutuações que necessariamente se relacionam com as estações do ano. Para descrever tais componentes sazonais existem estimadores específicos para esta finalidade com suas respectivas partes AR(P) e MA(Q) sazonais. As letras maiúsculas simbolizam a ordem sazonal do modelo que é identificada através da observação dos *lags* sazonais [9].

A identificação das ordens do modelo SARIMA é feita através das funções FAC e FACP, separando a informação não-sazonal (p, d, q) da informação sazonal (P, D, Q) do modelo [47].

2.1.2
Modelo Periódico Auto-regressivo de ordem p - PAR(p)

Algumas séries históricas exibem uma estrutura de autocorrelação que depende além do intervalo de tempo entre as observações, também do período

observado [30, 37]. O modelo Periódico Auto-Regressivo (PAR) é uma generalização do modelo auto-regressivo (AR) e é amplamente utilizado na literatura de séries temporais para captar a correlação periódica, sendo sua principal característica a variação dos parâmetros de acordo com o período [30].

Seja Z_t um processo estocástico tal que o índice de tempo t seja dado por

$$t = (r-1) \cdot s + m$$

onde $r, s \in \mathbb{Z}$ e $m = 1 \ldots s$. Por exemplo, para dados mensais temos: m representa o mês, r representa o ano e $s = 12$.

Dessa forma, um modelo PAR(p) é obtido a partir da definição de um modelo AR para cada mês m do ano r [38]. Assim, p é um vetor cujos elementos fornecem a ordem de cada mês m:

$$p = (p_1, p_2, ..., p_s)$$

A equação 2-10 apresenta a definição do modelo PAR(p_m), como é usado no setor elétrico [39, 37]:

$$\left(\frac{Z_t - \mu_m}{\sigma_m}\right) = \sum_{i=1}^{p_m} \phi_i^m \cdot \left(\frac{Z_{t-i} - \mu_{m-i}}{\sigma_{m-i}}\right) + a_t \qquad (2\text{-}10)$$

onde,

Z_t**:** é uma série sazonal de período s;

s**:** é o número de períodos ($s = 12$ para séries mensais);

t**:** é o índice de tempo, $t = 1, 2, \ldots, s \cdot r$

r**:** é o número de anos;

m**:** é o período, $m = 1 \ldots s$;

μ_m**:** é a média histórica do período m;

σ_m**:** é o desvio padrão histórico do período m;

p_m**:** é a ordem do modelo auto-regressivo do período m;

ϕ_i^m**:** é i-ésimo coeficiente auto-regressivo do período m;

a_t**:** série de resíduos independentes com média zero e variância $\sigma_a^{2(m)}$.

A ordem p_m do modelo é obtida através da função de autocorrelação parcial (FACP), considerando o último *lag* k onde ela é significativa. Para

encontrar esse lag, considere $\rho^m(k)$ a correlação de *lag* k para o mês m, ou seja, a correlação entre Z_t e Z_{t-k}, definida pela equação 2-11 [37].

$$\rho^m(k) = E\left[\left(\frac{Z_t - \mu_m}{\sigma_m}\right) \cdot \left(\frac{Z_{t-k} - \mu_{m-k}}{\sigma_{m-k}}\right)\right] \tag{2-11}$$

O conjunto de funções de autocorrelação $\rho^m(k)$, com $m = 1, 2, ..., s$, descreve a estrutura de dependência temporal da série. Estas funções podem ser obtidas ao se multiplicar ambos os lados da equação 2-10 por

$$\left(\frac{Z_{t-k} - \mu_{m-k}}{\sigma_{m-k}}\right)$$

e tomar o seu valor esperado, obtendo-se a equação 2-12 para cada período:

$$E\left[\left(\frac{Z_t-\mu_m}{\sigma_m}\right) \cdot \left(\frac{Z_{t-k}-\mu_{m-k}}{\sigma_{m-k}}\right)\right] = \sum_{i=1}^{p_m} \phi_i^m \cdot E\left[\left(\frac{Z_{t-i}-\mu_{m-i}}{\sigma_{m-i}}\right) \cdot \left(\frac{Z_{t-k}-\mu_{m-k}}{\sigma_{m-k}}\right)\right] + E\left[a_t \cdot \left(\frac{Z_{t-k}-\mu_{m-k}}{\sigma_{m-k}}\right)\right] \tag{2-12}$$

Conhecidos os parâmetros do modelo PAR(p), as funções de autocorrelação $\rho^m(k)$ são obtidas através da equação 2-12 e podem ser demonstradas por uma combinação de decaimentos exponenciais e/ou senoidais. Isto implica que $\rho^m(k)$ tenda a zero a medida que k aumenta.

Por definição

$$E\left[a_t \cdot \left(\frac{Z_{t-k} - \mu_{m-k}}{\sigma_{m-k}}\right)\right] = 0$$

assim, a equação 2-12 pode ser reescrita pela equação 2-13:

$$\rho^m(k) = \sum_{i=1}^{p_m} \phi_i^m \cdot \rho^{m-i}(k-i) \tag{2-13}$$

Fixando m e variando o *lag* k ($k = 1, 2, ..., p_m$) na equação 2-13, obtém-se um conjunto de equações, denominado de equações de Yule-Walker [39]. Existe a relação $\rho_{-j}^{m-k} = \rho_j^{m-k+j}$ entre as correlações de diferentes meses, que permite substituir as autocorrelações com *lags* negativos pelas respectivas com *lags* positivos. Assim, pode-se escrever o sistema de forma mais simples, como ilustra a equação 2-14:

$$\begin{bmatrix} \phi_1^m \\ \phi_2^m \\ \vdots \\ \phi_{p_m}^m \end{bmatrix} = \begin{bmatrix} \rho_0^{m-1} & \rho_1^{m-1} & \rho_2^{m-1} & \cdots & \rho_{p_m-1}^{m-1} \\ \rho_1^{m-1} & \rho_0^{m-2} & \rho_1^{m-2} & \cdots & \rho_{p_m-2}^{m-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \rho_{p_m-1}^{m-1} & \rho_{p_m-2}^{m-2} & \rho_{p_m-3}^{m-3} & \cdots & \rho_0^{m-p_m} \end{bmatrix}^{-1} \cdot \begin{bmatrix} \rho_1^m \\ \rho_2^m \\ \vdots \\ \rho_{p_m}^m \end{bmatrix} \tag{2-14}$$

Seja ϕ_{kj} o j-ésimo parâmetro auto-regressivo de um processo de ordem k, então o último parâmetro deste processo é dado por ϕ_{kk} e as equações Yule-

Walker podem ser reescritas como mostra a equação 2-15:

$$\begin{bmatrix} \phi_{k1}^m \\ \phi_{k2}^m \\ \vdots \\ \phi_{kk}^m \end{bmatrix} = \begin{bmatrix} \rho_0^{m-1} & \rho_1^{m-1} & \rho_2^{m-1} & \cdots & \rho_{k-1}^{m-1} \\ \rho_1^{m-1} & \rho_0^{m-2} & \rho_1^{m-2} & \cdots & \rho_{k-2}^{m-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \rho_{k-1}^{m-1} & \rho_{k-2}^{m-2} & \rho_{k-3}^{m-3} & \cdots & \rho_0^{m-k} \end{bmatrix}^{-1} \cdot \begin{bmatrix} \rho_1^m \\ \rho_2^m \\ \vdots \\ \rho_k^m \end{bmatrix} \tag{2-15}$$

Ao conjunto de valores ϕ_{kk}^m, $k = 1, 2, \ldots$ denomina-se função de autocorrelação parcial do mês m, que é uma outra forma de representar a estrutura de dependência temporal da série [53]. Em um processo autoregressivo de ordem p_m, os termos ϕ_{kk}^m da função de autocorrelação parcial serão diferentes de zero para $k \leq p_m$ e iguais a zero para $k > p_m$.

Se a ordem p_m do modelo fosse previamente conhecida, os parâmetros do PAR(p_m) poderiam ser obtidos através da resolução do sistema da equação 2-14. No entanto, a identificação do modelo consiste em descobrir, por tentativas, a ordem p_m mais apropriada dos operadores auto-regressivos, para cada mês m. Na prática, aumenta-se a ordem do modelo variando p_m de 1 até k, sendo k o valor no qual o parâmetro ϕ_{kk}^m (obtido pela equação 2-15) não é mais significativo [37]. Assim, a ordem p_m é igual a $k-1$, último valor para o qual o parâmetro $\phi_{p_m p_m}^m$ foi significativo, e os coeficientes do modelo PAR($k-1$) são os elementos $\phi_{k-1,1}^m, \phi_{k-1,2}^m, \ldots, \phi_{k-1,k-1}^m$ obtidos pela resolução do sistema da equação 2-15. Esta forma de estimar os parâmetros do modelo é conhecida como método dos momentos [30].

A identificação da ordem do modelo PAR(p_m) também pode ser realizada por meio da análise do gráfico da função de autocorrelação parcial. Neste gráfico, os coeficientes de autocorrelação parcial para cada *lag* são apresentados e a ordem do modelo é identificada visualmente, sendo aquela na qual ocorreu o último valor significativo de $\phi_{p_m p_m}^m$. A determinação de que o valor é significativo pode ser feita através de um intervalo de confiança e, usualmente, considera-se o intervalo de confiança de 95% para $\phi_{p_m p_m}^m$ dado por $\frac{1.96}{n}$, sendo n o número de anos da série hidrológica observada [38].

Os parâmetros do modelo para o m-ésimo mês podem ser estimados de maneira independente dos parâmetros de qualquer outro período. Em [37] coloca-se que cada um dos m sistemas resultantes pode ser resolvido por máxima verossimilhança. Porém, em modelos AR também é possível resolver tais sistemas pelo método dos momentos, que é mais simples, por não exigir que o sistema seja otimizado, até que se encontre os parâmetros ótimos do modelo, e é tão eficaz quanto o método da máxima verossimilhança [30].

A metodologia para o ajuste de modelos estocásticos da família ARIMA

a séries temporais, sugerida por Box & Jenkins [9], pode ser estendida para modelos da família PAR(p) [37]. Essa metodologia é composta de 3 etapas:

Identificação do modelo: consiste em escolher a ordem do modelo, que na modelagem auto-regressiva periódica consiste em determinar o vetor p.

Estimação do modelo: consiste em obter as estimativas para os parâmetros do modelo.

Verificação do modelo: consiste em verificar através de testes estatísticos se o modelo estimado é adequado, verificando se é capaz de gerar ruídos brancos após a aplicação do filtro auto-regressivo. Caso contrário, retorna-se a primeira etapa até que os resultados sejam satisfatórios.

O esquema para a determinação do modelo PAR(p_m) para cada mês m consiste em:

1. Leitura dos dados da série histórica;
2. Cálculo da média e do desvio padrão para os 12 meses;
3. Cálculo da função de autocorrelação para os 12 meses;
4. Cálculo da função de autocorrelação parcial para os 12 meses - construção do sistema Yule-Walker;
5. Identificação da ordem do modelo AR de cada mês m;
6. Estimação dos parâmetros do modelo AR de cada mês m;
7. Cálculo dos resíduos ajustados;

A grande maioria dos testes, utilizados na etapa de verificação do modelo, se baseia numa análise detalhada dos resíduos ajustados a_t, que devem ser independentes com média zero e variância $\sigma_a^{2(m)}$. As estimativas de $\sigma_a^{2(m)}$ podem ser obtidas utilizando-se a expressão fornecida pela equação 2-16 [30, 53].

$$\sigma_a^{2(m)} = 1 - \phi_1^m \cdot \rho^m(1) - \phi_2^m \cdot \rho^m(2) - \ldots - \phi_{p_m}^m \cdot \rho^m(p_m) \qquad (2\text{-}16)$$

Se o modelo estimado for considerado adequado, isso significa que ele é capaz de gerar séries sintéticas, igualmente prováveis à série histórica [39].

2.2 Método de Amostragem Quase Monte Carlo

Uma das técnicas para se reduzir a variância das amostras é espalhá-las o mais uniformemente possível sobre o espaço amostral. Essa é a idéia básica do método de amostragem Quase Monte Carlo, também conhecido como seqüências de baixa discrepância.

A utilização de seqüências de baixa discrepância, ou seqüências quase aleatórias, auxilia a convergência de simulações estocásticas, como por exemplo a Simulação de Monte Carlo [27], exigindo um número menor de simulações para atingir a precisão desejada [65]. Essas seqüências são construídas de forma que, cada elemento adicionado à seqüência, fique o mais distante possível dos demais elementos pertencentes à mesma. Assim, os maiores espaços entre os elementos da seqüência são preenchidos e a aglomeração de grupos de elementos são evitadas.

Existem diversas formas de se obter seqüências quase aleatórias, mas os três tipos de seqüências mais conhecidos são: de Halton [26], de Sobol [61] e de Faure [19].

A seqüência de Halton utiliza uma base de números primos diferente para cada dimensão:

- para dimensão 1 usa base 2;
- para dimensão 2 usa base 3;
- para dimensão 3 usa base 5 e assim por diante.

Uma vez definida a base prima b a ser usada, o i-ésimo elemento da seqüência de Halton pode ser obtido através do seguinte procedimento:

1. decompor o número $i-1$ na base b escolhida;
2. somar os termos da decomposição divididos por potências crescentes de b.

Uma base mais elevada significa maior número de ciclos e conseqüentemente maior tempo de processamento para gerar os resultados. Além disso, as seqüências de Halton de dimensão elevada apresentam uma degradação na dispersão dos pontos, como ilustra a Figura 2.3. No primeiro caso, usou-se um espaço com dimensão 14x15, o que já apresenta um pouco de degradação na dispersão dos pontos, e, piora no segundo caso, onde a dimensão do espaço é 20x21.

A seqüência de Sobol segue a mesma idéia da seqüência de Halton, todavia é utilizada uma mesma base para as dimensões ($b = 2$). Os elementos

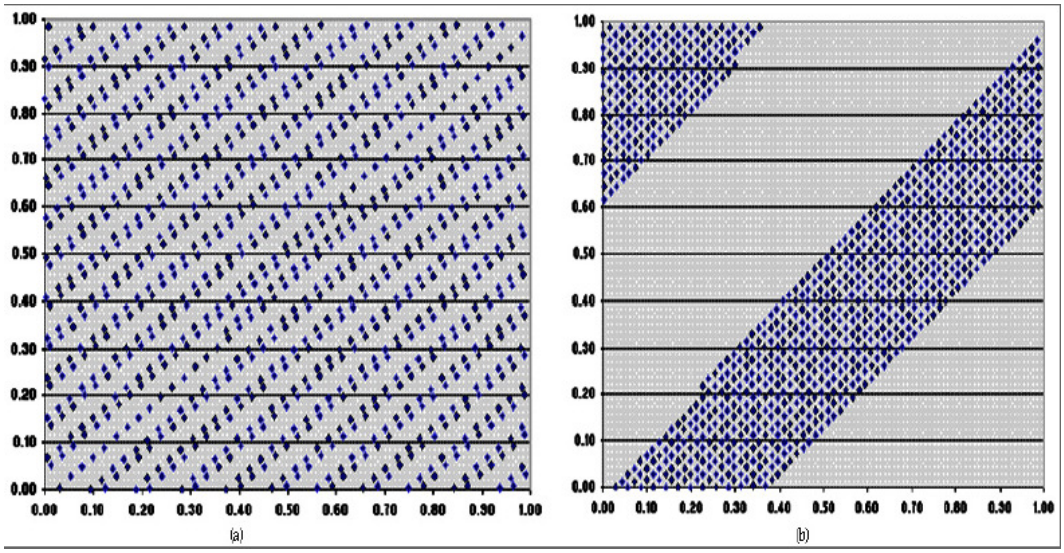

Figura 2.3: Seqüências de Halton: (a) dimensão 14x15 (b) dimensão 20x21

da seqüência são re-ordenados por técnicas de permutação aleatória, dentro de cada dimensão. A Figura 2.4 ilustra a comparação entre pontos gerados por uma seqüência pseudo-aleatória bidimensional simples e pontos gerados por uma seqüência Quase Monte Carlo de Sobol bidimensional. Em ambas as seqüências foram gerados 1000 pontos amostrais.

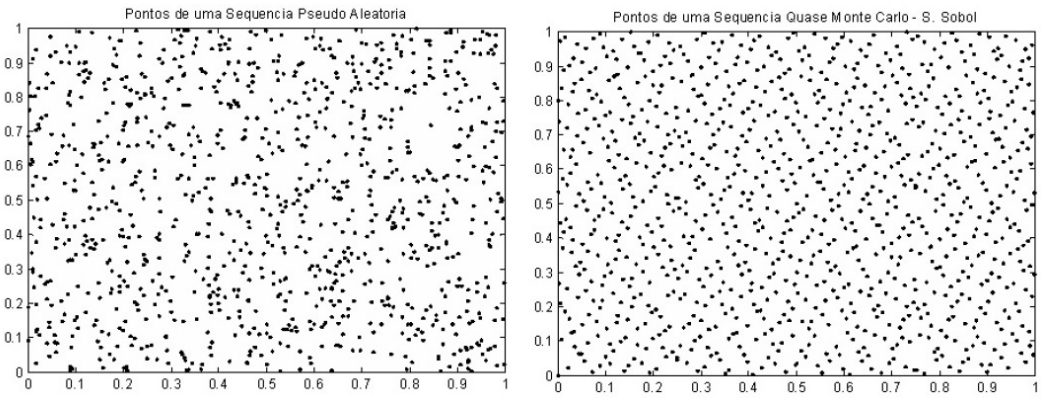

Figura 2.4: Comparação de pontos de uma seqüência pseudo aleatória e seqüência Quase Monte Carlo de Sobol.

Como pode ser observado, a seqüência de Sobol consegue preencher o espaço bidimensional de forma mais uniforme, enquanto a seqüência pseudo-aleatória apresenta regiões não preenchidas e regiões com pontos agrupados.

A seqüência de Faure é similar à seqüência de Sobol, utiliza a mesma base para todas as dimensões e os elementos da seqüência são re-ordenados.

Entretanto, esse número que representa a base não é fixo, ele é definido como o menor número primo maior ou igual ao tamanho da amostra.

A seqüência de Sobol é mais simples do que a seqüência de Faure, pois usa a base 2 para todas as dimensões, independente do tamanho da amostra. A base 2 (binária) a torna computacionalmente mais rápida, o que é uma grande vantagem para ser usada em uma simulação estocástica.

2.3 Testes de Hipótese

Testes de hipótese ou de significância consistem em uma regra decisória para rejeitar ou aceitar uma hipótese estatística com base nos resultados de uma amostra. Esta hipótese estatística corresponde a uma suposição que é feita em relação a um valor de um parâmetro populacional ou uma afirmação dada sobre a natureza da população [17].

No teste são consideradas duas hipóteses:

H_0**:** Hipótese Nula - é a hipótese a ser testada.

H_a**:** Hipótese Alternativa - é a hipótese que contraria H_0.

As hipóteses acima podem ser hipóteses simples ou hipóteses compostas, podendo ainda serem classificadas como unilaterais ou bilaterais, dependendo do interesse do estudo [40]. A realização do teste consiste em aceitar uma das hipóteses acima. Os possíveis resultados de um teste de hipótese são:

H_0	Aceitar	Rejeitar
Verdadeira	Correto	Erro tipo I Falso Positivo
Falso	Erro tipo II Falso Negativo	Correto

Tabela 2.1: Resultados de um teste de Hipótese.

Portanto, os dois erros que podem ser cometidos ao se realizar um teste de hipótese são:

1. Rejeitar a hipótese H_0, quando tal hipótese é verdadeira, denominado erro tipo I.

2. Não rejeitar a hipótese H_0, quando tal hipótese é falsa, denominado erro tipo II.

Uma parte importante do teste de hipótese é controlar a probabilidade de cometer os erros:

$\alpha =$ P(rejeitar H_0 | H_0 é verdadeira) - probabilidade do erro tipo I.

$\beta =$ P(não rejeitar H_0 | H_0 é falsa) - probabilidade do erro tipo II.

A situação ideal é aquela em que ambas probabilidades α e β são próximas de zero, entretanto, à medida que se diminui α, a probabilidade β tende a aumentar. Geralmente, o erro do tipo I é considerado mais sério do que o erro tipo II e, conseqüentemente, mais importante de ser evitado [40].

Dá-se o nome de nível de significância do teste à probabilidade α do erro do tipo I. Deve-se estabelecer um nível de significância α para cada teste a ser realizado. Por convenção, costuma-se utilizar um nível de significância de 5%, mas qualquer valor entre 0 e 1 pode ser utilizado. Portanto, o nível de significância de um teste é tal que a probabilidade de engano de se rejeitar a hipótese nula sendo ela verdadeira não é mais do que a probabilidade α indicada.

Normalmente, os métodos empregam uma estatística de teste e uma distribuição de amostragem. A estatística de teste pode ser uma média, uma proporção, diferença entre as médias, *z-score*, entre outros, calculada a partir dos dados da amostra. A escolha de uma estatística de teste depende do modelo de probabilidade escolhido e das hipóteses do teste. Se a probabilidade estatística de teste for inferior ao nível de significância α, a hipótese nula H_0 do teste é rejeitada.

A faixa de valores da estatística de teste que levam à rejeição da hipótese nula H_0 é denominada Região Crítica (RC) ou de rejeição. Já a faixa de valores que levam à não rejeição da hipótese H_0, em função do nível α, é denominada Região de Aceitação, representada por RA. Ou seja, o espaço amostral para a estatística de teste é dividido em duas regiões: uma RC e uma RA. Assim, se o valor observado da estatística de teste é um membro da RC, determina-se "Rejeitar H_0"; por outro lado, se ele não é um membro da RC, então ele é membro da RA e determina-se "Aceitar H_0".

Calcula-se também a probabilidade de se obter uma estatística de teste, no mínimo tão significativa quanto a que foi efetivamente observada na amostra, supondo que a hipótese nula é verdadeira. A essa probabilidade dá-se o nome de p-valor. A interpretação direta é que, se o p-valor é inferior ao nível de significância exigido, então diz-se que a hipótese nula é rejeitada ao nível de significância determinado.

2.3.1
Procedimento para a realização de um Teste de Hipótese

Para aplicar um teste de hipótese é preciso seguir os seguintes passos:

1. Enunciar as hipóteses H_0 e H_a;

2. Fixar o nível de significância α e identificar a estatística do teste;

3. Determinar a RC (região crítica) e a RA (região de aceitação) de acordo com α pelas tabelas estatísticas apropriadas;

4. Calcular o valor observado da estatística do teste a partir das observações ou amostras;

5. Concluir: se estatística do teste $\in$ RC $\Longrightarrow$ rejeita H_0. Caso contrário, aceita H_0.

2.3.2 Testes de Aderência

Os testes de aderência correspondem a uma classe dos testes de hipóteses que têm como função verificar a forma de uma distribuição de probabilidade, isto é, verificar se os dados referentes a uma distribuição de probabilidade se aderem à curva de um modelo distributivo hipotético [48].

Enfim, os testes de aderência são instrumentos da matemática estatística para determinar se uma dada amostra se adere ou não a um determinado modelo distributivo, ou seja, para saber qual o modelo que descreve o comportamento probabilístico da amostra dada.

A seguir são apresentados os testes de aderência escolhidos para serem utilizados nesse trabalho. Estes testes são muito empregados na comparação de amostras e seus modelos distributivos, que serão úteis para validar o modelo dessa tese.

Teste de Kolmogorov-Smirnov

O teste de Kolmogorov-Smirnov (K-S) é um teste de aderência não paramétrico [2] [17], que compara se duas distribuições de probabilidade diferem uma da outra ou se uma distribuição de probabilidade difere da distribuição em hipótese, com base em amostras finitas. É um teste para ser usado com variáveis de distribuição contínua e não com variáveis aleatórias discretas [42].

A função de distribuição acumulada empírica F_n de uma amostra de n observações X_i é dada pela equação 2-17.

$$F_n(x) = \frac{1}{n} \cdot \sum_{i=1}^{n} I_{X_i \leq x} \tag{2-17}$$

[2] são procedimentos matemáticos para testar a hipótese estatística não fazendo suposições sobre as distribuições de probabilidade das variáveis a serem avaliadas.

onde

$$I_{X_i \leq x} = \begin{cases} 1 & \text{se } X_i \leq x \\ 0 & \text{caso contrário} \end{cases}$$

sendo x uma variável aleatória.

A estatística do teste é baseada na maior diferença absoluta entre a função de distribuição acumulada $F_n(x)$ de duas amostras, a primeira com n_1 e a segunda com n_2 observações, ou a diferença entre a função de distribuição acumulada $F_n(x)$ de uma amostras, com n observações, e a função de distribuição em hipótese $F(x)$. A equação 2-18 fornece a estatística do teste [42].

$$\begin{gathered} D_n = \sup_x |F_n(x) - F(x)| \\ \Rightarrow \text{quando compara a distribuição de uma amostra.} \\ D_{n_1,n_2} = \sup_x |F_{n_1}(x) - F_{n_2}(x)| \\ \Rightarrow \text{quando compara a distribuição de duas amostras.} \end{gathered} \tag{2-18}$$

As hipóteses do teste são:

H_0**:** as distribuições do teste são idênticas.

H_a**:** as distribuições do teste não são idênticas.

A conclusão do teste consiste em rejeitar a hipótese H_0 ao nível de significância α se for satisfeita a equação 2-19 [42]. Caso contrário, aceita-se a hipótese H_0.

$$\begin{aligned} \sqrt{n} D_n &> K_\alpha && \Rightarrow \text{no teste de uma amostra.} \\ \sqrt{\tfrac{n_1 \cdot n_2}{n_1 + n_2}} D_{n_1,n_2} &> K_\alpha && \Rightarrow \text{no teste de duas amostras.} \end{aligned} \tag{2-19}$$

onde K_α representa o limite da Região de Aceitação de H_0, pois a proporção de valores menores ou iguais a K_α é dada por:

$$P(K \leq K_\alpha) = 1 - \alpha.$$

Teste Chi-quadrado

O teste Chi-quadrado, simbolizado por χ^2, é um teste de hipótese não paramétrico cujo princípio básico é comparar se um conjunto de dados observados é compatível com os valores esperados [17]. O teste verifica se a distribuição de freqüência (que contém o número de ocorrências) de certos eventos observados em uma amostra se adere a uma dada distribuição teórica específica.

O teste trabalha com as hipóteses:

H_0**:** Os dados seguem uma distribuição especificada.

H_a**:** Os dados não seguem a distribuição especificada.

Os eventos observados da amostra são divididos em n categorias e a estatística do teste é determinada pela equação 2-20 [14].

$$\chi^2 = \sum_{i=1}^{n} \frac{(Obs_i - Esp_i)^2}{Esp_i} \tag{2-20}$$

onde Obs_i é a freqüência observada da classe i e Esp_i é a freqüência esperada da classe i.

A freqüência esperada de cada classe é calculada pela equação 2-21.

$$F(Esp_i) = N \cdot (F(Y_{u_i}) - F(Y_{l_i})) \tag{2-21}$$

onde N é o tamanho da amostra, F é a função de distribuição acumulada para a distribuição a ser testada, Y_{u_i} e Y_{l_i} são, respectivamente, os limites superior e inferior para a classe i.

Quando as freqüências observadas são muito próximas às esperadas, o valor de χ^2 é pequeno. Mas, quando as divergências são grandes, χ^2 assume valores altos [14].

A estatística do teste encontra-se associada a um grau de liberdade definido pela subtração do número de classes de distribuição de freqüências do número de parâmetros estimados e da freqüência mínima esperada [17].

A estatística qui-quadrado é comparada com um valor tabelado, representado por χ^2_c, proveniente de uma distribuição qui-quadrado, que depende do grau de liberdade e do nível de significância adotado. A tomada de decisão é feita comparando-se os dois valores:

$\chi^2 \geq \chi^2_c \Rightarrow$ rejeita H_0.

$\chi^2 < \chi^2_c \Rightarrow$ aceita H_0.

Teste t

Para saber se uma amostra difere significativamente de outra, isto é, para saber se as amostras podem ser consideradas como extraídas da mesma distribuição, deve-se comparar as variâncias e as médias das amostras, que devem ser estatisticamente iguais, ou seja, não devem se diferenciar significativamente. A comparação direta das medidas não é adequada, pois é necessário considerar a dispersão dessas medidas [11]. Portanto, é preciso estabelecer se há desvio significativo entre as variâncias e as médias das duas amostras.

O Teste t compara a diferença entre as médias de duas amostras diferentes [48]. Dada duas amostras X_1 e X_2, a primeira com n_1 e a segunda com n_2 observações, o teste trabalha com as hipóteses:

H_0: $\bar{X}_1 - \bar{X}_2 = 0$

H_a: $\bar{X}_1 - \bar{X}_2 \neq 0$

sendo $\bar{X}_1$ a média da amostra X_1 e $\bar{X}_2$ a média da amostra X_2. Considerando s_1^2 e s_2^2, respectivamente, as estimativas das variâncias das amostras X_1 e X_2, o parâmetro t é determinado pela equação 2-22.

$$t = \frac{|\bar{X}_1 - \bar{X}_2|}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}} \tag{2-22}$$

Para uso em testes de significância, a distribuição da estatística de teste é aproximada por uma distribuição *t-Student* simples.

A análise do teste pode ser baseada no p-valor (definido no início dessa Seção), que impede de se rejeitar H_0 caso o seu valor esteja acima do nível de significância α (probabilidade de se cometer o erro tipo I).

Teste de Levene

Este teste verifica a homogeneidade de variâncias [2]. Sejam consideradas $k \geq 2$ amostras aleatórias independentes entre si. A amostra i representa uma coleção de n_i variáveis aleatórias independentes e identicamente distribuidas, com distribuição G_i de média μ_i e variância σ_i^2, sendo G_i, μ_i e σ_i^2 desconhecidos.

O teste trabalha com as hipóteses:

H_0: $\sigma_1 = \ldots = \sigma_k$

H_a: $\sigma_q \neq \sigma_r$ para algum $q \neq r, q = 1, \ldots, k$ *e* $r = 1, \ldots, k$

A equação 2-23 apresenta os desvios absolutos das variáveis $X_{i,j}$ com relação à média amostral do grupo X_i, denotada por $\bar{X}_i$:

$$\bar{X}_i = \frac{\sum_{j=1}^{n_i} X_{i,j}}{n_i}$$

$$Z_{i,j} = |X_{i,j} - \bar{X}_i| \tag{2-23}$$

com $j = 1, \ldots, n_i$ e $i = 1, \ldots, k$.

A estatística do teste de Levene é denotada por W_0 e calculada pela equação 2-24.

$$W_0 = \left(\frac{n-k}{k-1}\right) \cdot \frac{\sum_{i=1}^{k} n_i \cdot \left(\bar{Z}_i - \hat{Z}\right)^2}{\sum_{i=1}^{k} \sum_{j=1}^{n_i} (Z_{i,j} - \bar{Z}_i)^2} \tag{2-24}$$

onde

$$\bar{Z}_i = \frac{\sum_{j=1}^{n_i} Z_{i,j}}{n_i}$$

$$\hat{Z} = \frac{\sum_{i=1}^{k} n_i \cdot \bar{Z}_i}{n}$$

$$n = \sum_{i=1}^{k} n_i$$

O teste de Levene rejeita a hipótese H_0 se a estatística do teste W_0 for maior que o quantil de ordem $1 - \alpha$ da distribuição $F_{(k-1,n-k)}$ [2], sendo α a probabilidade de se cometer o erro tipo I.

2.4 Redes Neurais Artificiais

Segundo Haykin [28], redes neurais artificiais (RNAs) são modelos computacionais não-lineares, inspirados na estrutura massivamente paralela do cérebro humano, capazes de realizar as operações de aprendizado, associação, generalização e abstração através de conhecimento experimental.

As RNAs são estruturas compostas de diversas unidades processadoras, chamadas de neurônios artificiais, que são interligados através de conexões, denominadas pesos sinápticos, que armazenam o conhecimento adquirido pela rede.

O neurônio artificial é inspirado no neurônio biológico, cujo esquema é apresentado de maneira extremamente simplificada na Figura 2.5. Como pode ser observado, um neurônio biológico é formado por: um corpo celular ou soma que contém o núcleo da célula; diversos dendritos, através dos quais impulsos elétricos são recebidos; e um axônio, através do qual impulsos elétricos são enviados. As interligações entre neurônios são efetuadas através de sinapses, pontos de contato (controlados por impulsos elétricos e por reações químicas devidas às substâncias chamadas neurotransmissores) entre dendritos e axônios, formando uma rede de transmissão de informações [68].

Considera-se que o aprendizado acontece justamente nas sinapses, nas conexões axônio-sinapse-dendrito, onde ocorre a tradução do sinal que passa pelo axônio de um neurônio e que pode excitar (ou inibir) o neurônio seguinte. O cérebro humano possui cerca de 10^{11} neurônios e o número de sinapses é de mais de 10^{14}, possibilitando a formação de interconexões muito complexas que

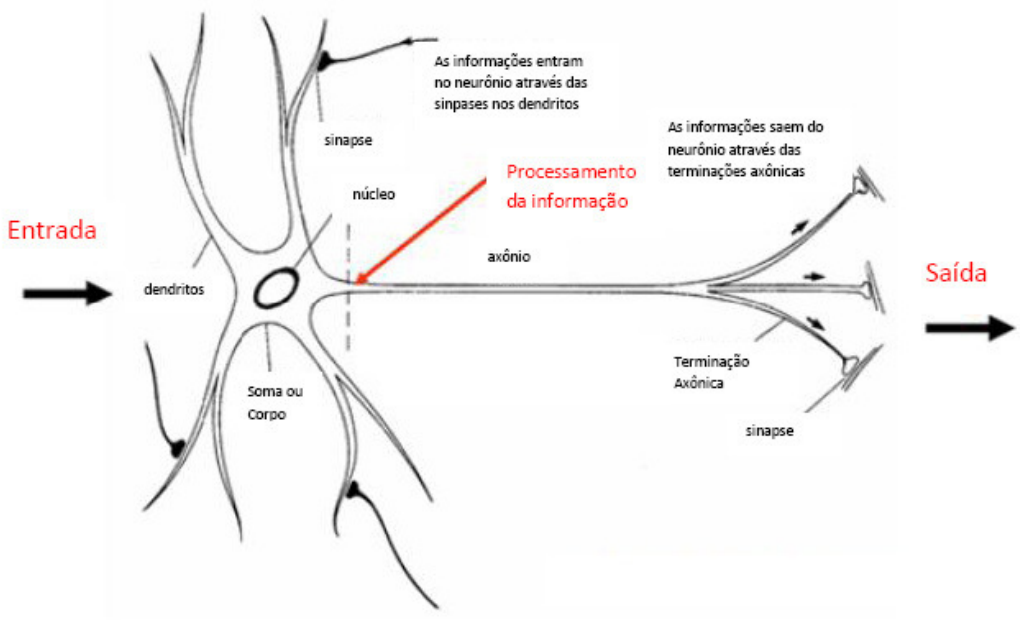

Figura 2.5: Modelo simplificado de um neurônio biológico [22].

concedem a ele um processamento altamente paralelo.

A inspiração do neurônio biológico é ilustrada na Figura 2.6. Observe que o neurônio artificial apresenta um conjunto de entradas, representadas por $x_1 \dots x_m$, simulando os dendritos, e uma saída, representada por y_i, simulando o axônio. As entradas do neurônio são ponderadas por pesos sinápticos, representados por $\omega_{i,1} \dots \omega_{i,m}$, e somadas pelo $\sum$ (que simula o corpo celular) fornecendo o potencial interno do processador, representado por net_i. O *bias*, representado por θ_i, é um termo de polarização do neurônio artificial, que pode ser tratado como um peso sináptico cuja entrada é sempre $|1|$, e o seu objetivo é aumentar (ou diminuir) a influência do valor da combinação linear das entradas [28]. A saída do neurônio é obtida através da aplicação de uma função de ativação, representada por φ, ao net_i, como pode ser visto na equação 2-25.

$$y_i = \varphi(net_i) = \varphi(\sum_{j=1}^{m} x_j * \omega_{i,j} + \theta_i). \tag{2-25}$$

A função de ativação é utilizada para limitar a amplitude da saída de um neurônio, e às vezes, introduzir não-linearidade ao modelo [6]. Existem quatro tipos de função de ativação que são muito utilizadas em RNAs, ilustradas pelas equações 2-26, 2-27, 2-28 e 2-29:

1. **Função linear**:

$$y_i = \varphi(net_i) = net_i \tag{2-26}$$

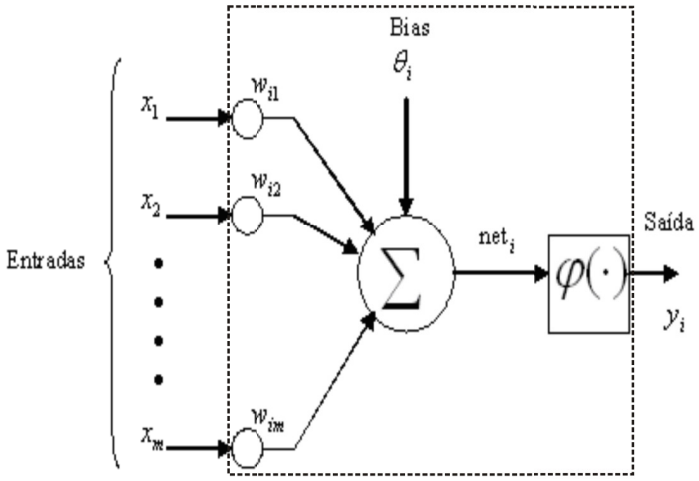

Figura 2.6: Esquema de um neurônio artificial.

Geralmente é indicada quando a saída do neurônio pode atingir qualquer valor, ou seja, não possui valores limites.

2. **Função degrau ou de limiar**:

$$y_i = \begin{cases} 1 & \text{se } net_i \geq 0 \\ 0 & \text{se } net_i < 0 \end{cases} \tag{2-27}$$

A saída dessa função assume valores rígidos de 0 ou 1. Algumas vezes é necessário que a função de ativação se estenda de -1 a +1. Quando isso ocorre na função de limiar ela é chamada de função sinal.

3. **Função sigmóide**: é uma das funções mais utilizadas na construção de RNAs. A função logística dada pela equação 2-28 assume sempre valores positivos.

$$y_i = \frac{1}{1 + \epsilon^{-\alpha \cdot net_i}} \tag{2-28}$$

onde α é um valor real.

Ela é definida como uma função estritamente crescente que exibe um balanceamento adequado entre comportamento linear e não-linear. Quando se deseja utilizar um intervalo contínuo de valores entre -1 e 1, utiliza-se a função tangente hiperbólica dada pela equação 2-29.

$$y_i = tanh(\alpha \cdot net_i) \tag{2-29}$$

Três características básicas identificam os diversos tipos de RNAs existentes:

1. o modelo (a função de ativação) do neurônio artificial;

2. a topologia da rede neural, isto é, a forma como os neurônios são interconectados na RNA;

3. a regra de aprendizagem da rede RNA.

Existem dois tipos de topologia de RNAs que são muito empregados na literatura:

Redes Neurais Não Recorrentes (sem memória) ou *Feedforward*: As redes não recorrentes são aquelas que não possuem conexões ligando neurônios da mesma camada ou ligando um neurônio de uma camada com um outro neurônio de uma camada anterior ou de uma camada não consecutiva, ou seja, não apresentam realimentação de suas saídas para as suas entradas [25]. A Figura 2.7 ilustra uma rede neural *Feedforward*. A rede possui um conjunto de nós de entradas, que apenas distribuem os padrões de entrada para a rede; uma camada de saída, cujos neurônios fornecem as saídas finais da rede neural, e uma, ou mais, camada(s) intermediária(s) ou oculta(s), cujas saídas dos neurônios são as entradas dos neurônios da camada seguinte;

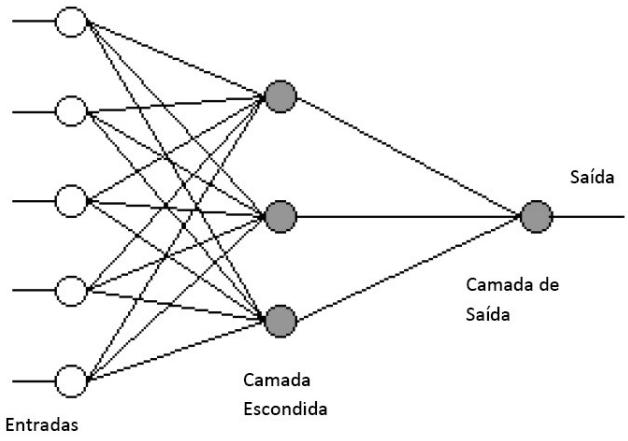

Figura 2.7: Exemplo de uma rede neural *feedforward*.

Redes Neurais Recorrentes: As redes neurais recorrentes são redes que contêm realimentação das saídas para as entradas. Desta forma, as saídas destas redes são determinadas pelas entradas atuais mais as saídas anteriores. As estruturas das redes neurais recorrentes podem apresentar interligações entre neurônios da mesma camada e/ou entre neurônios de camadas não consecutivas [18]. Logo, as estruturas das redes neurais recorrentes apresentam interconexões mais complexas que as das redes neurais *feedforwards*, como pode ser visto na Figura 2.8.

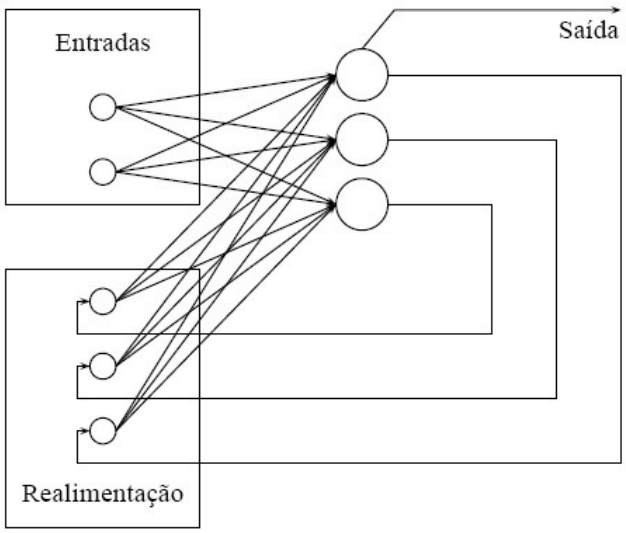

Figura 2.8: Exemplo de uma rede neural recorrente.

A arquitetura de uma RNA envolve os seguintes parâmetros: número de entradas, número de camadas, número de neurônios por camada, tipo de função de ativação dos neurônios e os padrões de conectividade. A escolha da arquitetura da RNA é decisiva para se obter um treinamento bem sucedido e depende do tipo de problema abordado e das necessidades da aplicação [55]. Cada modelo de rede neural pode ser usado para um determinado grupo de tarefas. Um dos pontos que pode ser crucial para o desempenho da rede é um bom dimensionamento do número de neurônios por camada. Embora um número maior possa dar ao modelo um maior poder computacional, isso pode torná-lo mais sensível aos problemas de *overffiting* [28], além de aumentar o tempo de treinamento. Por outro lado, um número muito pequeno de neurônios pode não ser o suficiente para modelar adequadamente o problema abordado. Ou seja, para uma RNA ter um bom desempenho, ela deve ser grande o suficiente para aprender o problema, mas também pequena o bastante para generalizar bem [1].

Para que uma rede neural, ao receber um conjunto de dados de entradas, consiga produzir uma saída desejada ou no mínimo consistente, ela precisa passar por um processo de treinamento. O treinamento é um algoritmo que ajusta os pesos sinápticos, fazendo-os convergir gradualmente para determinados valores, de forma que estes armazenem o conhecimento que é transmitido pelos conjuntos de padrões de dados apresentados à rede.

Como a rede neural aprende a partir de padrões de dados, é de extrema importância a escolha das variáveis a serem tratadas como entradas da rede. É preciso escolher variáveis que sejam bem representativas e que contenham

observações suficientes para se extrair o conhecimento necessário para o bom aprendizado da rede neural [68].

Os procedimentos de treinamento podem ser classificados em dois tipos:

1. **Treinamento Supervisionado**: Os padrões de dados são formados pelo conjunto de entradas e pelo conjunto de saídas desejadas. Durante o treinamento, as entradas são apresentadas à RNA e são calculadas as saídas da rede. Em seguida, cada saída da RNA é comparada com a respectiva saída desejada do padrão de entrada apresentado, gerando um erro da diferença entre as duas. O algoritmo de treinamento ajusta os pesos sinápticos com o objetivo de minimizar este erro encontrado. Este processo é repetido até que o erro, para todos os padrões de dados de treinamento, atinja um valor mínimo aceitável.

2. **Treinamento Não Supervisionado**: Os padrões de entradas não contêm saídas desejadas, logo, não existem comparações que geram sinais de erro. O processo de treinamento extrai as propriedades estatísticas do conjunto de padrões de entrada, formando agrupamentos de padrões similares.

A apresentação completa de todo o conjunto de padrões de treinamento é chamada de época e, geralmente, define-se um número máximo de épocas como um dos critérios de parada do algoritmo de treinamento.

Após o treinamento da rede neural, é apresentado à mesma um conjunto de padrões de teste que contém padrões que nunca foram vistos antes pela rede neural mas que pertençam à mesma população dos padrões de treinamento. Se o aprendizado foi bem sucedido, a RNA tem que ser capaz de fornecer uma saída correta para um padrão de teste. Portanto, diz-se que a RNA tem uma boa capacidade de generalização quando ela consegue fazer um mapeamento entrada-saída correto, mesmo que a entrada seja um pouco diferente dos exemplos que a rede já conhece [28].

As RNAs tem sido aplicadas com sucesso em muitos problemas reais de previsão, classificação e controle, por representarem técnicas capazes de modelar funções extremamente complexas [55]. Contudo, o desempenho das RNAs para a solução de um determinado problema é totalmente dependente de parâmetros importantes do seu projeto de construção.

De acordo com o problema a ser abordado é preciso definir: o tipo de modelo de RNA, a arquitetura da rede e o algoritmo de aprendizado dos pesos [1]. Além disso, antes de serem apresentados à RNA, os dados de entrada passam por transformações como o pré-processamento e a normalização,

medidas necessárias para adaptá-los à uma forma mais apropriada para sua utilização nas RNAs [55]. Por exemplo, as variáveis nominais devem ser transformadas em classes para serem utilizadas em RNAs. Já a normalização é necessária para igualar as magnitudes das variáveis numéricas, evitando que variáveis com valores de maior ordem de magnitude dominem ou distorçam os pesos da rede [10]. Assim os valores dos dados de entrada devem ser normalizados dentro do intervalo definido de acordo com a funções de ativação dos neurônios da rede. Geralmente, esse intervalo é entre 0 e 1. O desempenho do treinamento das RNAs também é dependente da quantidade de dados disponíveis para o treinamento, pois, com uma quantidade pequena de dados, o aprendizado das RNAs pode ficar comprometido [22].

2.4.1 Redes Neurais FeedForward

A topologia de rede neural artificial (RNA) utilizada nesse trabalho é a de RNAs *feedforward*. Como foi visto, são redes não recorrentes, logo, as entradas externas se propagam sempre para frente, através das camadas da rede, neurônio por neurônio, até atingir a saída, onde termina como um sinal de saída da rede neural.

As primeiras RNAs *feedforward* que apareceram na literatura foram *Perceptron* de Rosenblatt [57] e Adaline de Widrow [66]. Depois surgiram as redes neurais multicamadas, referenciadas como redes MLP (*MultiLayer Perceptron* [58]). As redes MLP, com uma ou duas camadas ocultas e um número suficiente de neurônios, são consideradas aproximadores universais, ou seja, estas redes podem representar uma grande gama de funções [6].

O aprendizado desta rede é do tipo supervisionado e o algoritmo de treinamento mais utilizado é o algoritmo de retropropagação do erro (*back-propagation*). O algoritmo de retropropagação do erro consiste em calcular o erro na saída da rede neural e retropropagá-lo pela rede modificando os pesos sinápticos, utilizando a técnica de busca do gradiente decrescente dos erros [28].

O algoritmo de retropropagação basicamente consiste das seguintes fases:

1. Propagação do sinal de entrada: um padrão de entrada é processado pelos neurônios da rede e seu efeito se propaga, camada por camada, até produzir um conjunto de saídas da rede neural. Durante esta etapa, os pesos da rede ficam fixos. Na camada de saída, a resposta do neurônio i, representada por y_i, é comparada com a saída desejada do padrão de entrada apresentado, representada por $\tilde{y}_i$, e é calculado um erro desta variação, representado por e_i, como apresenta a equação 2-30.

$$e_i(n) = \tilde{y}_i(n) - y_i(n) \tag{2-30}$$

onde i refere-se ao neurônio da camada de saída e n ao padrão de dados de treinamento apresentado.

2. Retropropagação do erro: o erro é retropropagado pela rede neural, ajustando os pesos sinápticos, com uma quantidade proporcional ao gradiente deste erro, visando aproximar as respostas da RNA às respostas desejadas.

Este processo iterativo deve ocorrer época por época, até que um critério de parada do treinamento seja atingido. Um dos critérios de parada mais utilizado é o número máximo de épocas que o algoritmo deve ser executado. Geralmente, utiliza-se também como critério de parada uma função de custo da rede neural, que é calculada com base nos erros gerados nos neurônios da camada de saída para todos os padrões de treinamento. O algoritmo é encerrado quando o valor dessa função de custo atinge um nível mínimo aceitável, indicando que a rede neural conseguiu aprender a solucionar o problema apresentado. Logo, a função de custo é uma medida de desempenho do aprendizado da rede neural [68].

Uma função de custo muito utilizada é a soma dos erros quadráticos (SSE), apresentada na equação 2-31.

$$SSE = \sum_n \sum_i e_i(n)^2 \tag{2-31}$$

onde n corresponde ao padrão de dados apresentado em que foi calculado o erro e_i do neurônio i da camada de saída, conforme a equação 2-30.

O objetivo do algoritmo é ajustar os pesos de forma a minimizar a função de custo utilizada. Quando a função de custo é a SSE, esse ajuste é dado por uma quantidade Δ, como ilustram as equações 2-32 e 2-33 [28].

$$\Delta\omega_{ij}(t) = \eta \cdot \delta_i(t) \cdot s_j(t) \tag{2-32}$$

$$\omega_{ij}(t+1) = \omega_{ij}(t) + \Delta\omega_{ij}(t) \tag{2-33}$$

onde t corresponde à época de atualização, η é um parâmetro denominado de taxa de aprendizado, os índices i e j referem-se ao neurônio i da camada posterior e ao neurônio (ou entrada) j da camada anterior e s_j é o valor de entrada do neurônio i. O termo δ_i, para um neurônio da camada de saída, é calculado como apresenta a equação 2-34

$$\delta_i = e_i \cdot \varphi'(net_i) \tag{2-34}$$

onde φ' é a derivada da função de ativação do neurônio i. Já para o caso do neurônio i pertencer a uma camada oculta, o termo δ_i é calculado de acordo com a equação 2-35

$$\delta_i = (\sum_k \delta_k \cdot \omega_{ki}) \cdot \varphi'(net_i) \tag{2-35}$$

onde k é o índice dos neurônios da camada posterior.

Existem variações do algoritmo de treinamento *backpropagation* que são muito utilizadas. Um dos exemplos é o algoritmo de treinamento *Levenberg-Marquardt* (LM), considerado o método mais rápido para treinamento de RNAs *feedforward* com uma quantidade moderada de pesos sinápticos. Ele se baseia, para a aceleração do treinamento, na determinação das derivadas de segunda ordem do erro em relação aos pesos, diferindo do algoritmo *backpropagation* tradicional que considera as derivadas de primeira ordem [25].

Para que a rede neural consiga aprender bem, o conjunto de dados deve conter amostras suficientes para cobrir todo o domínio de interesse do problema. Geralmente, costuma-se dividir este conjunto de amostras em três subconjuntos distintos:

1. Conjunto de dados de treinamento: contém dados que serão usados na fase de treinamento da rede neural;

2. Conjunto de dados de validação: contém dados que serão usados para fazer o teste de generalização da rede durante a sua fase de aprendizado;

3. Conjunto de dados de teste: contém dados que serão usados para fazer o teste de generalização da rede após o seu treinamento.

Após a passagem de um número de épocas de treinamento, onde os pesos sinápticos foram ajustados, costuma-se fazer uma propagação com os dados de entrada do conjunto de validação. Nenhum dos dados do conjunto de validação podem ter sido utilizados na fase de treinamento da rede neural, para que assim possa-se medir a capacidade de generalização da rede.

Calcula-se uma função de custo, igual à utilizada no treinamento, com os erros obtidos através da diferença entre as saídas da RNA e as saídas desejadas do conjunto de validação. Essa função de custo de validação pode ser um outro tipo de critério de parada do algoritmo de treinamento. Quando o valor dessa função de custo começar a aumentar, é uma indicação de que a rede está perdendo a sua capacidade de generalizar. Costuma-se dizer que a rede neural passou a decorar os dados de treinamento. Logo, o treinamento precisa ser interrompido, o que é chamado na literatura de *early stopping* (parada antecipada) [28], como ilustra a Figura 2.9.

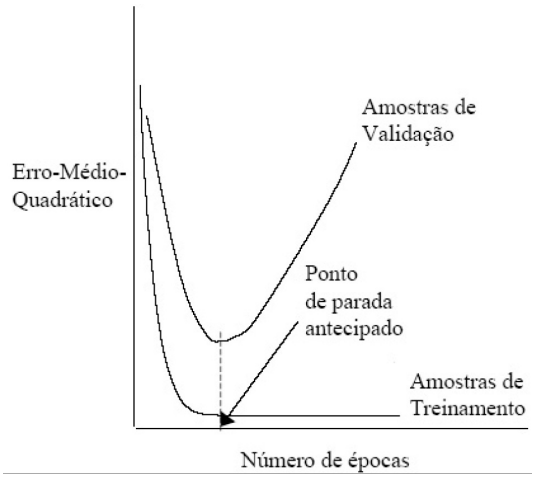

Figura 2.9: Monitoramento da função de custo de validação ocasionando um "early stopping" no treinamento de uma rede neural [28].

2.4.2
Aplicação de Redes Neurais FeedForward em Séries Temporais

Muitos problemas reais apresentam características complexas, tais como não-linearidades e comportamento caótico, onde uma aproximação linear pode fornecer como resultado um modelo pouco eficiente, de aplicabilidade limitada ou inadequada. Alguns modelos estatísticos não-lineares mais tradicionais têm a desvantagem de necessitar de um conhecimento aprofundado do domínio para a sua construção.

Recentemente, muitas pesquisas vêm aplicando redes neurais artificiais (RNAs) na área de análise e modelagem de séries temporais. As RNAs são modelos não-lineares com alto poder computacional (por apresentarem um grande número de parâmetros), que têm potencial de modelar adequadamente uma quantidade maior de séries em comparação aos modelos estatísticos lineares clássicos. Uma das características que tornam vantajosa a sua aplicação é que as RNAs não precisam assumir nenhum tipo de distribuição dos dados a priori, aprendem a distribuição através de exemplos, e têm habilidade para manipular dados adquiridos de diferentes fontes e com diferentes níveis de precisão e ruídos [12]. Também podem ser aplicadas em séries temporais não-estacionárias sem necessitar de transformações para obter comportamento estacionário, como os modelos estatísticos lineares [55].

No contexto de RNAs, existem duas redes candidatas para o processamento temporal [28]: as RNAs *feedforward* e as RNAs recorrentes. As redes *feedforward* mais usadas em séries temporais são as redes MLP (*Multilayer Perceptron*) [58] e as redes RBF (*Radial Basis Function*) [15]. Uma das princi-

pais diferenças entre esses dois modelos está no tipo de função de ativação dos neurônios da camada oculta. Enquanto as redes MLP fazem uso de funções sigmóides, as redes RBF fazem uso de funções gaussianas na camada oculta. Já as redes recorrentes mais utilizadas são as redes de Elman [18]. As redes recorrentes permitem que as características temporais da informação sejam memorizadas [24]. Apesar de serem mais gerais e potencialmente mais poderosas que as redes *feedforward*, as redes recorrentes apresentam algumas desvantagens, como por exemplo, o aprendizado mais lento [55].

Para que uma RNA *feedforward* se comporte como um modelo de processamento temporal, é preciso que ela apresente habilidade de memória de curto prazo, de forma que sua saída se torne uma função do tempo [28]. Uma forma simples de inserir memória de curto prazo na estrutura de uma RNA é através de atrasos de tempo, que podem ser implementados, por exemplo, na camada de entrada da rede. A memória de curto prazo permite à RNA capturar a informação temporal contida na entrada, inserindo essa informação em seus pesos sinápticos. Segundo Haykin [28], os modelos de redes neurais *feedforward* onde a estrutura da memória é localizada no terminal de entrada dos neurônios da primeira camada são definidos como TLFN (*Time Lagged Feedforward Networks*).

No contexto de séries temporais, uma RNA TLFN é considerada um modelo auto-regressivo não-linear, onde a camada de entrada recebe os regressores da série, fornecendo a cada instante de tempo apenas uma visão parcial da série, e a camada de saída retorna o valor da série em um dado tempo. A camada de entrada dessas redes recebe o nome de janela de tempo e o tamanho da janela é referenciado como a ordem do modelo [55]. A Figura 2.10 ilustra um exemplo de uma RNA *feedforward* MLP, com uma camada oculta com três neurônios, e com uma janela de tempo de ordem 4.

2.5 Algoritmos Genéticos

Os Algoritmos Genéticos (AG) são algoritmos de busca inspirados nos mecanismos da genética e da seleção natural e foram introduzidos por *John Holland* [31]. Esses algoritmos manipulam uma população de indivíduos, onde cada indivíduo corresponde a uma solução candidata para o problema no qual o algoritmo genético é aplicado. O espaço que compreende todas as possíveis soluções candidatas do problema forma o seu espaço de busca ou região viável, que mantendo a analogia com a natureza, corresponde ao ambiente desta população.

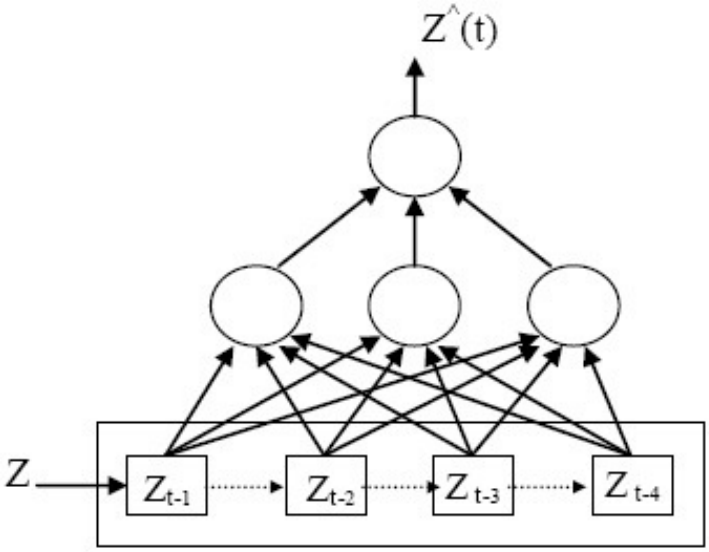

Figura 2.10: RNA *feedforward* implementada como um modelo auto-regressivo de ordem 4 [55].

Na natureza, cada ser vivo pluricelular é constituído de células que possuem uma certa quantidade de unidades chamadas cromossomos (filamentos contidos nos núcleos das células que dispõem os genes em uma ordem linear). De maneira análoga, nos algoritmos genéticos, cada indivíduo da população é representado por um cromossomo (cadeia de informações relativas à função objetivo do problema) [5].

Na natureza, os genes são segmentos funcionais de *DNA*, localizados em posições particulares nos cromossomos (chamadas *locus*), que contêm informações para produzir uma proteína específica. De maneira análoga, nos algoritmos genéticos, os genes são porções do cromossomo que codificam o(s) parâmetro(s) otimizáveis da função objetivo do problema no qual o AG é aplicado [46].

Na natureza, os alelos são uma das formas que o gene pode assumir em um determinado *locus* em um cromossomo. Por exemplo, no gene que determina a cor dos olhos de um indivíduo, os alelos deste gene assumem valores azul, verde, castanho ou preto. De maneira análoga, os alelos, nos algoritmos genéticos, são os valores que um gene pode assumir. Por exemplo, no caso do cromossomo ser representado por uma cadeia de bits, os alelos dos genes são 0 ou 1 [23]. A Figura 2.11 ilustra um cromossomo que representa um indivíduo da população. Este cromossomo é particionado em genes e o alelo de um gene é destacado para sua identificação.

Na definição de um AG, é preciso escolher uma forma para representar

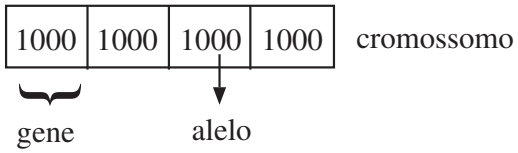

Figura 2.11: Um indivíduo de uma população de AG

as soluções candidatas do problema, dado que os AG's trabalham sobre uma codificação das mesmas [5]. Seguindo a analogia com a genética, cada solução candidata é um fenótipo e a sua codificação o seu genótipo. A codificação das variáveis do problema a ser otimizado proporciona um grande impacto no desempenho de busca, devendo ser o mais simples possível sem perder as características de representação do problema [46]. Matematicamente, esta codificação é uma função que associa os elementos do espaço de fenótipos (espaço de busca) aos elementos do espaço de genótipos (ambiente da população de cromossomos do AG).

A forma mais tradicional de codificação, introduzida por John Holland [31], é a codificação binária, cujos alelos pertencem ao conjunto $\{0, 1\}$. Porém, para variáveis reais existe a codificação real, que oferece uma maneira mais direta de se trabalhar em domínios contínuos. Neste tipo de codificação, cada cromossomo é um vetor pertencente a $\Re^n$. As vantagens da representação real são:

- permite que conhecimentos prévios sobre o problema sejam incorporados com mais facilidade e naturalidade;
- permite o uso de cromossomos de comprimentos menores, gerando ganhos significativos de memória;

Em contrapartida, torna-se necessário um ajuste de uma série de aspectos do AG, desde a inicialização da população à utilização de operadores de *crossover* e mutação.

De acordo com a seleção natural, os indivíduos mais aptos ao ambiente possuem maior chance de sobreviver e repassar sua herança genética aos seus descendentes. De maneira análoga, nos AG's, cada indivíduo da população é associado a uma função de aptidão que avalia a qualidade daquele indivíduo, ou seja, a qualidade da solução candidata ao problema [23]. Em geral, quando o problema em questão é um problema de otimização, a função de aptidão corresponde à função objetivo do problema (à função que se quer otimizar). A função objetivo fornece uma medida de qualidade em relação ao problema e a

função de aptidão transforma essa medida em uma grandeza que representa a oportunidade de reprodução dos indivíduos [29].

Os AG's selecionam os indivíduos (cromossomos) para gerar os cromossomos filhos na população da próxima geração. Este processo de escolha dos indivíduos é chamado de esquema ou método de seleção. Inúmeros esquemas de seleção já foram propostos e implementados na história dos AG's, sendo alguns não biologicamente plausíveis [34].

Um dos esquemas de seleção mais simples e difundidos em AG's é o método da roleta, onde cada indivíduo recebe um setor da roleta de acordo com a sua probabilidade de seleção, proporcional à sua aptidão [46]. Seja f_i a função de aptidão do i-ésimo indivíduo de uma população com n indivíduos. A sua probabilidade de vir a ser selecionado para reprodução é dada por p_i, como ilustra a equação 2-36. Assim, os indivíduos mais aptos recebem uma porção maior da roleta.

$$p_i = \frac{f_i}{\sum_{j=1}^{n} f_j} \tag{2-36}$$

Um outro esquema de seleção muito difundido é o método de seleção por *Rank*. Neste esquema de seleção, os indivíduos são ordenados na população de acordo com o valor de suas aptidões, do melhor para o pior indivíduo. O processo de seleção não utiliza o valor da aptidão e sim a posição relativa de cada indivíduo na população ordenada, denominada *rank* [23]. O primeiro indivíduo do *rank* é o melhor indivíduo da população e, conseqüentemente, tem a maior probabilidade de seleção.

Na natureza, através da reprodução, novos indivíduos são gerados, recebendo as características genéticas de seus indivíduos genitores - o que chamamos de hereditariedade. Na reprodução sexuada, ocorre a recombinação genética ou *crossing over*, onde os novos indivíduos recebem parte dos genes da mãe e parte dos genes do pai (pela fusão de parte de seus cromossomos), como pode ser visto na Figura 2.12.

De maneira análoga, nos algoritmos genéticos, existe a operação genética de recombinação ou cruzamento, onde os indivíduos de uma população se recombinam geneticamente, gerando novos indivíduos que herdam parte dos genes do indivíduo pai_1 e parte dos genes do indivíduo pai_2 [31]. O processo de recombinação constitui um dos passos mais importantes no mecanismo evolutivo dos algoritmos genéticos, por isso, na literatura existem diversos tipos de operadores de cruzamento. A escolha de um operador genético é fortemente dependente da codificação usada nos cromossomos dos indivíduos da população e do problema a ser solucionado [34].

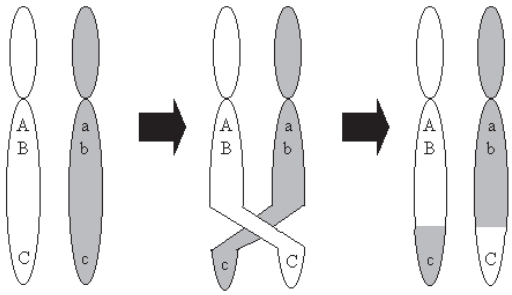

Figura 2.12: Recombinação genética de dois cromossomos

Na natureza, a mutação pode ser definida como um evento que dá origem a alterações qualitativas ou quantitativas no material genético. As mutações ocorrem ao acaso e são raras, mas existe a possibilidade de desenvolverem novas características desejáveis. Por exemplo, já foram obtidas mutantes de cevada que apresentam resistência a doenças causadas por fungos, caule mais rijo, aumento no conteúdo de proteínas e sementes sem casca [52].

Nos algoritmos genéticos, além da operação genética de recombinação, também existe a operação genética de mutação que consiste em pequenas alterações realizadas nos genes dos novos indivíduos gerados. A mutação garante uma boa exploração de todo o espaço de busca [5], mantendo a variabilidade da população, podendo, ao acaso, encontrar regiões do espaço de busca onde se encontram ótimas soluções. Assim como os operadores de cruzamento, existem diversos tipos de operadores de mutação na literatura, sendo a sua aplicação também fortemente dependente do tipo de codificação utilizado nos indivíduos e do problema a ser solucinonado [34].

Mantendo a analogia com a natureza, cada iteração do processo evolutivo dos AG's, onde os novos indivíduos gerados pela recombinação são inseridos em uma nova população, é chamada de geração. Cada população é, portanto, identificada pela sua geração e, após várias gerações, existe uma grande probabilidade de que a população seja composta por indivíduos bem adaptados ao ambiente, isto é, que a população contenha boas soluções para o problema no qual o AG é aplicado [46].

Um pseudo-código genérico capaz de englobar a maioria dos AG's existentes é apresentado na Figura 2.13.

Existem diversas variações deste código que vão desde diferenças conceituais a detalhes de implementação, mas a idéia básica é sempre a mesma[3].

Os parâmetros que influenciam no desempenho de um AG são:

[3]Para maiores detalhes de implementação de um algoritmo genético, consulte [23]

Algoritmo Genético Genérico

Inicio

Inicialize a população de cromossomos (geração i = 1)

Avalie os indivíduos na população

Repita (processo evolutivo)

Selecione indivíduos para reprodução

Aplique operadores de crossover e mutação

Avalie indivíduos gerados na população

Selecione indivíduos para sobreviver (geração i = i +1)

Até critério de parada

Fim

Figura 2.13: Algoritmo Genético Genérico

- **Tamanho da População**: a população não deve ser muito pequena, caso contrário, o AG não consegue realizar uma boa cobertura do espaço de busca e, assim, corre o risco de não explorar uma área que contenha boas soluções. Por outro lado, uma população muito grande exige muito tempo de processamento, prejudicando assim o desempenho do AG [34].
- **Taxa de *Crossover***: é a probabilidade de ocorrer recombinação entre dois indivíduos pais selecionados. Se esta taxa for muito pequena, ocorrerá pouca recombinação genética na população, o que pode tornar a busca por uma boa solução muito lenta [5]. Por isso, geralmente se utiliza uma taxa alta de *crossover*.
- **Taxa de Mutação**: é a probabilidade de ocorrer mutação em um indivíduo. A mutação ajuda o algoritmo a explorar melhor o espaço de busca, evitando que a busca fique estagnada em uma região [46]. Entretanto, não se deve utilizar uma taxa de mutação muito alta, senão a busca pode se tornar totalmente aleatória, o que prejudicará a convergência do algoritmo para uma boa solução.
- **Número de Gerações**: é um dos critérios de parada de um AG, que representa o número total de gerações de populações a serem criadas. Um número pequeno de gerações faz com que o algoritmo tenha uma convergência prematura para uma solução não muito boa. Um número muito grande de gerações permite que o AG percorra melhor o domínio

do espaço de busca, mas, em contrapartida, exige maiores recursos computacionais [34].

2.6 Resumo

Este capítulo apresentou um breve resumo dos fundamentos teóricos dos modelos utilizados neste trabalho. Foi dado um enfoque especial aos conceitos de processos estocásticos e de redes neurais artificiais, na sua utilização em séries temporais.

O próximo capítulo apresenta o modelo proposto nessa tese, que consiste de um processo estocástico baseado em redes neurais artificiais.