3
Métodos de Previsão de Séries Temporais

3.1
Séries Temporais

Pode-se definir uma série temporal como sendo um conjunto de dados observados e ordenados segundo parâmetro de tempo e com dependência serial, sendo esse espaço de tempo entre os dados disponíveis equidistantes (horários, diário, semanal, mensal, trimestral, anual, etc.) (Souza & Camargo, 2004).

Para que uma determinada série seja classificada como uma série temporal, é necessário que ela preencha outro pré-requisito: os dados também devem apresentar uma dependência serial entre eles. Por exemplo: os dados de uma variável aleatória Z (consumo de energia) no instante t, com t variando de 1 até N, possa, de certa maneira, conter informações necessárias para que seja determinado o valor dessa variável no instante t+1. Cabe mencionar que, N representa o número de observações da série temporal em questão. As séries temporais podem ser classificadas como discretas, contínuas, determinísticas, estocásticas, multivariadas e multidimensionais.

Segundo a abordagem de componentes não observáveis, as séries temporais podem ser representadas como a combinação de quatro componentes (Mendenhall, 1993):

- Tendência;
- Cíclica;
- Sazonal;
- Erro.

As componentes de tendência são frequentemente, aquelas que produzem mudanças graduais em longo prazo. São normalmente provocadas, por exemplo, pelo crescimento constante na população, no produto interno bruto, no efeito da

competição, ou por outros fatores que falham na tentativa de produzir mudanças repentinas, mas produzem variações graduais e regulares ao longo do tempo.

As componentes *cíclicas* são aquelas que provocam oscilações de subida e de queda nas séries, de forma suave e repetitiva, ao longo da componente de tendência.

Geralmente os efeitos cíclicos em uma série são causados por mudanças na demanda do produto, por ciclos de negócios e, em particular, pela inabilidade de se suprir as necessidades do consumidor.

As componentes *sazonais* em uma série são aquelas oscilações de subida e de queda que sempre ocorrem em um determinado período do ano, do mês, da semana, do dia ou horário. A diferença essencial entre as componentes sazonais e cíclicas é que a primeira possui movimentos facilmente previsíveis, ocorrendo em intervalos regulares de tempo, por exemplo, ano a ano, mês a mês, semana a semana, ou mesmo dia a dia. Já os movimentos cíclicos tendem a ser irregulares, ocorrendo sobre um período de muitos anos.

A quarta componente da série, chamada de *componente de erro*, apresenta

movimentos ascendentes e descendentes da série após a ocorrência de um efeito de tendência, um efeito cíclico, ou de um efeito sazonal. Nas componentes de erro aparecem flutuações de período curto, com deslocamento inexplicável e geralmente são causadas, entre outros motivos, por eventos políticos e oscilações climáticas imprevisíveis.

Quanto à variabilidade das observações, podem-se classificar as séries temporais em **estacionárias,** quando as suas estatísticas não são afetadas por variações no tempo, e **ergódigas,** se apenas uma realização do processo estocástico é suficiente para se obter todas as estatísticas do mesmo (Moretin & Toloi, 2006).

A maioria dos métodos de previsão baseia-se na idéia de que as observações passadas contêm informações sobre o padrão de comportamento da série temporal. O propósito dos métodos é distinguir o padrão de qualquer ruído que possa estar contido nas observações e então usar esse padrão para prever valores futuros da série. Uma grande classe de modelos de previsão, tenta tratar ambas as causas de flutuações em séries de tempo e a das suavizações (média móvel e amortecimento exponencial). Técnicas específicas desse tipo assumem que os valores extremos da série representam a aleatoriedade e assim, por meio da suavização desses extremos, pode-se identificar o padrão básico (Morettin e Toloi, 2006).

Os modelos de previsão podem ser classificados em univariados, os quais têm a previsão dos valores futuros explicados somente pelos valores passados da própria série ou causais, os que levam em conta outras informações relevantes como influentes para a previsão de uma variável

3.2
Persistência

Um dos modelos mais utilizados na previsão de curto-prazo da velocidade do vento é o método da persistência. Este método corresponde ao método da média móvel simples em que a previsão é a média das N observações mais recentes da série X como pode ser observado na equação (3.1):

$$X_t = \frac{1}{N}\sum_{i=1}^{N} X_{t-1} \qquad (3.1)$$

O método de persistência é considerado o método de previsão mais simples, visto que realiza a previsão com base nos últimos valores da série. Esse método é muito utilizado no caso de insuficiência de dados relacionados a velocidade de ventos e deve ser utilizado apenas para previsão de curto-prazo (até poucas horas à frente). No caso particular em que N é igual 1, o método da persistência corresponde ao método de previsão ingênuo (naive).

A seguir são apresentados os modelos de Box& Jenkins, regressão harmônica, redes neurais artificiais e o sistema de infererência neuro-*fuzzy adaptativo* (ANFIS)

3.3
Modelos de Box & Jenkins

Uma metodologia bastante utilizada na análise de modelos paramétricos é conhecida como abordagem de Box & Jenkins (1970). Tal metodologia consiste em ajustar modelos Auto-Regressivos integrados de médias móveis, ARIMA (p, d, q), a um conjunto de dados. A estratégia para construção deste modelo é baseada em um ciclo interativo, no qual a escolha da estrutura do modelo baseia-se nos próprios dados (Morettin e Toloi, 2004).

A estratégia para a construção do modelo será baseada em um ciclo interativo, na qual a escolha da estrutura do modelo é baseada nos próprios dados.

As etapas do ciclo interativo são:

- Especificação: uma classe geral do modelo é considerada para análise ;
- Identificação de um modelo, com base na análise de autocorrelações, autocorrelações parciais e outros critérios;

- Fase de estimação, na qual os parâmetros de modelo identificado são estimados;
- Fase de verificação ou diagnóstico do modelo ajustado, através de uma análise de resíduos, para se saber se este é adequado para fins em vista (previsão, por exemplo).

Um processo estocástico pode ser entendido como um modelo que descreve a estrutura de probabilidade de uma seqüência de observações ao longo do tempo.

Considere um processo estocástico como sendo uma família $Z=\{Z_t , t\in N\}$ tal que para cada t, Z_t é uma variável aleatória. Suponha que Z_t tenha origem em um experimento que pode ser repetido sob condições idênticas, a cada experimento obtém-se um registro dos valores de Z_t ao longo do tempo. Cada registro particular é uma realização do processo estocástico e uma série temporal é uma realização amostral do processo estocástico, i.e., é uma amostra finita do conjunto de todas as trajetórias possíveis que podem ser geradas pelo processo estocástico. Por exemplo, uma série temporal com *m* observações sucessivas pode ser considerada como uma realização amostral entre todas as seqüências de tamanho *m* que poderiam ser geradas por um mesmo processo gerador dos dados ou processo estocástico.

Um processo estocástico está determinado quando são conhecidas suas funções de distribuição de probabilidade conjuntas, porém, como estas não são conhecidas e dispõe-se de apenas uma amostra do processo (a série temporal observada) assumem-se os pressupostos de estacionariedade e ergodicidade do processo estocástico.

A estacionariedade significa que as características do processo estocástico permanecem invariantes ao longo do tempo. Em um sentido estrito, a estacionariedade implica que as variáveis aleatórias Z_t e Z_{t+k} têm idênticas distribuições de probabilidade qualquer que seja k. Uma condição menos restritiva é a estacionariedade em sentido lato ou de segunda ordem na qual considera-se como sendo estacionário um processo com valor médio, $E(Z_t)$, e variância, $E[(Z_t - \mu)^2]$ constantes e autocovariâncias, $Cov(Z_t,Z_{t+k})$, dependentes apenas do intervalo de tempo (*lag*) k entre as observações, ou seja:

$$E(Z_t) = E(Z_{t+k}) = \mu \ \forall \ t \qquad (3.2)$$

$$E[(Z_t- \mu)^2] = \sigma^2 \ \forall \ t \qquad (3.3)$$

$$Cov(Z_t,Z_{t+k}) = Cov(Z_{t+m},Z_{t+m+k}) \ \forall \ m \qquad (3.4)$$

Se o processo estocástico for Gaussiano (Z_t segue uma distribuição normal) e estacionário em sentido lato, ele será estritamente estacionário, pois a distribuição normal é determinada unicamente em termos do primeiro e do segundo momento.

Quando se trabalha com uma série temporal extraída de um processo estocástico estacionário está-se diante de uma realização amostral que apresenta uma forma geral similar á outras amostras que poderiam ter sido extraídas o que torna possível estimar as características do processo e fazer previsões.

O pressuposto da ergodicidade de um processo estocástico significa que apenas uma realização do processo estocástico é suficiente para se obter todas as estatísticas do mesmo. Todo o processo ergódico também é estacionário, pois uma realização de um processo não estacionário não poderá conter todas as informações necessárias para a especificação do processo.

Assim, tendo-se como base uma determinada série temporal, gerada por um processo estocástico estacionário, onde o valor atual é dado por Z_t, Box & Jenkins propõem o seguinte modelo para descrever o processo estocástico gerador da série:

$$Z_t = \phi y_{t-1} + ... + \phi_p y_{t-p} + \varepsilon_t - \theta_1 \varepsilon_{t-1} - ... - \theta_q \varepsilon_{t-q} \qquad (3.5)$$

que pode ser reescrita pela seguinte equação obtida com o auxílio do operador defasagem ($B^d Z_t = Z_{t-d}$) em termos de dois polinômios:

$$\left(1 - \phi_1 B - ... - \theta_p \phi^p\right) Z_t = \left(1 - \theta_1 B - ... - \theta_q B^q\right) \varepsilon_t \qquad (3.6)$$

onde, ϕ e θ denotam os parâmetros do modelo e ε_t é um ruído branco, um processo estocástico Gaussiano com média nula, variância constante e não autocorrelacionado:

$$E(\varepsilon_t)=E(\varepsilon_{t+k}) = 0 \ \forall \ t \qquad (3.7)$$

$$E[\varepsilon_t^2] = \sigma_\varepsilon^2 \ \forall \ t \qquad (3.8)$$

$$Cov(\varepsilon_t, \varepsilon_{t+k}) = Cov(\varepsilon_{t+m}, \varepsilon_{t+m+k}) = 0 \ \forall \ m \qquad (3.9)$$

Conforme indicado na equação 3.5, o modelo univariado de Box & Jenkins consiste em explicar uma variável aleatório Z por meio de seus valores passados, bem como de choques aleatórios, não havendo outras variáveis explicativas.

Na equação 3.5, as defasagens da variável Z_t no polinômio do lado esquerdo representam a parte autoregressiva do modelo (AR), enquanto as defasagens dos choques aleatórios ε_t no polinômio do lado direito representam a parte média móvel (MA). A equação representa uma ampla classe de modelos denominados ARMA(p,q), onde p representa a ordem de defasagem do termo autoregressivo e q a ordem de defasagem do termo de média móvel.

Por exemplo, para $p = 1 e q = 0$ tem-se o modelo autoregressivo de primeira ordem ou AR(1), no qual o valor da série no instante t depende somente do valor da série no instante t-1:

$$Z_t = \phi_1 y_{t-1} + \varepsilon_t \qquad (3.10)$$

Lembrando que $BZ_t = Z_{t-1}$, a equação acima pode ser escrita como:

$$(1-\phi 1B)Zt = \varepsilon t. \qquad (3.11)$$

Em um caso mais geral tem-se o modelo auto-regressivo de ordem p AR(p) ou ARMA(p,0) , no qual a observação corrente Z_t , depende de realizações anteriores como $Z_{t-1}, Z_{t-2}, ... Z_{t-p}$ da mesma série temporal (Moretin & Toloi, 2006).

$$Z_t = \sum_{j=1}^{p} \phi_j Z_{t-j} \qquad (3.12)$$

Para se aplicar a metodologia de Box & Jenkins, a série em estudo deve ser pelo menos estacionária de segunda ordem, ou seja, a série tem que ter média, variância e covariância finitas e constantes. O exame de estacionaridade pode ser realizado observando-se a Função de Autocorrelação (ρ_k) ou FAC da série temporal investigada. Em função das correlações entre os valores de uma série temporal (Hippert, 2005), o valor de Z_t depende probabilisticamente do valor de Z_{t-1}. Dessa forma, a previsão se torna possível por causa dessa dependência entre os valores, sendo possível fazer estimativas dos valores futuros da série. Tal correlação entre os valores da série é chamada de autocorrelação.

A função de autocorrelação (FAC) mostra a dependência entre os diversos termos da série. Seu gráfico é chamado de correlograma e mostra a autocorrelação de lag k,

entre Z_t e Z_{t-k}, para diferentes valores da defasagem k no tempo. Matematicamente, a de defasagem k pode ser definida como:

$$\rho_k = \frac{\gamma_k}{\gamma_0} = Cov \frac{\left[Z_t, Z_{t+k}\right]}{\sqrt{Var\left(Z_t\right)Var\left(Z_{t+K}\right)}} \qquad (3.13)$$

Um decaimento lento da FAC indica que a série não é estacionária na média e precisa passar por uma ou mais diferenciações, se necessário.

Além da FAC também útil analisar o comportamento da Função de Autocorrelação Parcial (FACP), onde na medição da correlação entre duas observações seriais $\mathbf{Z_{t+1}}$ e $\mathbf{Z_{t+k}}$ elimina-se a dependência dos termos intermediários $\mathbf{Z_{t+1}, Z_{t+2}, Z_{t+k-1}}$,:

$$\phi_{KK} = Cov\ (Z_t, Z_{t+k} \mid Z_{t+1}, \ldots, Z_{t+k-1}) \qquad (3.14)$$

A análise da FAC e da FACP é de fundamental importância para o procedimento de previsão de séries temporais, pois é com ela que são identifica das ordem p e q de um modelo ARMA.

No processo auto-regressivo AR, a FAC terá um decaimento exponencial ou em senoidal amortecida, se $\boldsymbol{\varphi_1} < 0$, os sinais serão alternados. A FACP terá picos significativos nos *lags* 1 até p, depois cai a zero. Para um AR de ordem 1 – AR(1) – o pico da FACP será no *lag* 1 depois cai a zero. O pico será positivo se $\boldsymbol{\varphi_1} > 0$ e negativo se $\boldsymbol{\varphi_1} < 0$. No caso de um processo de médias móveis – MA(q) – a FAC terá picos nos *lags* 1 até q e depois cairá a zero. Em se tratando de um MA(1), o pico será no *lag* 1, caindo depois a zero. Caso $\boldsymbol{\theta_1} < 0$ será positivo o pico e se $\boldsymbol{\theta_1} > 0$ será negativo. Na FACP há a presença de um decaimento exponencial ou senoidal amortecida. Para recapitular, tem-se um resumo abaixo na Tabela 6:

Tabela 6-Características principais dos modelos AR(p), MA(q) e ARMA (p,q).

Características	AR(p)	MA(q)	ARMA(p,q)
Estrutura do Modelo	$\phi(B).\ Zt = at$	$Zt = \theta(B)\ .\ at$	$\phi(B)\ .\ Zt = \theta(B)at$
Função de Auto-correlação ρk(FAC)	Infinita (Exponenciais/ e/ou Senóides Amortecidas)	Finita (corte após lag "q")	Infinita (Exponenciais e /ou Senóides Amortecidas)
Função de Auto-correlação Parcial ϕkk (FACP)	Finita (corte após lag "p")	Infinita (Exponenciais/ e/ou Senóides Amortecidas)	Infinita (Exponenciais e/ou Senóides Amortecidas)

Fonte: adaptado de Souza & Camargo (1996)

Para Souza & Camargo (1996), uma das características fundamentais da metodologia de Box e Jenkins é interpretar uma dada série temporal como sendo uma realização de um vetor aleatório multivariado, cuja dimensão é a da série temporal disponível. A partir de uma única realização do processo e, com os argumentos de estacionaridade e ergodicidade do processo subjacente, procura-se detectar o sistema gerador da série, através de informações contidas na mesma. A filosofia da modelagem de Box & Jenkins se utiliza de duas idéias: o princípio da parcimônia e a construção de modelos por meio de um ciclo iterativo. O princípio da parcimônia estabelece que deve-se escolher um modelo com o menor número possível de parâmetros, para uma adequada representação matemática. Um ciclo iterativo é uma estratégia de seleção de modelos a ser empreendida até que tenha-se um modelo satisfatório.

Se a série temporal em estudo apresentar uma componente de tendência, então o processo estocástico gerador da série é não estacionário. Neste caso a série deve passar por d diferenças simples para tornar-se estacionária, condição básica para a aplicação da metodologia Box & Jenkins. Por exemplo, para remover uma tendência linear basta tomar a primeira diferença da série (d=1):

$$\Delta y = Z_t - Z_{t-1} \tag{3.15}$$

Caso a primeira diferença não seja estacionária, o operador diferença deverá ser aplicado na série obtidas pelas diferenças simples e uma segunda filtragem é efetuada, a qual poderá ser repetida quantas vezes necessárias, até tornar a série estacionária.

O processo de diferenciação ΔZ consecutiva de d vezes é realizado conforme apresentação na expressão 3.15, até que se obtenha uma série $\Delta^d Z$ estacionária e que possa ser modelada por um modelo ARMA (p,q), que será descrita a seguir.

$$\Delta Z_t = Z_t - Z_{t-1}$$

$$\Delta^2 Z_t = \Delta^{d-1} Z_t - \Delta Z_{t-1} \qquad (3.16)$$

.

.

.

$$\Delta^d Z_t = \Delta^{d-1} Z - \Delta^{d-1} Z_{t-1}$$

Neste caso, a metodologia Box & Jenkins é aplicada na série resultante das diferenciações e o modelo é denominado autoregressivo – média móvel – integrado ou ARIMA (p,d,q), onde d representa a ordem das diferenças simples:

$$\left(1-\phi_1 B-...-\theta_p \phi^p\right)(1-B)^d y_t = \left(1-\theta_1 B-...-\theta_q B^q\right)\varepsilon_t \qquad (3.17)$$

Como antes, a identificação das ordens dos termos auto-regressivos p e de médias móveis q baseia-se na análise do perfil das Funções de Autocorrelação (FAC) e Autocorrelação Parcial (FACP), porém da série obtida após as d diferenciações.

O modelo ainda pode ser adaptado para ser aplicável em séries sazonais. No caso geral, as séries temporais podem apresentar componentes sazonais e não sazonais. Neste caso, o processo estocas tico pode ser descrito pelo modelo SARIMA$(p,d,q)(P,D,Q)_s$ expresso pela seguinte equação:

$$\left(1-\phi_1 B-...-\phi_p B^p\right)\left(1-\Phi_1 B^S-...-\Phi_P B^{PS}\right)\left(1-B^S\right)^D\left(1-B^d\right)y_t = \left(1-\theta_1 B-...-\theta_q B^q\right)\left(1-\Theta_1 B^S-...-\Theta_Q \right. \qquad (3.18)$$

onde,

- p e q são, respectivamente, os graus dos polinômios das partes autoregressiva e de média móvel da componente não sazonal;
- P e Q são, respectivamente, os graus dos polinômios das partes autoregressiva e de média móvel da componente sazonal;
- d é a ordem das diferenças simples para remover a tendência da série
- D é a ordem das diferenças sazonais para remover a sazonalidade.
- S é o período sazonal, por exemplo, para séries discretizadas mensalmente S=12.
- ϕ e θ são, respectivamente, os coeficientes das partes autoregressiva e de média móvel da componente não sazonal;
- Φ e Θ são, respectivamente, os coeficientes das partes autoregressiva e de média móvel da componente sazonal.

A diferenciação sazonal visa remover a sazonalidade da série. Dado o período sazonal S, a diferenciação sazonal é:

$$\Delta_S^D y_t = y_t - y_{t-S} \tag{3.19}$$

Cabe ressaltar que o processo de obtenção desse modelo segue os mesmos passos empregados para achar o modelo ARIMA não sazonal (Zanini, 2007). No modelo ARIMA(p,d,q)(P,D,Q)s, a metodologia Box & Jenkins é aplicada na série supostamente estacionária e sem sazonalidade resultante das diferenciações simples e sazonais.

Em qualquer modelo da família Box & Jenkins, a identificação das ordens dos termos auto-regressivos (p e P) e de médias móveis (q e Q) baseia-se na análise do perfil da FAC e da FACP. A FAC também mostrará se há componente sazonal, o que pode ser observado quando esta segue um padrão periódico de picos e depressões.

A FAC e a FACP tem comportamentos teóricos conhecidos, entretanto na prática, como estas funções são estimadas, a análise dos gráficos da FAC e da FACP amostrais pode ser uma tarefa difícil, o que compromete a identificação precisa da ordem do modelo. A incorporação de coeficientes adicionais (modelos de maior ordem) melhora o grau de ajustamento, reduzindo a soma dos quadrados dos resíduos, no entanto, ressalta-se que modelos mais parcimoniosos produzem melhores previsões (maior capacidade de generalização) que os modelos sobreparametrizados.

Existem vários critérios de seleção de modelos que permitem encontrar um ponto de equilíbrio entre a redução na soma do quadrado dos resíduos e a parcimônia do modelo. Os critérios mais usados são o AIC (*Akaike Information Criterion*) e o BIC (*Bayesian Information Criterion*), cujas fórmulas são dadas por:

$$AIC = T\ ln\ (\sigma^2_{\varepsilon,ML}) + 2n \quad (3.20)$$

$$BIC = T\ ln\ (\sigma^2_{\varepsilon,ML}) + n\ ln(T) \quad (3.21)$$

onde, *n* é o número de parâmetros estimados, *T* é o número de observações da série temporal e σ^2_{ε}, ML é a estimativa de máxima verossimilhança de ε_t (Morettin & Toloi, 2006).

Comparando-se os valores de AIC e BIC de modelos com diferentes ordens, o melhor modelo é o que apresenta os menores valores nestas duas estatísticas.

No entanto, a seleção do melhor modelo não deve se basear apenas nos critérios AIC e BIC, a análise dos resíduos de modelos alternativos (concorrentes) ajustados é de extrema importância na escolha final do modelo que melhor explica a dinâmica da série temporal em estudo.

Se os resíduos são autocorrelacionados, então a dinâmica da série em estudo não é completamente explicada pelos coeficientes do modelo ajustado. Deve-se excluir do processo de escolha modelos com esta característica. Uma análise da existência (ou não) da autocorrelação serial de resíduos é feita com base na estatística Q de Box-Pierce-Ljung, (Souza e Camargo, 1996), representada formalmente como:

$$Q_{BPL} = T \cdot (T+2) \cdot \sum_{j=1}^{K} \frac{r_j^2}{T-j}, \quad (3.22)$$

onde, r_j é a autocorrelação de ordem j dos resíduos do modelo estimado et:

$$r_j = \frac{\sum_{t=j+1}^{T} e_t e_{t-j}}{\sum_{t=1}^{T} e_t^2} \quad (3.23)$$

A estatística Q_{BPL} é utilizada para testar se um conjunto de autocorrelações dos resíduos até a ordem K é (ou não) estatisticamente diferente de zero. Se os dados da série estudada são gerados por um processo estacionário, então a estatística Q_{BPL} tem distri-

buição qui-quadrado com K graus de liberdade. Observa-se que valores altos das autocorrelações dos resíduos implicam em valores altos de Q_{BPL}. Por outro lado, em um ruído branco todas as autocorrelações são nulas e Q_{BPL} é nulo. As considerações acima permitem testar as seguintes hipóteses com base na estatística Q_{BPL}:

H_0: as K primeiras autocorrelações são nulas.

H_1: de que pelo menos uma autocorrelação, r_j, é estatisticamente diferente de zero.

Um procedimento recomendado para identificar a melhor ordem de um modelo ARIMA em obter um modelo inicial a partir da análise das estimativas da FAC e da FACP e em seguida fazer o teste da sobrefixação (Souza & Camargo, 1996), onde são realizadas várias análises para diferentes valores de p, d, q, P, D, Q a partir do modelo inicial, em especial o teste t-student para verificar a significância do coeficiente de cada termo adicional na ordem do modelo.

Além das especificações acima, o modelo ARIMA pode ser adaptado para outras situações específicos e desta forma aumentar a sua aplicabilidade. Por exemplo, Chen et al.(2010) propõem um modelo estocástico para previsão da produção de energia eólica com base no modelo auto-regressivo média móvel integrado (ARIMA), que leva em conta a não estacionariedade da série temporal e limites físicos da geração estocástica de energia eólica. Os autores propõem a introdução de um limitador no modelo ARIMA para representar o limite superior e inferior da geração de energia eólica, o modelo proposto denominado por LARIMA foi ajustado a uma série histórica com medições durante um ano realizadas no parque eólico *offshore Nysted* na Dinamarca.

3.4
Regressão Harmônica

Na análise de série temporal, a regressão harmônica faz a aproximação de uma função do tempo por meio da combinação linear de harmônicos (componentes senoidais), cujos coeficientes são as transformadas de Fourier discretas da série (Morettin & Toloi, 2006). A análise harmônica é uma das ferramentas para análise de séries temporais no domínio da frequência. Como a série da velocidade de vento apresenta um comportamento sazonal, a análise harmônica por meio da combinação de funções trigonométricas é uma das técnicas indicadas para a modelagem da sazonalidade.

Na regressão harmônica a variável aleatória Z_t é expressa como sendo uma combinação de funções trigonométricos mais um ruído ε :

$$Z_t = \mu + \sum_{n=1}^{H}(A_n \cos n(w_n \cdot t) + B_1 \operatorname{senn}(w_n \cdot t)) + \varepsilon_t \ , \qquad (3.24)$$

onde, μ é a média da de Z_t;

n = 1, 2, 3 ... H identifica o número de harmônicos incluídos no modelo; A_n e B_n são os respectivos coeficientes das funções trigonométricas cosseno e seno para o harmônico "n" das séries de Fourier;

$w_n = \frac{2\pi n}{N}$ é a frequência do n-ésimo harmônico.

N é o período, ou seja o número de dados observados.

t_i - ordenação numérica das horas do vento correspondentes da série (i = 0, 1, ...)

Os coeficientes, μ, **A** e **B** são obtidos, respectivamente, pelas seguintes expressões (Morettin & Toloi, 2006):

$$\mu = \overline{Z} = \frac{1}{N}\sum_{i=1}^{N} Z_i \qquad (3.25)$$

$$A_n = \frac{2}{N}\sum_{i=1}^{N} Z_i \cos(w_i t) \qquad (3.26)$$

$$B_i = \frac{2}{N}\sum_{i=1}^{N} Z_i sen(w_i t) \text{ e } B_n = 0 \quad \text{para } w=\pi \qquad (3.27)$$

A especificação do número de harmônicos no lado direito da equação de regressão harmônica baseia-se na análise do periodograma para a determinação da freqüência w. Os estimadores μ, **A** e **B** dependerão de w e, portanto, serão denotados respectivamente por $\hat{\mu}(w), \hat{A}(w) e \hat{B}(w)$.

As frequências w devem minimizar a soma de quadrados residual, SQR, dada pela expressão 3.28:

$$SQR \cong SQT - \frac{N}{2}\tilde{R}^2 , \qquad (3.28)$$

ou equivalentemente, maximizar a quantidade

$$\tilde{R}^2(w) = \tilde{A}^2(w) + \tilde{B}^2(w), \tag{3.29}$$

com $\tilde{A}(w)\, e\, \tilde{B}(w)$ dadas pelas expressões 3.30 e 3.31

$$\tilde{A} = \frac{2}{N}\sum_{i=1}^{N}(Z_t - \bar{Z})\cos(wt) \tag{3.30}$$

$$\tilde{B} = \frac{2}{N}\sum_{i=1}^{N}(Z_t - \bar{Z})sen(wt) \tag{3.31}$$

O que, é equivalente a maximizar a quantidade:

$$I(w) = \frac{N}{8\pi}\tilde{R}^2(w) \tag{3.32}$$

$$I(w) = \frac{1}{2\pi N}\left[\left(\sum_{t-1}^{N}\left(Z_t - \bar{Z}\right)\cos w_t\right)^{\wedge 2} + \left(\sum_{t-1}^{N}\left(Z_t - \bar{Z}\right)sen w_t\right)^{\wedge 2}\right], \tag{3.33}$$

denominada periodograma.

Assim, estima-se w maximizando $\tilde{R}^2(w)$ na expressão 3.29 ou, equivalentemente, maximizando o periodograma da equação obtemos os demais estimadores do modelo utilizando as expressões abaixo:

$$\mu = \frac{\sum_{t-1}^{N} Zt}{N} = \bar{Z}, \tag{3.34}$$

$$\hat{A} = \frac{2}{N}\sum_{t-1}^{N} Z_t \cos(w_t), w \neq \pi, \tag{3.35}$$

$$\hat{A} = \frac{2}{N}\sum_{t=1}^{N} Z_t(-1)^t \text{ e } \hat{B} = 0, se\ w = \pi, \tag{3.36}$$

Para mais detalhes sobre regressão harmônica consulte Morettin (2006) e Toloi (2006). A análise harmônica pode ser combinada com a abordagem de Box& Jenkins na modelagem de séries temporais com múltiplos ciclos de sazonalidade, como é o caso das séries horárias de velocidade de vento analisadas nesta dissertação, onde o ajuste de um modelo auto-regressivo média móvel (ARMA) é precedido pela análise harmônica com a finalidade de remover os múltiplos ciclos sazonais. O método ARMA é aplicado

na modelagem da série dos resíduos resultantes da diferença entre a série da velocidade do vento e a série estimada pela regressão harmônica.

3.5 Rede Neural Artificial

Uma rede neural artificial (*RNA*) é um sistema de computação composto de elementos processadores (EPs) altamente interligados, trabalhando em paralelo para desempenhar uma determinada tarefa. Estes elementos processadores, linspirados nos neurônios biológicos, são organizados de tal forma que podem, em alguns casos, lembrar a anatomia do cérebro. Contudo, os EPs são bem mais simples que suas inspirações naturais e contêm basicamente apenas um algoritmo matemático que executa o processamento da informação em resposta a estímulos procedentes de outros EPs (Haykin, 2001).

O cérebro humano é composto por cerca de 100 bilhões de células nervosas, conhecidas por neurônios, que se conectam massivamente umas as outras através de ligações eletroquímicas, denominadas sinapses, formando uma grande rede de processamento. Cada neurônio recebe estímulos através dos dendritos, os processa em seu corpo celular e, dependendo do seu estado de ativação, gera e transmite um estímulo pelo seu axônio para que atinja outros neurônios. A estrutura e o funcionamento do neurônio biológico podem ser modeladas pelo neurônio artificial ilustrado na Figura 6.

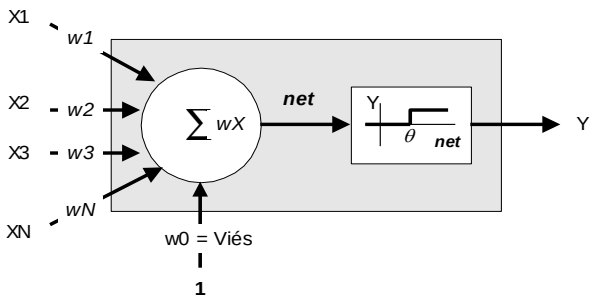

Figura 6-Neurônio artificial de McCulloch & Pitts

No neurônio artificial os *N* terminais de entrada representam os dendritos e o único terminal de saída representa o axônio. A intensidade das sinapses é representada pelos pesos (*w*) nos terminais de entrada, cujos valores podem ser negativos ou positivos,

definindo sinapses inibidoras ou excitadoras respectivamente. De forma análoga ao que acontece no cérebro humano, onde as sinapses são reforçadas ou enfraquecidas, os pesos são ajustados durante a evolução do processo de aprendizagem da rede.

O comportamento do corpo celular é emulado por um mecanismo que faz a soma ponderada dos valores recebidos (cálculo do *net*). No modelo mais simples, esta soma ponderada é comparada com um limiar θ. Neste modelo, a função de ativação que relaciona a resposta do neurônio com a soma ponderada dos valores recebidos é uma função degrau: se $x_1 w_1 + x_2 w_2 + ... + x_N w_N \geq \theta$ o neurônio é ativado e produz uma saída igual a 1, caso contrário o neurônio não dispara e a saída é zero. Enfim, o corpo celular é responsável pelo mapeamento dos sinais de entrada em um único sinal de saída. No lugar da função degrau, a função de ativação pode assumir diferentes formas, em geralmente não-lineares, o que transformam as redes neurais em sistemas computacionais capazes de resolver problemas complexos. Assim, destacam-se as seguintes funções de ativação:

- Função linear: os neurônios com esta função de ativação podem ser utilizados como aproximadores lineares;
- Função Logística sigmoidal: mapeia os sinais de entrada dos neurônios no intervalo [0,1]. É a função geralmente adotada, por ser contínua monotônica, não linear e facilmente diferenciável em qualquer ponto;
- Função tangente hiperbólica: mapeia os sinais de entrada dos neurônios no intervalo [-1,+1]. Possui as mesmas características e emprego da função logística sigmoidal, possibilitando que as saídas sejam simétricas.

As RNA são sistemas paralelos distribuídos, compostos por unidades de processamento simples (neurônios) dispostas em uma ou mais camadas que são interligadas por um grande número de conexões geralmente unidirecionais e com pesos para ponderar a entrada recebida por cada neurônio. Através de uma analogia com o cérebro humano, pode-se afirmar que os pesos das conexões armazenam o conhecimento ou a memória da rede neural.

A organização dos vários neurônios artificiais em uma estrutura e a forma de como eles se interligam define a arquitetura de uma RNA. A arquitetura mais usual é a rede perceptron de múltiplas camadas ou *Multilayer Perceptron* (MLP) com três camadas, conforme mostra a Figura 7.

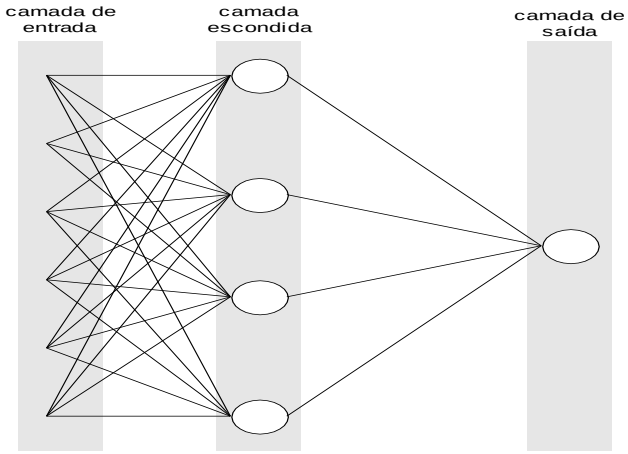

Figura 7-Rede neural com três camadas

A primeira camada da rede é a camada de entrada, sendo a única camada exposta às variáveis de entrada. Esta camada transmite os valores das variáveis de entrada para os neurônios da camada escondida, para que estes extraiam as características relevantes ou padrões dos sinais de entrada. Por sua vez a camada intermediária transmites os resultados para a camada de saída, a última camada da rede.

A construção de um modelo baseado em RNA envolve a busca da melhor configuração para a rede, i.e., a definição do número adequado de camadas escondidas e a definição da quantidade ideal de processadores nestas camadas. A inexistência de regras que definam claramente a configuração adequada faz desta busca um processo empírico e que depende da distribuição dos padrões de entrada, da quantidade de ruído presente nos exemplos de treinamento e da complexidade da função a ser aprendida (Haykin, 2001). Diferentes configurações devem ser avaliadas, entretanto, deve-se sempre empregar o princípio da parcimônia, e saber que uma rede com apenas uma camada oculta é capaz de aproximar qualquer tipo de função contínua (Haykin, 2001), embora em algumas situações específicas sejam usadas duas camadas.

Uma RNA possui duas fases de processamento: aprendizado e utilização.

No processo de aprendizado, os pesos de conexão são ajustados em resposta ao estímulo apresentado à rede neural, ou seja, a rede se modifica em função da necessidade de aprender a informação que lhe foi apresentada. O processo de utilização é a maneira pelo qual a rede responde a um estímulo de entrada sem que ocorram modificações na sua estrutura (Caldeira et al, 2007). Segundo Treleaven (1989), o processo de

aprendizagem ocorre através de um processo interativo de ajuste dos parâmetros livres, pesos sinápticos e por estimulação do ambiente.

Os paradigmas de aprendizado são: aprendizado supervisionado e aprendizado não supervisionado descritos resumidamente a seguir.

- **Aprendizado Supervisionado:** Esta forma de aprendizado se baseia em um conjunto de exemplos de entrada-saída que é apresentada a rede. A partir da entrada, a rede realiza seu processamento e a saída obtida é comparada com a saída esperada. Caso não sejam iguais, um processo de ajuste de pesos é aplicado buscando-se um erro mínimo ou aceitável. O algoritmo de aprendizado supervisionado mais comum é o *backpropagation* (Haykin, 2001).
- **Aprendizado não supervisionado**: É caracterizado pela ausência de algum elemento externo supervisor, ou seja, um padrão de entrada fornecido permite que a rede livremente escolha o padrão de saída a partir das regras de aprendizado adotadas. Possui duas divisões: aprendizado por reforço, que consiste no mapeamento entrada-saída através da interação com o ambiente, e aprendizagem não-supervisionada ou auto-organizada onde, a partir de métricas de qualidade do aprendizado ocorre a otimização dos parâmetros livres da rede. Pode, por exemplo, ser utilizada a regra de aprendizagem competitiva. Os algoritmos de aprendizado não supervisionado mais importantes são: Algoritmo de Hopfield e Mapas de Kohonen (Haykin, 2001).

A aprendizagem supervisionada é comumente aplicada na previsão de séries temporais, enquanto a aprendizagem não supervisionada é usual na análise de agrupamentos (*cluster analysis*).

A previsão de valores futuros de uma série temporal, por meio de uma RNA (Werbos, 1990), inicia-se com a montagem do conjunto de treinamento, que depende da definição do tamanho da janela de tempo para os valores passados das variáveis explicativas e da própria variável que se deseja prever, bem como do horizonte de previsão.

O padrão de entrada é formado pelos valores passados das variáveis explicativas que podem incluir os valores passados da própria série que se deseja prever (modelo auto-regressivo) e a saída desejada é o valor da série temporal no horizonte de previsão. A Figura 8 ilustra como deve ser construído o conjunto de treinamento no caso da previsão basear-se nos quatro últimos valores passados. A construção dos padrões de trei-

namento da rede consiste em mover as janelas de entrada e saída ao longo de toda série temporal:

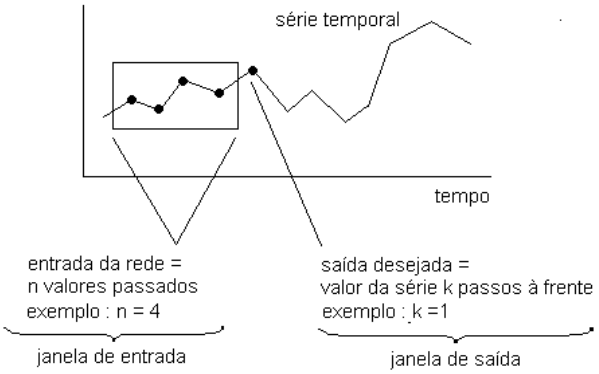

Figura 8- Montagem do conjunto de treinamento

Cada par de janelas entrada/saída funciona como um padrão de treinamento e deve ser apresentado repetidas vezes até que o algoritmo de aprendizado alcance a convergência.

A arquitetura da RNA exerce grande influência sobre o desempenho do processo de aprendizagem. Em uma rede neural pequena há dificuldade de armazenar todos os padrões necessários e por isso a convergência do algoritmo de treinamento é mais lenta. Em uma rede pequena os processadores ficam sobrecarregados e lidam com muitas restrições na tentativa de encontrar uma representação ótima. Porém, deve-se ter o cuidado de não utilizar processadores demais, pois a rede pode memorizar os padrões de treinamento, ao invés de extrair as características gerais que permitirão o reconhecimento de padrões não vistos durante o treinamento.

Com relação às redes com mais de uma camada escondida é importante observar que cada vez que o erro é propagado para a camada anterior, ele se torna menos útil e preciso. Apenas a camada imediatamente anterior à camada de saída tem uma noção precisa do erro, todas as camadas escondidas anteriores recebem uma estimativa do erro. Por esta razão a convergência dos pesos destas camadas é mais lenta.

O processo de treinamento de uma rede neural é nada mais que o ajuste de parâmetros, guiado pelo processo de minimização da função do erro entre as saídas desejadas e as apresentadas pela rede. Durante o processo, vários padrões de entrada e as respectivas saídas desejadas são apresentados à rede neural, de tal forma que os pesos

das sinapses sejam corrigidos iterativamente pelo algoritmo do gradiente decrescente com o objetivo de minimizar a soma dos quadrados dos erros:

$$E = \frac{1}{2}\sum_{p}\sum_{j=1}^{n}(d_j^p - y_j^p)^2, \qquad (3.37)$$

onde, p - o número de padrões de treinamento (padrões de entrada e saída);

n - o número de neurônios da camada de saída;

d_j - é a saída desejada para o j-ésimo neurônio da camada de saída;

y_j - é a saída gerada pelo j-ésimo neurônio da camada de saída.

O principal algoritmo de treinamento é o *backpropagation*, onde o ajuste dos pesos se dá pela execução de um processo de otimização realizado em duas fases: *forward* e *backward*, conforme mostra a Figura 9,

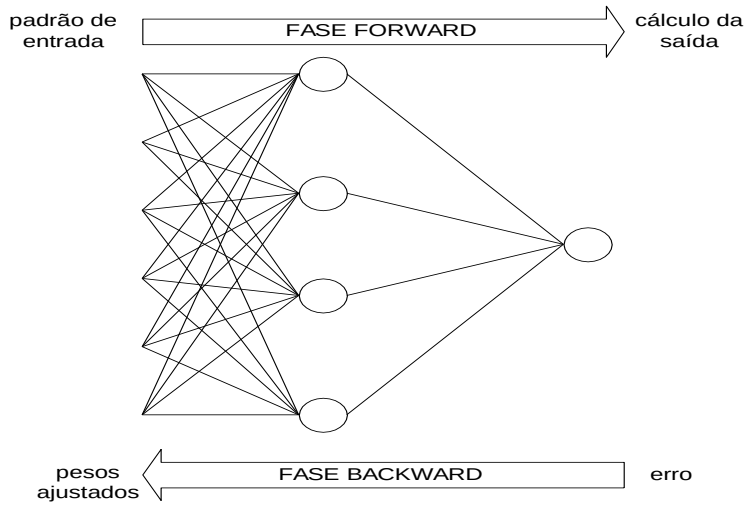

Figura 9-Rede neural com três camadas

Na fase *forward* os dados de entrada alimentam a rede e são propagados para frente até que as saídas dos nós da última camada sejam obtidas, considerando-se fixos todos os parâmetros da rede.

Já na fase *backward*, o desvio (erro) entre a resposta desejada (alvo) e a resposta efetivamente fornecida pela rede é utilizado para ajustar os pesos das conexões da rede. O sinal de erro é propagado na direção da camada de entrada e o gradiente decrescente é usado para ajustar os parâmetros da rede.

Para minimizar a soma dos quadrados do erro o algoritmo *backpropagation* se baseia no método gradiente descendente, por isso, afim de que esse método seja utilizado a função de ativação precisa ser contínua, diferenciável e de preferência não decrescente, por exemplo, a função logística. O algoritmo *backpropagation* pode ser resumido nas seguintes operações (Haykin, 2001):

1º Passo - Inicialize aleatoriamente os pesos da rede e faça o contador de épocas (t) igual a zero.

2º Passo - Apresente uma época de exemplos de treinamento à rede. Uma época indica o número de vezes que o conjunto de treinamento, ou melhor, os padrões de entrada (x) e o respectivo padrão de saída (yd) são apresentados à rede. Para cada exemplo de treinamento realizar os passos 3 e 4 a seguir.

3º Passo – Fase forward: Propague o padrão de entrada (Xp) para frente, camada por camada, até chegar na camada de saída. Para cada neurônio calcular a combinação dos sinais recebidos da camada anterior:

$$net_j^I(t) = \sum_{i=0}^{m} w_{j,i}^I(t) \cdot y_i^{I-1}(t) \tag{3.38}$$

onde, $y_i^{I-1}(t)$ é a saída produzida pelo neurônio i da camada anterior I-1 e $w_{j,i}^I(t)$ é o peso sináptico da conexão do neurônio j na camada I com o neurônio i da camada anterior I-1.

Para i=0 tem-se o viés aplicado ao neurônio j na camada I: $y_0^{I-1}(t) = 1$ e $w_{j,0}^I(t) = b_j^I(t)$.

Se o neurônio j está na primeira camada oculta, i.e., I=1 $y_i^0(t) = x_j(t)$.

Denotando por f_j a função de ativação do neurônio j, o sinal produzido por ele é:

$$y_i^I(t) = f_j\left(net_j(t)\right). \tag{3.39}$$

No final calcule o erro, ou seja, a diferença entre a resposta desejada e a resposta fornecida pela rede:

$$e_j(t) = y_j^d - y_j \tag{3.40}$$

4º Passo – Fase backward: Propague o erro calculado no passo 2 para trás, começando na camada de saída e terminando na camada de entrada. Neste processo ajuste os pesos conforme a expressão :

$$w_{j,i}^{I}(t+1) = w_{j,i}^{I}(t) + \eta \delta_j^I(t) y_i^{I-1}(t) \tag{3.41}$$

onde, η é a taxa de aprendizagem, $y_i^{I-1}(t)$ é a resposta do neurônio i situado na camada anterior I-1 e $\delta_j^I(t)$é o gradiente local do neurônio j da camada I, definido de acordo com a localização do neurônio na rede.

Se a camada I onde está o neurônio é uma camada de saída tem-se:

$$\delta_j^I(t) = e_j(t) \cdot \frac{df_j(net_j(t))}{dnet_j} \tag{3.42}$$

Porém, se a camada I onde o neurônio estiver é uma camada escondida, o gradiente local é:

$$\delta_j^I(t) = \frac{df_j(net_j(t))}{dnet_j} \cdot \sum_{k=1}^{m} \delta_k^{I+1}(t) \cdot w_{kj}^{I+1}(t) \tag{3.43}$$

onde, m é número de neurônios da camada I+1, $\delta_k^{I+1}(t)$ é o gradiente local do neurônio k situado na camada I+1 e $w_{kj}^{I+1}(t)$ é o peso sináptico da conexão entre o neurônio j na camada I e o neurônio k na camada I+1

5º Passo – Após terminar uma época de exemplos faça *t=t+1* e itere para frente e para trás os passos 3 e 4 e pare apenas quando o critério de parada[1] for satisfeito.

O *backpropagation* usa o algoritmo do gradiente descendente durante na otimização dos pesos das sinapses. Um aprimoramento do gradiente descendente é o algoritmo

[1] Usualmente o critério de parada fixa um determinado número de iterações ou estabelece uma tolerância para o erro.

de Levenberg-Marquardt o qual propõe uma solução de compromisso entre o algoritmo do gradiente decrescente e o método iterativo de Gauss-Newton. Sua regra de atualização dos pesos é:

$$x_{i+1} = x_1 - (H + \lambda I)^{-1} \nabla f(x_i) \qquad (3.44)$$

onde, x - representa o vetor de pesos;

∇f - representa gradiente de erro médio quadrático;

H - representa a matriz Hessian;

λ - um fator de ajuste.

Assim, a regra de atualização leva em consideração tanto a inclinação da superfície do erro (método do gradiente decrescente) quanto à curvatura desta superfície (método de Gauss-Newton). O fator de ajuste indica qual dos dois métodos será predominante: para fatores de ajuste grandes, o método do gradiente decrescente predomina e a atualização dos pesos ocorre fortemente na direção de inclinação da superfície do erro; caso contrário, o método de Gauss- Newton predomina e a atualização ocorre mais no sentido da curvatura da função.

Finalmente, é bom ressaltar que o problema encarado pelo algoritmo LM é exatamente o que ocorre no treinamento *backpropagation*, onde a função erro a ser minimizada é não linear.

Para mais informações sobre método de redes neurais podem ser consultados em Klir (1995) e Haykin (2001).

3.6 Redes Neuro-Fuzzy

Trata-se da fusão de duas ferramentas já conhecidas: redes neurais artificiais e a lógica *f*uzzy, no qual agregam-se as características de transparência de raciocínio da lógica *fuzzy* juntamente com a capacidade de aprendizado e generalização das redes neurais.

Assim uma rede *Neuro-Fuzzy* pode ser definida como um sistema *fuzzy* que é treinado como uma rede neural. Tendo em vista esta analogia, a união da rede neural com a lógica *fuzzy* vem com o intuito de amenizar a deficiência de cada um destes sistemas fazendo com que tenhamos um sistema mais eficiente, robusto e de fácil entendimento.

O problema das redes neurais está basicamente relacionado à falta de poder explicativo do sistema. Como forma de tentar solucionar estes problemas, foi criado os sistemas *Neuro-Fuzzy*. A principal vantagem deste sistema é associar a capacidade de aprendizado das Redes neurais e sua tolerância a falhas à interpretabilidade dos sistemas *fuzzy*.

Existem vários sistemas *Neuro-Fuzzy*, das quais podemos citar:

- ANFIS- *Adaptative Network Fuzzy Inference System* (JANG 1993);
- NEFCLASS – *Neuro-Fuzzy Classification* (NAUCK 1994);
- FSOM- *Fuzzy*- Self organized Map (VUORIMAA 1996);
- NFH- *Neuro-Fuzzy* Hierárquico (SOUZA 1997).

Para esse trabalho foi aplicado o *Adaptative Network Fuzzy Inference System* (ANFIS), uma vez que estamos tratando de dados de séries temporais. A seguir, tem-se uma breve descrição do sistema de inferência *fuzzy* já que o sistema em estudo é fundametado no sistema *fuzzy*.

3.6.1
Sistema de Inferência Fuzzy

Na lógica *fuzzy* o grau de verdade de uma declaração é representado por um número real no intervalo [0,1], ao contrário do que ocorre na lógica clássica em que o grau de verdade assume apenas dois valores: 0 (declaração falsa) e 1 (declaração verdadeira). Esta característica da lógica *fuzzy* é útil em muitas situações práticas onde o grau de intensidade de um fenômeno é descrito de maneira imprecisa por meio de variáveis lingüísticas: baixo, moderado baixo, médio, moderado alto ou alto. Exemplos desta situação são as sentenças temperatura baixa, temperatura normal e temperatura alta, onde a separação entre os conjuntos, por exemplo, normal e alta não é precisa. A principal contribuição da lógica *fuzzy* reside no tratamento destas questões lingüísticas por meio de funções de pertinência aos conjuntos *fuzzy*, conforme ilustrado a seguir na Figura 10 para a variável temperatura.

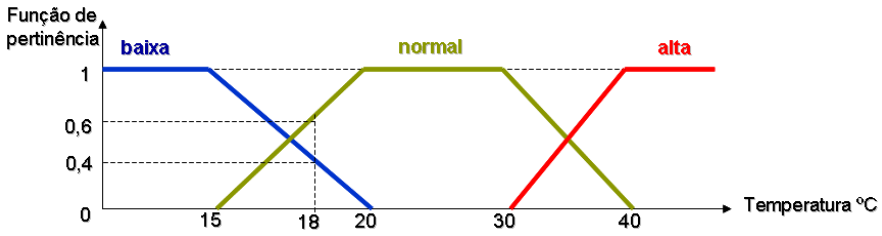

Figura 10-Conjuntos fuzzy e funções de pertinência para a variável temperatura

Na Figura 10 a variável temperatura foi modelada por três conjuntos *fuzzy* que representam as situações de temperatura baixa, normal e alta. Estes três conceitos lingüísticos não são delimitados de forma precisa, pois estão associados com a sensação térmica. Esta característica é representada pela sobreposição entre os conjuntos *fuzzy* para determinadas temperaturas. Por exemplo, uma temperatura de 18ºC é baixa ou normal? A lógica *fuzzy* responde esta questão atribuindo um grau de pertinência de 0,6 ao conjunto *fuzzy* temperatura normal e 0,4 ao conjunto *fuzzy* temperatura baixa e desta forma consegue tratar um conceito definido de forma imprecisa.

A teoria dos conjuntos *fuzzy* e os conceitos de lógica *fuzzy* podem ser utilizados para traduzir em termos matemáticos a informação imprecisa expressa por um conjunto de regras lingüísticas, sentenças fornecidas por um especialista e expressas através de implicações lógicas da forma *SE antecedente ENTÃO conseqüente* (Pacheco & Vellasco, 2007).

O processo de inferência *fuzzy* avalia os níveis de compatibilidade das variáveis de entradas com os antecedentes das várias regras, ativando os conseqüentes com intensidades proporcionais aos mesmos. O resultado desta operação é um conjunto *fuzzy* que é convertido em um número, a resposta do sistema de inferência *fuzzy*.

A estrutura de um sistema de inferência *fuzzy* é ilustrada na Figura 11 e na sequência são descritas as funções de cada um dos seus elementos

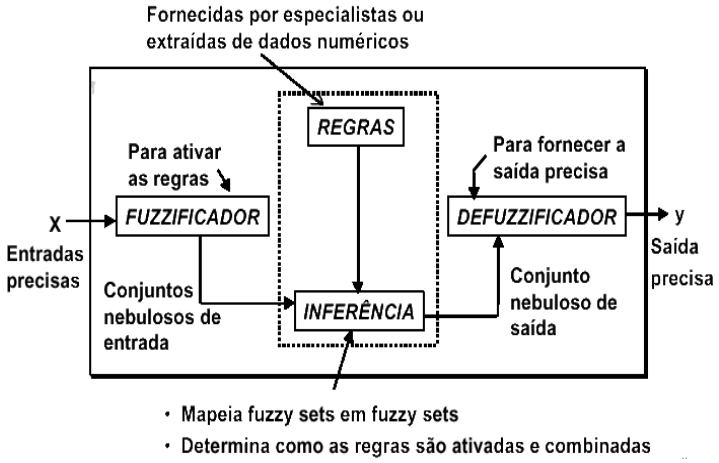

Figura 11-Sistema de inferência fuzzy ou FIS (Pacheco &Vellasco, 2007)

Fuzzificador: Mapeia os valores das variáveis de entrada (valores crisp) nos conjuntos *fuzzy* dos antecedentes das regras *fuzzy*. Isso se faz necessário de forma a ativar regras que estão no formato de variáveis lingüísticas, as quais possuem conjuntos *fuzzy* associados com elas (Klin & George, 1995).

Base de regras: As regras são fornecidas em geral por especialistas ou extraídas a partir dos dados, na forma de sentenças lingüísticas "se-então" (Caldeira, 2007).

Inferência: Realiza as operações lógicas com conjuntos *fuzzy*, combinação dos antecedentes das regras, implicação e modus ponês generalizado.

Defuzzificador: Transforma um conjunto *fuzzy* de saída em um elemento do universo de discurso (em geral, um número real), ou seja, o inverso da fuzzificação. Entre as técnicas utilizadas para tal processo, a mais usual é o do centróide (Caldeira et al. 2007).

A seguir, a Figura 12 ilustra o princípio de raciocínio da lógica *fuzzy* em um sistema de inferência *fuzzy* tipo Mandani com duas regras, cujos antecedentes são definidos pela composição de dois conjuntos *fuzzy* A e B e que representam o comportamento das variáveis de entrada x e y respectivamente. Cada regra oferece como resposta um conjunto *fuzzy* de saída C:

$$\text{Se x é } A_1 \text{ e y é } B_1 \text{ então z} = C_1$$
$$\text{Se x é } A_2 \text{ e y é } B_2 \text{ então z} = C_2 \qquad (3.45)$$

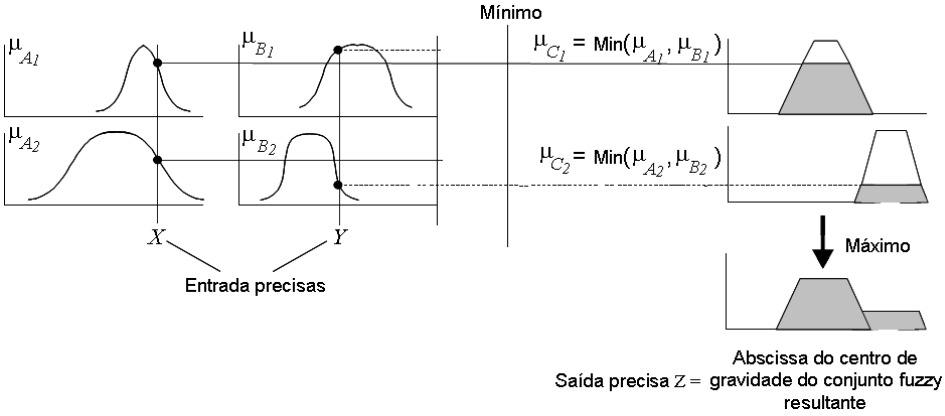

Figura 12-Modelo Mandani

Fonte: Jang, 1997

No modelo Mandami o processamento é denominado inferência Max-Min e corresponde às operações de união e interseção *fuzzy* (operadores máximo e mínimo). Os antecedentes de cada regra são processados por meio da interseção *fuzzy*, gerando um grau de disparo que limitará os valores máximos dos conjuntos de saída. Por exemplo, na Figura 12 o valor preciso da variável X tem pertinência μA1 no conjunto *fuzzy* A1 e o valor da variável Y tem pertinência μB1 no conjunto *fuzzy* B1. Logo, pela inferência Max-Min o grau de ativação da regra é min(μA1, μB1) que neste caso é μA1.

A composição de todas as regras disparadas (ou ativadas) é realizada através da operação de união *fuzzy* que gera o conjunto *fuzzy* de saída. Para obter uma saída precisa deve-se proceder à defuzzyficação do conjunto de saída. Há diversos métodos para realizar a transformação dos conjuntos *fuzzy* de saída em valores numéricos, tais como a Média dos Máximos e o Centro de Massa (também denominado Centro de Gravidade ou Centróide).

Uma alternativa ao modelo Mandani é o modelo Takagi-Sugeno-Kang ou TSK (Jang,1993 e Sun 1995) ilustrado na Figura 13, no qual cada regra oferece como resposta uma combinação linear das variáveis de entrada, sendo que a saída do sistema de inferência *fuzzy* é a média ponderada das respostas parciais, onde os pesos são os graus de ativação das regras ‘w’ e que expressam a compatibilidade das variáveis de entrada x e y com os antecedentes das regras. O modelo TSK pode ser visto como uma combinação

entre conhecimento lingüístico (parte antecedente) e regressão estatística (parte conseqüente), de tal forma que os antecedentes descrevem regiões nebulosas no espaço de entrada nas quais as funções conseqüentes são válidas. Uma regra típica de um sistema com duas variáveis de entrada utilizando o sistema TSK tem a forma:

$$\text{Se x é A e y é B então } z = px + qy + r \qquad (3.46)$$

No caso em p = q = 0 , temos z = r , chamado modelo TSK de ordem zero, que pode ser visto como um caso especial de um modelo de Mandani no qual o conseqüente é especificado por um conjunto unitário (singleton).

Como cada regra possui uma saída convencional, a saída global é obtida através da média ponderada de todos os resultados de saída, considerando-se os graus de pertinência de cada regra ativada:

$$y = \frac{\sum_{i=1}^{N} \mu_i . y_i}{\sum_{i=1}^{N} \mu_i} \qquad (3.47)$$

onde, y é a saída final, N representa o total de regras ativadas, e μ_i é o grau de pertinência em relação à contribuição de cada regra ativada.

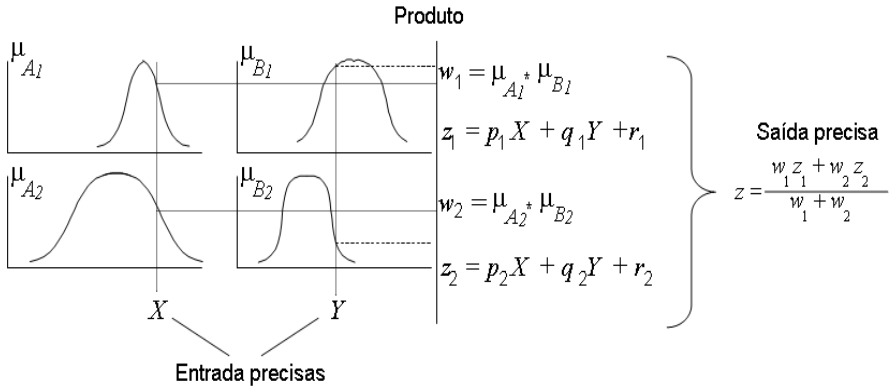

Figura 13-Modelo Takagi-Sugeno-Kang

Fonte: Jang et al, 1997

3.6.2
Sistema Neuro-Fuzzy Adaptativo (ANFIS)

O **ANFIS** é uma rede neural proposta por Jang (1993) cuja idéia básica é de implementar um sistema de inferência *fuzzy* através de uma arquitetura paralela distribuída, neste caso, a de uma RNA, de tal forma que os algoritmos de aprendizado possam ser usados para ajustar este sistema de inferência *fuzzy*.

Os parâmetros associados com as funções de pertinência são ajustados via um algoritmo de aprendizado. O ajuste destes parâmetros é efetuado utilizando o algoritmo de *backpropagation* ou uma combinação deste com um algoritmo do tipo: mínimos quadrados (*Least Squares*). Esta estrutura implementa sistemas do tipo Takagi-Sugeno (Takagi-Sugeno, 1985), com funções lineares ou constantes nos consequentes das regras que formam o sistema, tendo estas regras pesos unitários.

A rede adaptativa é uma espécie de grafo com nós interconectada por ramos direcionados. Alguns dos nós apresentam comportamento adaptativo, ou seja, sofrem alteração paramétrica no decorrer do treinamento, enquanto outros mantêm seu comportamento dinâmico inalterado (Caldeira, 2007).

O método une as várias partes de um sistema de inferência *fuzzy* em uma rede adaptativa *feedforward* com cinco camadas (Figura 14) e treinada de modo supervisionado.

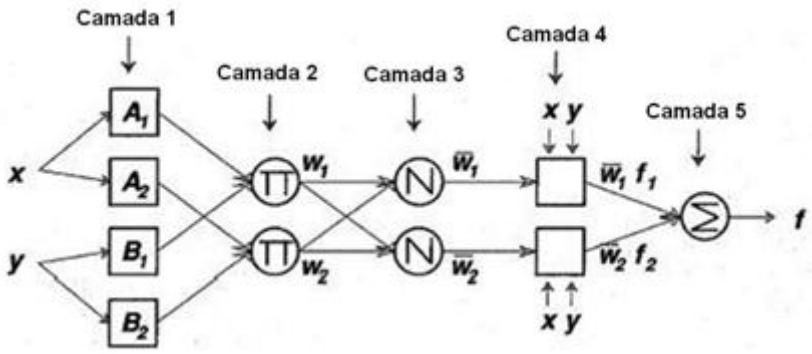

Figura 14-Arquitetura da ANFIS.

Fonte: Jang, 1993

A título de ilustração considere duas entradas x e y e uma saída z. No contexto da previsão de séries temporais, as variáveis x e y correspondem aos valores passados da variável que se deseja prever z(t), z(t-1),... ou valores passados de variáveis explicativas. Suponha que a base de regras contenha duas regras *fuzzy* "se-então":

Regra 1: Se x é A_1 e y é B_1 então $f_1=p_1x + q_1y + r_1$ (3.48)

Regra 2: Se x é A_2 e y é B_2 então $f_2=p_2x + q_2y + r_2$

onde , A_1 e A_2 são os conjuntos *fuzzy* da variável x e B_1 e B_2 os conjuntos *fuzzy* da variável y. Destaca-se que o conseqüente de cada regra é uma combinação linear das variáveis de entrada (x e y) e corresponde a uma previsão para o valor da variável de saída z, portanto, cada regra fornece uma previsão para a variável de saída.

Na camada 1, cada nó representa um conjunto *fuzzy* de uma variável de entrada (x ou y) e como resultado fornece o grau de pertinência μ do valor de entrada no conjunto *fuzzy*:

$Saída_i^1 = \mu A_i(x)$, grau de pertinência do valor da variável x no conjunto *fuzzy* A_i, i=1,2

$Saída_i^1 = \mu B_i(y)$, grau de pertinência do valor da variável y no conjunto *fuzzy* Bi, i=1,2

O grau de pertinência μ da entrada nos conjuntos *fuzzy* A e B pode ser definido por funções triangulares, trapezoidais, gaussianas, mas usualmente é descrito pela função sino generalizada:

$$\mu A_i(x) = \left(1 + \left|\frac{x-b_i}{a_i}\right|^{2c_i}\right)^{-1}, i = 1,2 \qquad \mu B_i(x) = \left(1 + \left|\frac{x-e_i}{d_i}\right|^{2f_i}\right)^{-1}, i = 1,2$$

(3.49)

Onde, ($a_i.b_i.c_i$) e ($d_i.e_i.f_i$) são conjuntos de parâmetros (*premise parameters*) ajustados durante o treinamento da rede.

Na camada 2 cada nó Π calcula o grau de ativação de uma regra *fuzzy*, definido pelo produto entre os graus de pertinência das variáveis de entrada nos conjuntos *fuzzy* que formam os antecedentes das regras:

$Saída_i^2 = w_i = \mu A_i(x) \cdot \mu B_i(y) =$ grau de ativação da i-ésima regra *fuzzy* i=1.2, (3.50)

A camada 2 não tem pesos a serem ajustados, é uma camada com elementos estáticos.

Na camada 3, cada nó N normaliza o grau de ativação de uma regra *fuzzy* dividindo o grau de ativação da i-ésima regra pela soma dos graus de ativação de todas as regras:

$$Saída_i^3 = \overline{w}_i = \frac{w}{w_1 -} \qquad \text{i=1,2} \tag{3.51}$$

O valor normalizado do grau de ativação fornece uma medida da importância de cada regra *fuzzy*, quanto maior o valor normalizado, maior a importância da respectiva regra. A camada 3 também é estática.

Na camada 4, cada nó calcula a resposta de uma regra *fuzzy*, ou seja, uma previsão para o valor da variável z, definida por uma combinação linear das variáveis de entrada:

$$Saída_i^4 = \overline{w}_i\left(p_i x + q_i y \right. \qquad \text{i=1,2} \tag{3.52}$$

onde, (p_i,q_i,r_i) são parâmetros (*consequent parameters*) a serem ajustados durante o treinamento.

Por fim, na camada 5, uma camada fixa, o único neurônio $\sum$ calcula a média ponderada das previsões parciais para a variável de saída, onde cada previsão parcial é ponderada pelo grau de ativação da respectiva regra *fuzzy*:

$$Saída_i^5 = \sum_j \overline{w}_j \left(p_j x + q_j y + r_j\right) = \frac{\sum_j w_j \left(p_j x + q_j y\right.}{\sum_j w_j} \tag{3.53}$$

No ajuste dos *premise* e *consequent parameters* a ANFIS usa o método dos mínimos quadrados para determinar os *consequent parameters* e a retropropagação do erro

(método do gradiente descendente) para aprender os *premise parameters* (Jang et al, 1997).

A rede adaptativa tem um funcionamento equivalente ao modelo de TSK.

O procedimento de previsão da ANFIS é similar da rede neural. Tem-se duas formas de previsão:

- Previsões *multi-step*

- Previsões s*ingle-step*

As previsões *multi-step* são aquelas que se caracterizam por possuir realimentação das saídas das *RNA*s para as entradas das mesmas. Neste tipo de previsão, o sistema neural usa um conjunto de valores correntes da série para prever os valores futuros desta série por um período fixo. Em seguida, esta previsão é realimentada na entrada do sistema para prever o próximo período. Estas previsões são muito usadas para identificar tendências e pontos de mudanças preponderantes nas séries. Devido ao erro que é inserido a cada nova previsão, o horizonte de previsões "*multi*-step" depende das características da série e do limite do erro estabelecido.

Nas previsões "single-step" não existe realimentação. As RNAs utilizam apenas os valores anteriores da série para prever um passo à frente. Todavia, este passo tanto pode ser para previsões de curto prazo como para previsões de médio e longo prazo, bastando que se tenha dados suficientes para treinar a rede. A previsão "single-step" também serve para avaliar a adaptabilidade e a robustez do sistema, mostrando que mesmo quando as RNAs fazem previsões erradas, elas são capazes de se auto corrigirem e fazer as próximas previsões corretamente.

3.7
Diagnostico do Modelo

Dada uma série histórica com n observações, a qualidade do ajuste e o desempenho de um modelo de previsão podem ser avaliados pelas seguintes estatísticas, onde O_t é o valor observado e E_t o valor estimado/previsto, ambos para o instante t. O desvio entre estes dois valores é o erro de previsão em t.

- Erro médio absoluto percentual (MAPE): $MAPE = \frac{100}{n}\sum_{t=1}^{n}\frac{|O_t - E_t|}{O_t}$
- Erro médio absoluto (MAD): $MAD = \sum_{t=1}^{n}\frac{|O_t - E_t|}{n}$
- Raiz do erro quadrático médio (RMSE) : $RMSE = \sqrt{\sum_{t=1}^{n}\frac{(O_t - E_t)^2}{n}}$
- U de Theil: $U\text{-}Theil = \frac{\sqrt{\sum_{t=2}^{n}\left(\frac{O_t - E_t}{O_{t-1}}\right)^2}}{\sqrt{\sum_{t=2}^{n}\left(\frac{O_t - O_{t-1}}{O_{t-1}}\right)^2}}$

A estatística U de Theil compara a previsão obtida pelo modelo de previsão com a obtida pelo método de previsão ingênuo (naive), no qual a previsão para o instante seguinte é o valor imediatamente anterior.