4
ANÁLISE DE SÉRIES TEMPORAIS

4.1.
Introdução

Neste capítulo definem-se as séries temporais e discutem-se as características básicas das técnicas de análise, com enfoque nos modelos de Box & Jenkins (1970) e em redes neurais artificiais. O objetivo não é apresentar tais modelos como alternativas para solução do problema, mas sim como técnicas complementares que podem e devem ser sempre que possível utilizados conjuntamente na previsão de séries temporais.

4.2.
Séries temporais

Uma série temporal é um conjunto de observações ordenadas no tempo, não necessariamente igualmente espaçadas, que apresentam dependência serial, isto é, dependência entre instantes de tempo. A notação usada aqui para denotar uma série temporal é $S_1, S_2, S_3 \ldots , S_T$ que indica uma série de tamanho T. Uma grande quantidade de fenômenos de natureza física, biológica, econômica, etc. pode ser enquadrada nesta categoria. A maneira tradicional de analisar uma série temporal é através da sua decomposição nas componentes de **tendência**, **ciclo** e **sazonalidade**. (Morettin, 1987).

A **tendência** de uma série indica o seu comportamento "de longo prazo", isto é, se ela cresce, decresce ou permanece estável, e qual a velocidade destas mudanças. Nos casos mais comuns trabalha-se com tendência constante, linear ou quadrática, como ilustrado na figura 4.1.

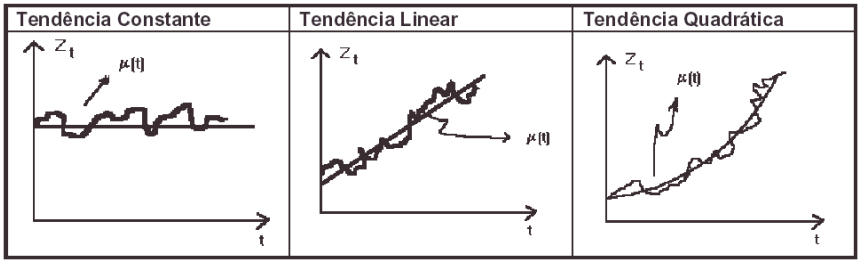

Figura 4.1 Tendências de uma série temporal (Barros, 2003).

Os **ciclos** são caracterizados pelas oscilações de subida e de queda nas séries, de forma suave e repetida, ao longo da componente de tendência. Por exemplo, ciclos relacionados à atividade econômica ou ciclos meteorológicos.

A **sazonalidade** em uma série corresponde às oscilações de subida e de queda que sempre ocorrem em um determinado período do ano, do mês, da semana ou do dia. A diferença essencial entre as componentes sazonal e cíclica é que a primeira possui movimentos facilmente previsíveis, ocorrendo em intervalos regulares de tempo, enquanto que movimentos cíclicos tendem a ser irregulares.

Em geral ao estudarmos uma série temporal estamos interessados em:

a) **Análise e modelagem da série temporal** - descrever a série, verificar suas características mais relevantes e suas possíveis relações com outras séries;
b) **Previsão na série temporal** - a partir de valores históricos da série (e possivelmente de outras séries também) procura-se estimar previsões de curto prazo (*forecast*). O número de instantes à frente para o qual é feita a previsão é chamado de horizonte de previsão.

4.2.1. Procedimentos estatísticos de previsão

Os procedimentos de previsão utilizados na prática variam muito, podendo ser simples e intuitivos, com pouca análise dos dados, ou complexos e racionais, envolvendo uma considerável trabalho de interpretação de séries temporais. Vale a pena ressaltar, no entanto, que a previsão não constitui um fim em si, mas deve ser vista como parte integrante de um complexo processo de tomada de decisão, visando a objetivos específicos. Dentre os procedimentos estatísticos de previsão podem ser citados:

- ***Modelos Univariados***: inclui os modelos que se baseiam em uma única série histórica. Como exemplo, podem ser citados: a) a decomposição por componentes não observáveis, que foi o mais utilizado até a década de 1960; b) os modelos automáticos que incorporam modelos de regressão, de médias móveis, ajustamento sazonal e alisamento exponencial; c) os modelos univariados de Box & Jenkins (1970) que consistem em uma classe geral de modelos lineares conhecidos como modelos ARIMA, os quais são explicados no item 4.3.
- ***Modelos de Função de Transferência***: nos quais a série de interesse é explicada não só pelo seu passado histórico, como também por outras séries temporais não correlacionadas entre si.
- ***Modelos Multivariados***: modelam simultaneamente duas ou mais séries temporais sem qualquer exigência em relação à direção da causalidade entre elas.

4.2.2. Estacionariedade de uma série

A estacionariedade numa série temporal significa que os dados oscilam sobre uma média constante, independente do tempo, com a variância das flutuações permanecendo essencialmente a mesma.

Diniz (1998) afirma que uma série temporal é estacionária se o processo aleatório oscilar em torno de um nível médio constante. Séries temporais sazonais ou com tendência linear ou exponencial são exemplos de séries temporais com comportamento não estacionário. A condição de estacionariedade de 2ª ordem implica em:

- Média do processo é constante;
- Variância do processo é constante.

No caso, das séries temporais consideradas nas aplicações deste trabalho, tanto a série da vazão no solo de fundação da ombreira esquerda como a maioria das séries históricas dos piezômetros do núcleo da barragem Corumbá-I apresentaram-se estacionárias.

4.2.3.
Análise de autocorrelação

Na adoção de um modelo para uma série temporal, seja considerando métodos estatísticos ou através de redes neurais artificiais, é necessário conhecer-se a relação entre as observações atuais e as anteriores. Uma forma de avaliá-la é através das funções de autocorrelação.

Seja s_t a representação, por exemplo, das leituras de poro-pressão de um piezômetro "A" em 6 instantes de tempo, conforme tabela 4.1, onde a coluna (3) representa uma série com atraso (*lag*) de 1 intervalo de tempo, correspondendo à série s_{t-1}, e a coluna (4) indica a série s_{t-2} com atraso (*lag*) de 2 intervalos de tempo.

(1) Tempo (ou período)	(2) Variável Original	(3) Variável com atraso (*lag*) de um tempo	(4) Variável com atraso (*lag*) de dois tempos
t	s_t	s_{t-1}	s_{t-2}
1	12	-	-
2	7	12	-
3	5	7	12
4	8	5	7
5	10	8	5
6	4	10	8

Tabela 4.1 Série temporal de poro-pressão em piezômetro "A". (Soto, 1999).

A autocorrelação entre as séries s_t e s_{t-1} (autocorrelação com lag 1) indicará como os valores de poro-pressão estão relacionados com seus valores imediatamente precedentes, enquanto que a autocorrelação entre s_t e s_{t-2} (autocorrelação com lag 2) fornecerá uma relação dos valores da série s_t com aqueles atrasados em dois intervalo de tempo.

Em geral, para uma série temporal com n elementos, a autocorrelação com atraso k é dada pela expressão:

$$r_k = \frac{\sum_{t=1}^{n-k} (s_t - \bar{s})(s_{t+k} - \bar{s})}{\sum_{t-1}^{n} (s_t - \bar{s})^2} \tag{4.1}$$

onde $\bar{s}$ é a média da série original de *n* elementos, admitida estacionária.

4.3.
Modelos ARIMA de Box & Jenkins

Os modelos ARIMA são modelos estatísticos lineares para análise de séries temporais. A abreviação em língua inglesa refere-se a "Auto-Regressive Integrated Moving Average model", ou seja, um modelo auto-regressivo integrado de médias móveis. Os termos auto-regressivos correspondem a defasagens da série transformada (isto é, série estacionária obtida por diferenciação) e as médias móveis a defasagens dos erros aleatórios. O termo "integrado" refere-se ao processo de diferenciação da série original para torná-la estacionária.

O modelo tem como premissa básica que a série temporal é gerada por um processo estocástico cuja natureza pode ser representada através de um modelo. A notação empregada para designação do modelo é normalmente ARIMA (p,d,q) onde p representa o número de parâmetros auto-regressivos, d o número de diferenciações para que a série torne-se estacionaria e q o número de parâmetros de médias móveis. Casos particulares são o modelo ARMA(p,q), o modelo auto-regressivo AR(p) e o modelo de médias móveis MA(q), todos para séries temporais estacionárias (d=0).

Os modelo auto-regressivo AR(p) é definido por,

$$S(t) = \sum_{i=1}^{p} \alpha_i S(t-i) + \varepsilon(t) = \phi_L(S(t-1),...,S(t-p)) + \varepsilon(t) \tag{4.2}$$

onde a estimativa da variável *S* para um tempo t depende de uma combinação linear de p termos da série observada, incluindo o termo aleatório ε(t) de ruído branco (erros de estimação com distribuição normal, média zero,

variância constante e não-correlacionados). Os coeficientes α_i são parâmetros que ponderam os valores de S_t, do instante imediatamente anterior *t-1* até o mais distante *t-p*, sendo determinados através de técnicas de minimização do erro. O modelo AR[p] é limitado pois assume a existência de uma relação linear entre os elementos da seqüência e baseia-se na hipótese de que a série é estacionária, isto é, a média e o desvio padrão das observações medidas não variam com o tempo.

Uma rede neural multicamada poderia ser utilizada para substituir a função linear ϕ_L da equação (4.2) por uma função não-linear ϕ_{NL} determinada através de um método de aprendizado, como o da retropropagação dos erros. Fazendo ϕ_{NL} dependente dos p elementos prévios da seqüência é equivalente a usar p unidades de entrada alimentadas por p elementos adjacentes. Este processo é conhecido como 'janelamento' temporal que será explicado mais adiante.

O modelo de médias móveis MA(q) assume que a série modelada é gerada através de uma combinação linear de q sinais de ruídos $\varepsilon(t-i)$, aleatórios e independentes entre si,

$$S(t) = -\sum_{i=1}^{q} \theta_i \varepsilon(t-i) + \varepsilon(t) = \phi_L(\varepsilon(t-1),...,\varepsilon(t-q)) + \varepsilon(t) \qquad (4.3)$$

A combinação dos modelos AR(p) e MA(q) dá então origem ao modelo ARMA(p,q), no qual

$$S(t) = \sum_{i=1}^{p} \alpha_i S(t-i) - \sum_{i=1}^{q} \theta_i \varepsilon(t-i) + \varepsilon(t) \qquad (4.4)$$

ou

$$S(t) = \phi_L(S(t-1),...,S(t-p),\varepsilon(t-1),...,\varepsilon(t-q)) + \varepsilon(t) \qquad (4.5)$$

As limitações previamente mencionadas no modelo AR(p), concernentes à linearidade e estacionariedade do fenômeno modelado, são também aplicáveis aos modelos MA(q) e ARMA(p,q).

4.3.1.
Etapas de modelagem

A aplicação dos modelos de Box & Jenkins para fins de previsão de séries temporais segue as seguintes etapas: identificação, estimação, verificação e previsão.

Na primeira fase o que se deseja é *identificar* o processo aleatório que gerou os dados, para em seguida *estimar* os parâmetros que o caracterizam e *verificar* se as hipóteses do modelo foram cumpridas. Caso negativo, uma nova fase de identificação deve ser considerada até que a verificação das hipóteses seja finalmente positiva, permitindo então a realização da *previsão*.

i) **Identificação do modelo**

Na etapa de identificação de um modelo ARIMA empregam-se procedimentos que possam identificar a estrutura do modelo, isto é, permitam conhecer os valores dos parâmetros d, p e q que caracterizam o processo estocástico.

Os procedimentos de identificação consistem de duas partes: a) inicialmente diferencia-se a série temporal original tantas vezes quantas necessárias para obtenção de uma série estacionária, de modo a possibilitar a análise do processo com o modelo ARMA(p,q); o número de diferenciações d é aquele necessário para que a função de autocorrelação amostral (ACF) da série transformada decresça rapidamente para zero; b) a identificação de um processo AR(p), MA(q) ou ARMA(p,q) é feita através da análise das funções de autocorrelação simples (ACF) e da autocorrelação parcial (PACF), com determinação dos valores dos parâmetros p, q (mais detalhes em Box & Jenkins, 1976; Morettin, 1987; Barros, 2003). A figura 4.2 mostra três exemplos do comportamento das funções de autocorrelação para processos auto-regressivo AR, de médias móveis MA e misto ARMA.

Um ponto importante na identificação do modelo ARIMA é a **parcimônia**, ou seja, deve-se buscar um modelo com melhor ajuste e menor número possível de parâmetros. Em outras palavras, dados dois modelos com ajustes igualmente bons, escolhe-se aquele com menor número de parâmetros. Existem duas justificativas para isso: i) simplicidade na estrutura identificada, o que permite

uma melhor compreensão do processo subjacente e ii) em geral modelos superparametrizados não são geralmente bons para previsões, pois se ajustam bem aos dados amostrais, mas não conseguem generalizar o comportamento de toda a série. Na verdade, o princípio da parcimônia é conceito geral, devendo ser aplicado a qualquer problema de modelagem estatística, e não apenas na utilização de modelos ARIMA.

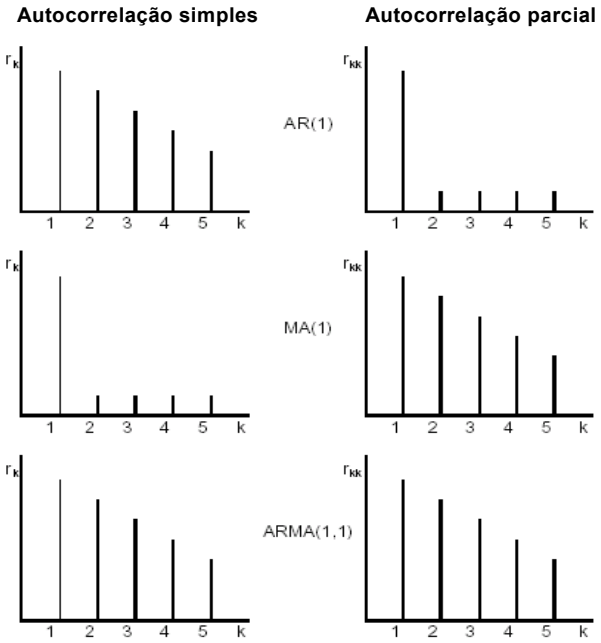

Figura 4.2 Funções de autocorrelação simples e parcial para alguns processos estocásticos AR(1), MA(1) e ARMA(1,1).

ii) Estimativas dos parâmetros

Após a identificação dos valores d, p, q passa-se à estimativa dos parâmetros do modelo. Esta etapa é geralmente executada através de "software" específico para análise de séries temporais como o programa computacional E-Views 4, utilizado no desenvolvimento desta pesquisa. Existem diferentes métodos para a estimativa dos parâmetros do modelo, e todos produzem resultados semelhantes, embora possam existir diferenças quanto à eficiência na implementação computacional. De maneira geral, durante a etapa de estimativa usa-se algum procedimento iterativo de estimação de mínimos quadrados não linear ou através dos métodos de máxima verossimilhança aproximada, máxima, etc. (Morettin, 1987).

iii) Verificação através de testes estatísticos

Ao ajustarmos um modelo ARIMA, estamos buscando capturar toda a estrutura de dependência serial presente na série. Logo, os resíduos do modelo não devem apresentar qualquer tipo de dependência no tempo. Num modelo ARIMA bem ajustado, os erros de previsão um passo à frente (resíduos) devem ser uma seqüência de ruídos brancos, ou seja, não devem exibir autocorrelações (e autocorrelações parciais) significativas. Outros testes estatísticos executados nesta etapa envolvem verificações da condição de estacionariedade, análise de significância dos parâmetros estimados e critérios de seleção do modelo (critério de Akaike, critério de Schwartz). O leitor interessado deve consultar textos específicos sobre modelos de Box & Jenkins para aprofundar conhecimentos sobre testes estatísticos em séries temporais.

iv) Previsão de valores

Uma vez identificado o processo estocástico que gera a série temporal de interesse e os parâmetros do modelo, passa-se então à etapa de previsão de valores futuros (*forecast*).

4.4. Redes neurais artificiais

Há pouco mais de uma década, o uso das redes neurais artificiais (RNA) vem sendo continuamente incrementado na solução de problemas geotécnicos. Uma revisão da literatura revela que as RNA estão sendo usadas com ótimos resultados na previsão da capacidade de carga de estacas, na modelagem do comportamento de solos, na identificação da estratigrafia do subsolo, na estabilidade de taludes, em problemas de liquefação, na investigação da permeabilidade de solos e em problemas de condutividade hidráulica, compactação, classificação de solos, etc. Shahin et al. (2001), Diminsky (2000), God (1995), entre outros, mostram um panorama geral das possíveis aplicações de redes neurais na engenharia geotécnica.

As RNA são sistemas paralelos distribuídos compostos por unidades de processamento simples denominados neurônios que calculam determinadas funções matemáticas normalmente não-lineares. Tais unidades são dispostas em uma ou mais camadas e interligadas por um grande número de conexões, geralmente unidirecionais. Na maioria dos modelos estas conexões estão associadas a pesos, os quais armazenam o conhecimento representado no modelo e servem para ponderar a entrada recebida por cada neurônio da rede. O funcionamento destas redes é inspirado em uma estrutura física concebida pela natureza: o cérebro humano.

A solução de problemas através de RNA é bastante atrativa, já que a forma como estes são representados internamente pela rede e o paralelismo natural inerente à arquitetura das RNA, criam bastante favoráveis à obtenção de um bom desempenho em relação aos modelos determinísticos convencionais. Na técnica de redes neurais o procedimento usual na solução de problemas passa inicialmente por uma fase de *aprendizagem,* na qual um conjunto de exemplos é apresentado à rede, que extrai automaticamente as características necessárias para representar a informação fornecida. Estas características são utilizadas posteriormente para gerar as respostas do problema sendo analisado (Haykin, 1999; Bishop, 1995; Zurada, 1992).

A capacidade de *aprender* através de exemplos e de *generalizar* a informação aprendida é, sem dúvida, o atrativo principal da solução de problemas através de redes neurais. A generalização, associada à habilidade da rede ser treinada através de um conjunto reduzido de exemplos e posteriormente estar apta a fornecer respostas coerentes para dados desconhecidos, é uma demonstração de que a capacidade da rede vai muito além do que simplesmente mapear relações de entrada e saída. As RNA são capazes de extrair informações apresentadas de forma não-explícita através de exemplos.

As RNA são capazes de atuar como mapeadores universais de funções multivariadas, com custo computacional que cresce apenas linearmente com o número de variáveis. Outra característica importante é a capacidade de auto-organização e de processamento temporal que, aliada àquelas citadas anteriormente, faz das RNA uma ferramenta computacional extremamente poderosa e atrativa para a solução de problemas complexos (Braga et al., 2000).

Uma estrutura típica das RNA consiste de um número de unidades de processamento, usualmente arranjadas em camadas: uma camada de entrada, uma camada de saída e uma ou mais camadas ocultas. A entrada x_i de cada neurônio é multiplicada por uma conexão ajustável denominado peso sináptico w_{ik}. Para cada neurônio, os pesos devido aos estímulos de entrada são somados e um valor de "*bias"* (b_k) é adicionado. Esta combinação de entradas υ_k passa através de uma função de transferência não linear *f(.)* para gerar a saída y_k. A saída de um neurônio vem ser posteriormente a entrada dos neurônios das camadas seguintes. Este processo é resumido nas equações (4.6) e (4.7) e ilustrado na figura 4.3.

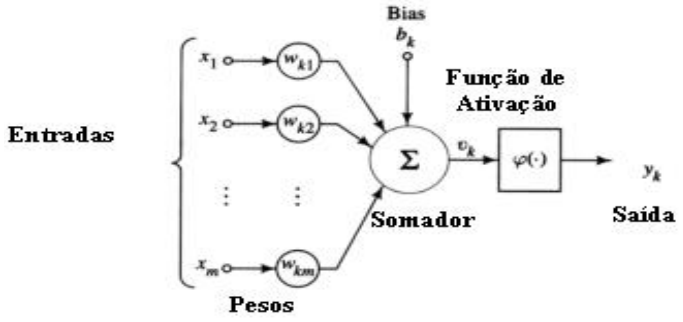

Figura 4.3 Modelo de neurônio artificial. (Haykin, 1999).

Combinação linear:

$$\upsilon_k = \Sigma w_{ki} * x_i + b_k \tag{4.6}$$

Função de transferência:

$$y_k = f(\upsilon_k) = \varphi(\upsilon) \tag{4.7}$$

Cada unidade de processamento está associada a um estado de ativação, que pode ser discreto ou contínuo. Quando a função de ativação é não linear a equação (4.7) pode ser avaliada por uma função sigmoidal que, por sua vez, pode assumir as seguintes formas mais usuais:

Função logística:

$$\varphi(\upsilon) = \frac{1}{1 + e^{-a\upsilon}} \tag{4.8}$$

Função tangente hiperbólica:

$$\varphi(\upsilon) = \frac{e^{a\upsilon} - e^{-a\upsilon}}{e^{a\upsilon} + e^{-a\upsilon}} = \tanh(a\upsilon) \tag{4.9}$$

Onde *a* é uma constante cujo valor determina a inclinação da curva, com os valores da função de ativação φ variando entre [0,1] se for do tipo logístico ou entre [-1,1] para a tangente hiperbólica. A função de ativação precisa ser diferenciável, para que o algoritmo de retropropagação tenha coerência, e também seja não decrescente, para que a sua derivada não troque de sinal e venha a comprometer a convergência do algoritmo.

A figura 4.4 apresenta uma rede neural artificial com 3 camadas totalmente conectadas com 6 padrões de entrada, 4 neurônios na camada oculta e 1 neurônio na saída.

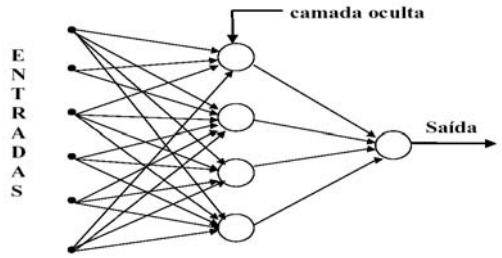

Figura 4.4 Rede neural multicamada totalmente conectada (MLP).

4.4.1.
Redes neurais na previsão de séries temporais

Entre as redes neurais convencionais, as MLP (*Multilayer perceptrons)* com algoritmo de aprendizado de retropropagação demonstraram capacidade de realizar mapeamentos dinâmicos. Entretanto, muitos pesquisadores dedicaram-se a obter arquiteturas de redes neurais artificiais adequadas para este tipo de mapeamento. Para que uma rede neural possa ter características dinâmicas e

realizar tarefas de caráter temporal é preciso que tenha propriedades de memória. A forma como a memória é representada em uma rede neural determina os diferentes tipos de RNA temporais. Segundo Soto (1999), existem duas formas de se incluir habilidades de memória na rede neural (figura 4.5): i) a rede considera entradas atrasadas no tempo; ii) a rede tem laços de realimentação.

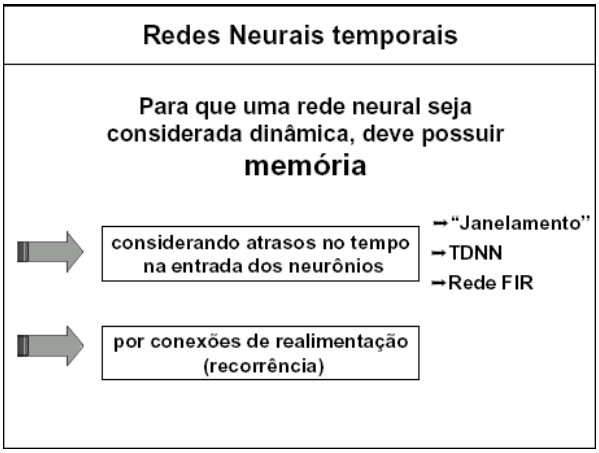

Figura 4.5 Classificação das redes neurais temporais. (Soto, 1999).

4.4.1.1. Redes neurais com atrasos no tempo

Os modelos de redes neurais mais utilizados para processamento temporal foram por muito tempo as redes com atraso no tempo. A idéia sempre foi introduzir memória à rede fornecendo aos neurônios valores atuais de entrada e valores temporalmente anteriores. A Figura 4.6 mostra um método básico, denominado de 'janelamento', onde só se introduz memória nos neurônios da primeira camada escondida. Além da entrada atual $x_{(k)}$, estes neurônios recebem também como entrada dois valores anteriores $x_{(k-1)}$ e $x_{(k-2)}$ criando, portanto, sinapses novas. Assim, o método do 'janelamento' proporciona à rede uma memória de ordem 2 na primeira camada escondida.

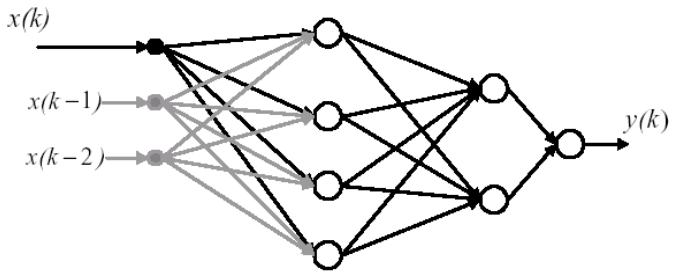

Figura 4.6 Método de 'janelamento' para processamento temporal. Soto (1999).

Outro modelo de rede com atraso no tempo e de caráter mais geral, é a denominada rede TDNN (*Time Delay Neural Network*), ilustrada na Figura 4.7. Esta rede proporciona memória a todos os neurônios das camadas escondidas e da camada de saída.

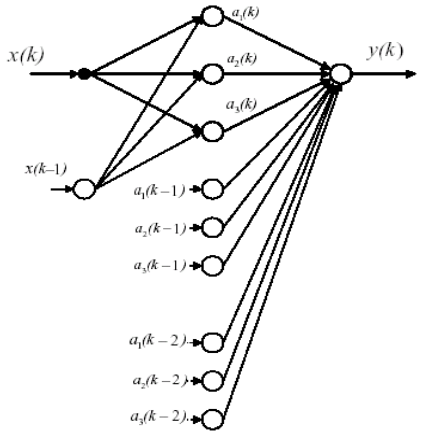

Figura 4.7 Rede TDNN para processamento temporal (Soto, 1999).

Como se observa na figura, a rede tem uma camada escondida (pode ter mais de uma, no entanto) com memória de ordem 1, com a camada de saída tendo memória de ordem 2. De maneira similar ao método de 'janelamento', existem sinapses novas, mas a rede final formada não conduz à forma da rede estática padrão totalmente interconectada, pois os neurônios novos na camada escondida não estão conectados aos elementos de entrada. Este fato faz a rede TDNN perder certa "simetria" no processo da exploração "*feedforward*" e de retropropagação, tornando-a mais complexa (Soto, 1999).

Outro tipo de rede é a denominada rede neural FIR (*Finite Response Impulse*) onde cada sinapse é agora formada por um filtro FIR linear que

representa a natureza temporal do problema. Pode-se dizer que este tipo de rede engloba todos os outros métodos de sua classe.

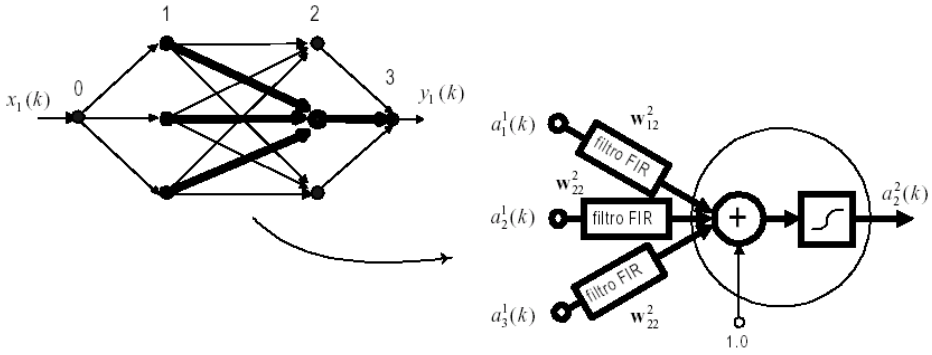

Figura 4.8 Neurônio FIR numa rede multicamada (Soto, 1999).

Dorffner (1996) faz uma revisão das RNA no processamento de séries temporais, destaca como estas podem representar os modelos estatísticos tradicionais no caso de previsão e generaliza, elegantemente, sua aplicação em relação à capacidade de aproximar funções arbitrárias não lineares. A figura 4.9 ilustra como as redes MLP com 'janelamento' temporal seguem um processo autoregressivo AR não linear. A notação F^{NN} indica uma função não linear.

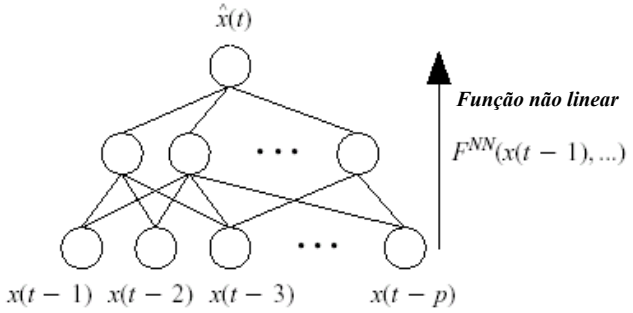

Figura 4.9 Rede neural *feedforward* com janela temporal incorporando o modelo autoregressivo AR(p) não linear (Dorffner, 1996).

Este modelo é mais poderoso do que o modelo autoregressivo linear clássico podendo modelar fenômenos não-lineares e não-estacionários, mas requer um grande número de exemplos durante a fase de aprendizado devido ao elevado número de graus de liberdade presentes. Além disso, problemas no aprendizado

podem ocorrer, como excesso de treinamento (*overfitting*), não convergência do processo iterativo, etc., os quais são mais severos do que no caso linear. Por estas razões, em muitas aplicações reais onde os dados disponíveis são limitados, muitas vezes opta-se pela utilização de um modelo linear mesmo quando a dependência não-linear do fenômeno for reconhecível.

4.4.1.2. Redes neurais recorrentes

São aquelas que possuem conexões de realimentação para simulação de comportamento dinâmico, conforme esquema da figura 4.10. As redes neurais recorrentes são redes que possuem uma ou mais conexões de realimentação. A realimentação pode ser *local,* se situada em nível de neurônio, ou *global,* se alguma(s) camada(s) completa(s) forem envolvidas. Na prática, são duas as maneiras que as redes recorrentes podem ser utilizadas: a) m*emórias associativas;* b) *mapeamento entrada-saída,* sendo a segunda maneira mais adequada para o processamento de séries temporais.

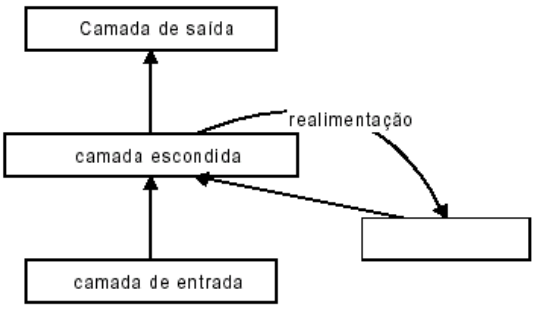

Figura 4.10 Esquema de processamento temporal utilizando redes neurais recorrentes (Soto 1999).

Um tipo de rede neural temporal é a denominada Elman, que possui realimentação global interna, onde cada um dos neurônios escondidos tem realimentação para as unidades de contexto, conforme figura 4.11.

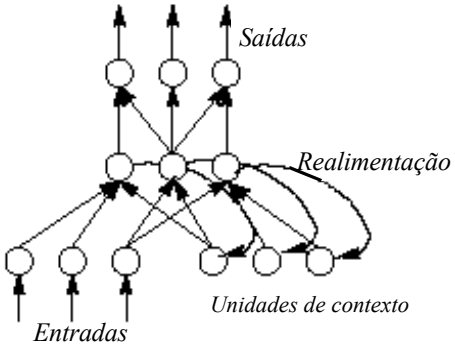

Figura 4.11 Rede neural tipo Elman. (Dorffner, 1996).

Diferentemente das anteriores, as redes tipo Jordan consideram realimentação dos valores de ativação de saída para as unidades de contexto (figura 4.12a). Dorffner (1996) mostra como as redes recorrentes tipo Jordan podem representar um modelo ARMA com características não lineares (figura 4.12b).

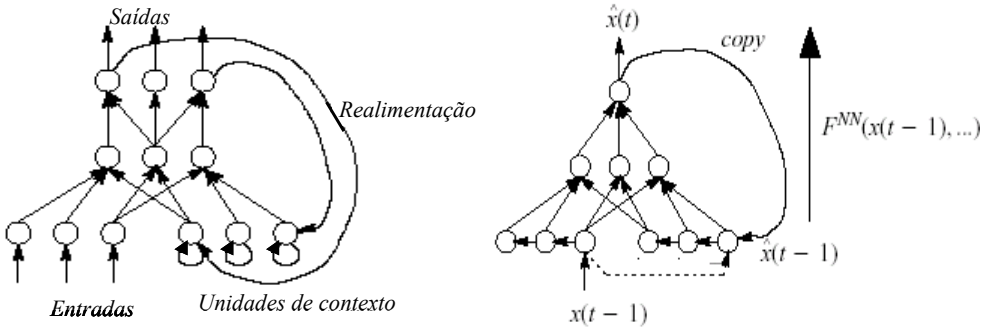

Figura 4.12 Redes recorrentes tipo Jordan. a) Rede tipo Jordan, esquema geral. b) Rede tipo Jordan simulando um processo ARMA não linear. (Dorffner, 1996).

4.4.2. Etapas de uma análise por redes neurais

Uma análise por redes neurais envolve várias etapas seqüenciais, a saber, coleta de dados, pré-processamento, escolha do modelo e definição da topologia de treinamento, pós-processamento e avaliação dos resultados, conforme esquema da figura 4.13.

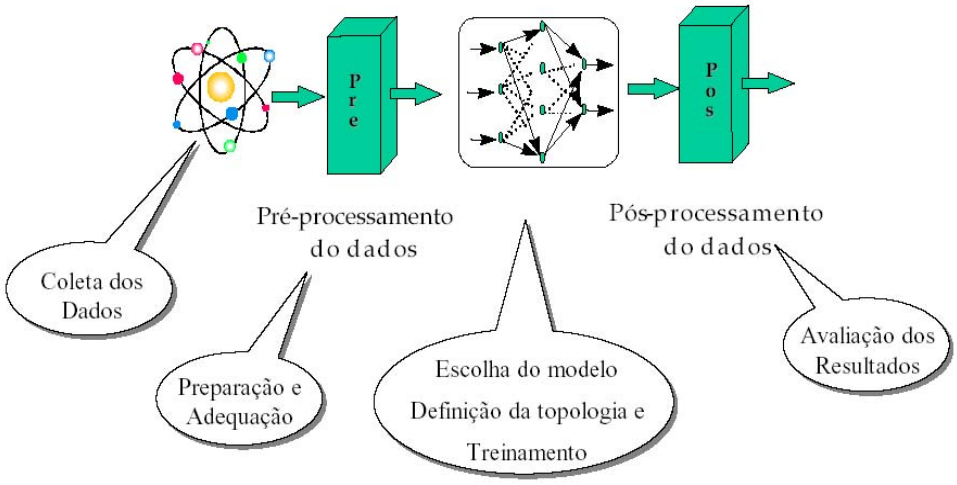

Figura 4.13 Esquema de uma análise por redes neurais.

A etapa de coleta dos dados refere-se à escolha criteriosa das variáveis representativas na avaliação do comportamento da grandeza a modelar, isto é, se tivermos grande quantidade de dados, deveremos escolher aqueles que realmente contribuem na grandeza física que se deseja prever.

O pré-processamento dos dados refere-se ao tratamento e identificação de todas as variáveis selecionadas na etapa anterior, envolvendo atividades como completar dados faltantes, detectar leituras anormais, normalizar séries, etc. A figura 4.14 mostra graficamente um exemplo de geração dos padrões de entrada e saída, com a construção de uma janela temporal de ordem 2, para a previsão de vazão num horizonte (um passo à frente), isto é, a janela considera valores atrasados de dois períodos de tempo (t, t+1) imediatamente anteriores ao valor a ser previsto em t+2. Nesta mesma figura, pode-se também notar que o 'janelamento' pode ter uso estendido para inclusão de outras variáveis de interesse, como a cota do reservatório.

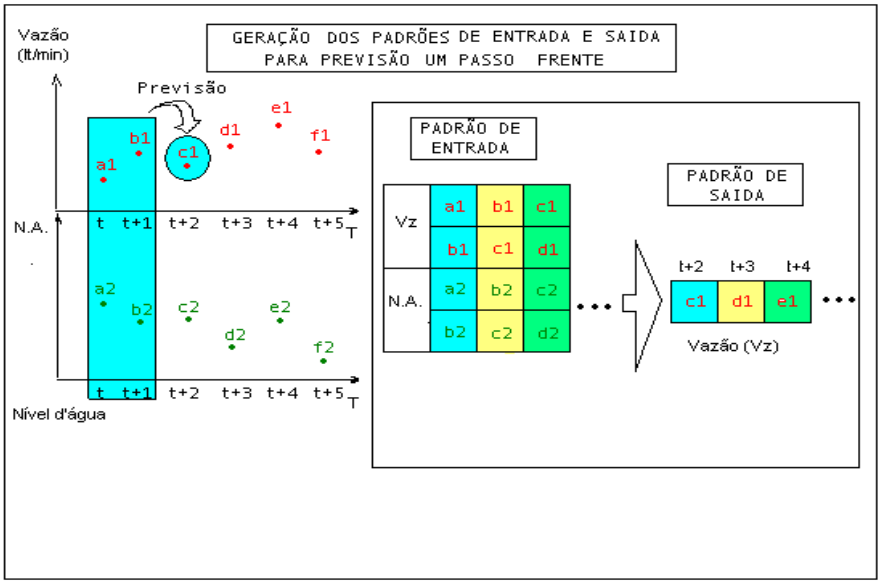

Figura 4.14 Exemplo de geração dos padrões de entrada e saída para previsão um passo à frente.

Uma vez determinados os padrões de entrada, passa-se então à etapa de definição da topologia da rede neural, isto é, a determinação do número de camadas, e de neurônios em cada uma delas. Determinada a "melhor" topologia, o desempenho dos resultados é avaliado através de uma análise de erros.

4.4.3. Treinamento de redes neurais

O objetivo do treinamento de uma rede neural é ajustar os pesos, de tal forma que a aplicação de um conjunto de entradas produza um conjunto de saídas desejadas. Antes de iniciar o processo de treinamento, todos os pesos devem ser inicializados randomicamente com valores pequenos, garantindo desta forma que a rede não fique saturada com grandes valores de pesos, o que a previne da ocorrência de certas patologias do processo de treinamento.

Os dados utilizados para treinamento da rede devem ser significativos e cobrir amplamente o domínio do problema, e não apenas envolver os casos de operações normais ou rotineiras, mas também incorporar casos de exceção e de situações limites.

O treinamento através do algoritmo de retropropagação (*backpropagation*) pode ser dividido nos seguintes passos:

a) Selecionar o próximo par do conjunto de treinamento e aplicar o vetor de entrada da rede (padrões de entrada);
b) Calcular a saída da rede;
c) Determinar o erro entre a saída da rede e a saída-alvo;
d) Ajustar os pesos da rede de maneira a minimizar o erro;
e) Repetir o passo (a) até o passo (d) para cada vetor do conjunto de treinamento, até que o erro se tornar suficientemente baixo para o conjunto total.

Pode-se observar que os passos (a) e (b) constituem a etapa de propagação para frente, onde o sinal de entrada é transmitido através da rede da entrada até a saída, enquanto que os passos (c) e (d), formam a etapa de retropropagação, onde o erro calculado é propagado de volta através da rede com o propósito de ajustar seus pesos.

Apesar do grande sucesso do algoritmo de retropropagação e de sua enorme popularidade, muitos problemas ainda o acompanham. Entre estes, podem ser mencionados:

- O longo período de treinamento, principalmente para problemas complexos, não havendo garantias que depois deste tempo o treinamento tenha sido feito com sucesso;
- Mínimos locais, já que a superfície de erro geralmente apresenta vales e desníveis. Como o algoritmo emprega o método do gradiente descendente, existe sempre a possibilidade de ficar-se estacionado em um mínimo local;
- Paralisia da rede, pois durante o treinamento os pesos podem ser ajustados para valores muito grandes, os quais podem levar a derivada da função de ativação próxima à zero, impedindo assim o aprendizado da rede com base no conjunto de treinamento.

Durante o treinamento, a diminuição do erro é a preocupação central, porém deve-se sempre estar atento à capacidade de generalização da RNA. Para se avaliar este aspecto, o treinamento da rede é validado utilizando-se de exemplos não fornecidos no conjunto de treinamento, sendo os erros calculados neste processo chamado de validação cruzada. A eficiência do treinamento da rede está relacionada à minimização do erro nos exemplos testados (Dyminski, 2000).

O gráfico da figura 4.15 mostra a relação entre o treinamento e a capacidade de generalização da rede, devendo-se cessar o treinamento da mesma quando

erros do teste de validação cruzada atingirem um valor mínimo. Para ter certeza de que o treinamento se deu de maneira satisfatória, ou seja, os erros da rede estejam situados em patamares aceitáveis para o problema proposto, sugere-se ainda utilizar um terceiro conjunto de dados, não pertencentes nem ao conjunto de treinamento nem ao conjunto de validação, chamado de conjunto de teste final de generalização do modelo neural.

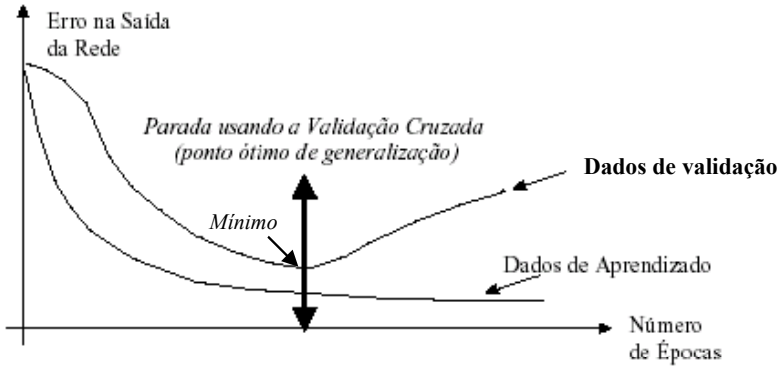

Figura 4.15 Validação cruzada como técnica de parada do treinamento.

4.5. Avaliação do desempenho das previsões

Para avaliação do desempenho das previsões obtidas tanto pelos modelos de Box & Jenkins como pelas redes neurais artificiais foram consideradas as seguintes métricas:

a) Erro percentual médio absoluto (MAPE) - o valor médio do erro percentual das previsões sobre todo o conjunto de teste.

$$MAPE = \frac{1}{N}\sum_{j=1}^{N}\left|\frac{alvo_j - pred_j}{alvo_j}\right| * 100 \qquad (4.10)$$

Onde $alvo_j$ são os valores reais desejados na previsão e $pred_j$ representa os valores previstos pela rede neural.

b) Raiz do erro médio quadrático (RMSE) - esta métrica penaliza muito mais os erros maiores.

$$RMSE = \sqrt{\frac{1}{N}\sum_{j=1}^{N}(alvo_j - pred_j)^2} \tag{4.11}$$

c) Coeficiente U de Theil (1966)

$$U = \sqrt{\frac{\sum_{j=1}^{N}\left(alvo_j - pred_j\right)^2}{\sum_{j=1}^{N}\left(alvo_j - alvo_{j-1}\right)^2}} \tag{4.12}$$

Esta métrica mede quanto os resultados obtidos são melhores do que uma previsão ingênua ou trivial, i.e. quando a melhor estimativa do próximo valor é o próprio valor atual. Através deste coeficiente pode-se analisar a qualidade de uma previsão através dos seguintes valores do coeficiente U de Theil:

- Quando U > 1, o erro do modelo é maior do que o erro na previsão ingênua;
- Quando U < 1, o erro do modelo é menor que o erro na previsão ingênua.

O coeficiente U de Theil menor do que 1 já indica uma previsão melhor que a previsão trivial; quanto mais próximo o mesmo for de zero, melhor será o resultado da previsão.