Capítulo 5

Radial Basis Function

As RNAs do tipo *Radial Basis Function* (RBFs) são redes supervisionadas, consideradas aproximadoras universais, assim como as RNAs *Multilayer Perceptron* (MLPs) treinadas pelo algoritmo *Backpropagation* abordadas no capítulo anterior. As arquiteturas particulares das duas redes são, entretanto, muito diferentes. É interessante comentar algumas diferenças importantes entre as redes MLPs e as redes RBFs:

1. As unidades escondidas em redes MLPs dependem de somas ponderadas das entradas, transformadas por funções de ativação monotônicas [5]. Uma função de ativação comumente aplicada às unidades escondidas de redes MLPs é a função sigmoidal, que é não-linear e continuamente diferenciável. Duas formas da função sigmoidal utilizadas são: a Função Logística e a Função Tangente Hiperbólica.

   Em contraste, em uma RNA RBF, a ativação de uma unidade escondida é determinada por uma função não-linear da distância entre o vetor de entrada e um vetor de referência. As unidades escondidas de uma RNA RBF possuem funções de ativação que são funções localizadas e que apresentam base radial definida sobre seu domínio [5].

2. Uma rede MLP forma uma representação distribuída no espaço de valores de ativação para as unidades escondidas, já que, para um dado vetor de entrada, muitas unidades escondidas contribuirão para a determinação do valor de saída, razão pela qual as redes MLP tendem a resultar em aproximações globais [7][5]. Durante o treinamento, as funções representadas pelas unidades escondidas devem ser tais que, quando linearmente combinadas pela camada final de pesos, gerem as saídas

corretas para um intervalo grande de possíveis valores de entrada. A "interferência" e o "acoplamento cruzado" entre as unidades escondidas levam a resultados durante o processo de treinamento da rede que são altamente não-lineares, resultando em problemas de mínimos locais ou em regiões quase planas na função de erro, fatores estes que podem levar a uma convergência muito lenta do procedimento de treino, mesmo com a utilização de estratégias avançadas de otimização [5].

Em contraste, as RNAs RBF, com funções de base localizadas, formam uma representação no espaço de unidades escondidas que é local com respeito ao espaço de entrada porque, para um dado vetor de entrada, tipicamente apenas algumas unidades escondidas apresentarão ativações significantes. Por esta razão, as redes RBF tendem a produzir aproximações locais [7][5].

3. Uma rede MLP freqüentemente tem muitas camadas de pesos e um complexo padrão de conectividade, de tal forma que nem todos os possíveis pesos em uma dada camada podem estar presentes. Ainda, uma variedade de diferentes funções de ativação podem se utilizadas na mesma rede [5].

   Uma rede RBF, no entanto, geralmente tem uma arquitetura simples, consistindo de duas camadas de pesos, em que a primeira camada contém os parâmetros das funções de base radial e a segunda camada forma combinações lineares das ativações das funções de base radial para gerar a saída [5].

4. Todos os parâmetros em uma rede MLP são usualmente determinados ao mesmo tempo, como parte de uma única estratégia global de treinamento, envolvendo treinamento supervisionado. Este tipo de treinamento apresenta um relativamente alto custo computacional, decorrente da necessidade de retro-propagação do erro, o que faz com que as redes MLPs apresentem uma característica de aprendizado lenta. No entanto, a performance de generalização de uma rede MLP é, em geral, robusta [5].

Já uma rede RBF é tipicamente treinada em dois estágios, com as funções de base radial sendo determinadas primeiramente por meio de técnicas não-supervisionadas, usando para tal apenas os dados de entrada e a segunda camada (de pesos) sendo subseqüentemente determinada por métodos lineares supervisionados, de rápida convergência [5].

A diferente estratégia de treinamento e a conseqüente diferença de velocidade de treinamento entre as duas redes faz com que as redes MLP se mostrem menos adequadas do que as redes RBF, quando estivermos interessados em operações dinâmicas, que envolvam aprendizado continuado (como, por exemplo, em predição de séries temporais e aplicações *on-line*) [6][7].

5. No contexto de aproximação de funções, sob idênticas condições do ambiente no qual estão inseridas, de uma forma geral pode-se afirmar que [5][6][7]:

   - erro final atingido por uma rede RBF é menor do que o erro final atingido por uma rede MLP;
   - a convergência de uma rede RBF pode chegar a ser uma ordem de grandeza mais rápida do que a convergência de uma rede MLP;
   - a capacidade de generalização de uma RNA MLP é, em geral, superior à capacidade de generalização de uma RNA RBF.

O projeto de uma rede neural RBF pode ser visto como um problema de ajuste de curvas (ou, dito de outra forma, um problema de aproximação de funções) em um espaço de alta dimensionalidade. De acordo com este ponto de vista, o aprendizado de uma RBF é equivalente a encontrar uma superfície em um espaço multi-dimensional que melhor se ajuste ao conjunto de dados de treinamento, sendo o critério para o "melhor ajuste" medido em algum sentido estatístico.

As redes neurais artificiais do tipo *Radial Basis Function* são ferramentas extremamente flexíveis em um ambiente dinâmico. Elas têm a capacidade de aprender rapidamente padrões complexos e tendências presentes nos dados e de se adaptar rapidamente às mudanças. Estas características as tornam especialmente adequadas para predição de séries temporais, especialmente aquelas séries regidas por processos não-lineares e/ou não-estacionários.

A primeira seção deste capítulo discorre sobre as redes neurais RBF no contexto da aproximação de funções, enquanto que a segunda seção descreve as redes RBF sob o enfoque da predição não-linear de séries temporais.

5.1 RNAs *Radial Basis Function* no Contexto de Aproximação de Funções

Conforme nos referimos na introdução deste capítulo, as redes neurais artificiais do tipo *Radial Basis Function* (RBF) compõem uma classe de redes neurais artificiais particularmente adequadas à aproximação de funções.

Uma RNA RBF possui em sua arquitetura uma camada escondida definida por um conjunto de funções de base radial, das quais a rede deriva seu nome.

O aprendizado de uma rede RBF é equivalente a ajustar uma superfície não-linear ao conjunto de dados, em um espaço muti-dimensional, de acordo com algum critério estatístico. O processo de generalização equivale a usar esta superfície multi-dimensional para interpolar outros pontos que não pertençam ao conjunto de treino, mas estejam em sua vizinhança.

Os neurônios da camada escondida de uma rede neural RBF são um conjunto de funções que constitui uma base arbitrária no espaço por eles formado, em cujo espaço o conjunto de entrada pode ser expandido. Os dados representados através de redes neurais RBF são, portanto, expandidos com referência a um conjunto finito de funções de ativação neurais, chamadas funções de base radial [7][6]. Cada uma destas funções é centrada em

uma particular coordenada do espaço multi-dimensional dos pontos que compõem o espaço de dados de entrada [6][5]. Cada uma destas coordenadas particulares caracteriza-se por definir o centro de uma – entre várias possíveis – região de maior aglomeração de pontos, ou *cluster* [5], do espaço de dados de entrada.

As redes neurais RBF foram originalmente desenvolvidas para interpolação de dados em espaços multi-dimensionais. Segundo B. Mulgrew [1], o problema da interpolação de dados pode ser assim formulado: dado um conjunto de vetores $\{\underline{u}_i\}$ e um conjunto de escalares $\{y_i\}$, busca-se uma função $F(.)$, tal que,

$$y_i = F(\underline{u}_i), \forall i \tag{5.1}$$

Desde que definida analiticamente, a função $F(.)$ pode ser usada para mapear vetores $\underline{u}$ que não pertençam ao conjunto original, no conjunto de pontos y associados. Uma possível solução para o mapeamento analítico é escolher $F(\underline{u})$, tal que:

$$F(\underline{u}) = \sum_i w_i \, \phi\left(\left\| \underline{u} - \underline{u}_i \right\|^2 \right) \tag{5.2}$$

onde $\phi\left(\left\| \underline{u} - \underline{u}_i \right\|^2 \right)$ é uma função escalar radialmente simétrica, tendo $\underline{u}_i$ como centro. Os vetores $\underline{u}_i$ são, por esta razão, referidos como centros no contexto de redes neurais RBF. O operador $\| \cdot \|$ é usualmente a norma Euclidiana – ou Norma L2 – e mede o módulo do vetor argumento, isto é, a distância Euclidiana da ponta do vetor à sua origem [3][7]. A norma Euclidiana de $\underline{u} \in \Re^M$ é expressa por $\|\underline{u}\| = \sqrt{\sum_{m=0}^{M-1} (u_m)^2} = \sqrt{\underline{u}^T . \underline{u}}$.

Em 1986 Micchelli indicou a existência de um conjunto de funções (tanto limitadas quanto ilimitadas) que são adequadas para interpolação por resultarem em um conjunto de equações lineares para as incógnitas w_i para as quais existe uma única solução [1][6]. A Tabela 5.1 apresenta exemplos destas funções, mais comumente utilizadas como funções de base radial.

O parâmetro σ controla o raio de influência de cada função. Este fator é particularmente evidente no caso da função multi-quadrática inversa e da função Gaussiana, em que ambas as funções são, além de localizadas, monotonicamente decrescentes ($\phi(\zeta) \to 0$ à medida que $\zeta \to \infty$). O parâmetro σ determina o quão rapidamente o valor da função de base radial cai a zero à medida que $\underline{u}$ se afasta do centro $\underline{u}_i$.

Lâmina *spline* fina	$\phi(\zeta) = \frac{\zeta}{\sigma^2} \log\left(\frac{\zeta}{\sigma}\right)$
Multi-Quadrática	$\phi(\zeta) = \sqrt{\zeta^2 + \sigma^2}$
Multi-quadrática inversa	$\phi(\zeta) = \frac{1}{\sqrt{\zeta^2 + \sigma^2}}$
Gaussiana	$\phi(\zeta) = \exp\left(-\frac{\zeta^2}{2\sigma^2}\right)$

Tabela 5.1: Algumas funções de base radial comumente utilizadas.

A função de base radial do tipo Gaussiana é a mais comumente utilizada em aplicações práticas, e será a função adotada nas redes neurais RBF estudadas neste capítulo. Neste tipo de função de base radial, o parâmetro σ corresponde ao desvio-padrão da função Gaussiana. Assim, σ define a distância Euclidiana média (raio médio) que mede o espalhamento dos dados representados pela função de base radial em torno de seu centro.

Os raios de cada uma das funções de base radial de uma mesma rede RBF podem assumir diferentes valores, no entanto, para redes RBF reais, o mesmo raio utilizado para cada neurônio não-linear já permite que a rede uniformemente aproxime qualquer função contínua, desde que haja número suficiente de funções de base radial. Na prática, o valor do raio das funções de base radial afeta as propriedades numéricas dos algoritmos de aprendizado, mas não afeta a capacidade geral de aproximação das redes RBF [1][5].

Originalmente, nas primeiras tentativas de aproximação de funções com redes RBF eram utilizadas tantas funções de base radial quantos fossem os padrões do conjunto de dados representativo da função a ser aproximada, objetivando a exatidão da aproximação. No entanto, esta abordagem não só era computacionalmente custosa, como também gerava o problema de *overfitting* quando o objetivo era não só a aproximação como também a interpolação dos pontos que geravam uma determinada função [1][5].

A solução para estes problemas foi apresentada por Broomhead e Lowe (1988) que sugeriram modificações ao procedimento de interpolação exata (que utiliza tantas funções de base radial quantos forem os padrões presentes nos dados). Uma delas é permitir que nem todos os vetores de entrada (do conjunto de dados) tenham uma função de base radial associada. A outra modificação sugerida exclui a necessidade de que a escolha dos centros das funções de base radial seja restrita ao conjunto original de vetores. Para tanto, Broomhead e Lowe reinterpretaram a rede RBF como um estimador de mínimos quadrados (*Least Squares Estimator*) [1][5][6].

Consideradas estas duas generalizações, o sistema de equações lineares cujas incógnitas são os pesos w_i será sobre-determinado. A solução de tal sistema de equações lineares é obtida através do uso da operação de pseudo-inversão matricial de Moore-Penrose para a matriz de interpolação (matriz $\mathbf{\Phi}$) [6][8] e, em conseqüência, será uma solução de mínimo erro médio quadrático.

Para que possamos compreender o parágrafo anterior, considere que, na rede RBF tratada por Broomhead e Lowe, a determinação do vetor de pesos sinápticos é obtida a partir de uma operação de inversão matricial aplicada a uma matriz de interpolação $\mathbf{\Phi}$. A matriz $\mathbf{\Phi}$ é função das funções de base radial (e, portanto, dos respectivos centros e variâncias escolhidos) e dos vetores de treino pertencentes ao conjunto de dados, aplicados à entrada da rede.

Analiticamente, a matriz de interpolação $\mathbf{\Phi}_{[N\times K]}$ é formada pelos vetores $\underline{\varphi}_i$, $i = 1,2,\cdots,N$, sendo N o número de vetores pertencentes ao conjunto de treino da RNA. Os vetores $\underline{\varphi}_i$ resultam da aplicação dos vetores $\underline{u}_i \in \Re^M$, $i = 1,2,\cdots,N$, à entrada da rede

RBF. Os elementos do vetor $\underline{\varphi}_i \in \Re^K$ são as saídas de cada centro Gaussiano em resposta ao vetor $\underline{u}_i$, $i = 1,2,\cdots,N$, aplicado à entrada da rede (sendo, portanto, funções dos vetores de entrada $\underline{u}_i$ e dos vetores centro das funções de base radial, vetores $\underline{t}_k$, $k = 1,2,\cdots,K$ onde K é o número de funções de base radial adotadas para a RNA RBF). Desta forma, $\mathbf{\Phi}$ pode ser expressa como

$$\mathbf{\Phi} = \begin{bmatrix} \underline{\varphi}_1^{\,T} \\ \underline{\varphi}_2^{\,T} \\ \vdots \\ \underline{\varphi}_N^{\,T} \end{bmatrix} = \begin{bmatrix} 1 & \varphi(\underline{u}_1;\underline{t}_1) & \varphi(\underline{u}_1;\underline{t}_2) & \cdots & \varphi(\underline{u}_1;\underline{t}_K) \\ 1 & \varphi(\underline{u}_2;\underline{t}_1) & \varphi(\underline{u}_2;\underline{t}_2) & \cdots & \varphi(\underline{u}_2;\underline{t}_K) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \varphi(\underline{u}_N;\underline{t}_1) & \varphi(\underline{u}_N;\underline{t}_2) & \cdots & \varphi(\underline{u}_N;\underline{t}_K) \end{bmatrix} \tag{5.3}$$

Broomhead e Lowe propuseram que nem todos os vetores de entrada tivessem uma função de base radial associada, ou seja, N (número de vetores pertencentes ao conjunto de treino da RNA) > K (número de funções de base radial adotadas para a RNA RBF). Com $N > K$ o sistema de equações lineares cujas incógnitas são os pesos w_i será sobre-determinado, ou seja, terá mais equações do que incógnitas. Para a solução de tal sistema utiliza-se o método dos mínimos quadrados, que possui solução através da pseudo-inversão da matriz de interpolação $\mathbf{\Phi}$.

A abordagem de Broomhead e Lowe resultou em uma significativa redução de custo computacional e no aumento da capacidade de generalização das redes RBF, o que possibilitou a sua aplicação a uma vasta gama de problemas em processamento digital de sinais (por exemplo), tais como predição de séries temporais, modelamento de sistemas, rejeição de interferência e equalização/desconvolução de canal.

A Figura 5.1 apresenta a arquitetura da rede neural RBF que é habitualmente usada em tais aplicações. A rede é composta de uma camada de nós fonte (que conectam a rede a seu ambiente externo), à qual é apresentado o vetor de entrada $\underline{u}(n) \in \Re^M$. Uma única camada intermediária de neurônios não-lineares, cada um deles computando uma função distância entre o vetor de entrada e o centro da função de base radial associada, constitui a

camada escondida. Na Figura 5.1 o mapeamento não-linear é expresso por funções de ativação Gaussianas, da forma

$$\varphi_k(n) = \varphi_k\left(\underline{u}(n), \underline{t}_k(n), \sigma_k^2(n)\right) = \exp\left[-\frac{1}{\sigma_k^2(n)}\left\|\underline{u}(n) - \underline{t}_k(n)\right\|^2\right], \tag{5.4}$$

onde $\underline{u}(n) \in \Re^M$ representa o vetor de entrada u no instante n, $\underline{t}_k(n) \in \Re^M$ representa o vetor centro da k–ésima função de base radial $k = 0,1,\cdots,K-1$, K é o número de funções de base radial, e $\sigma_k^2(n) \in \Re$ é a variância associada a cada uma das funções no instante n.

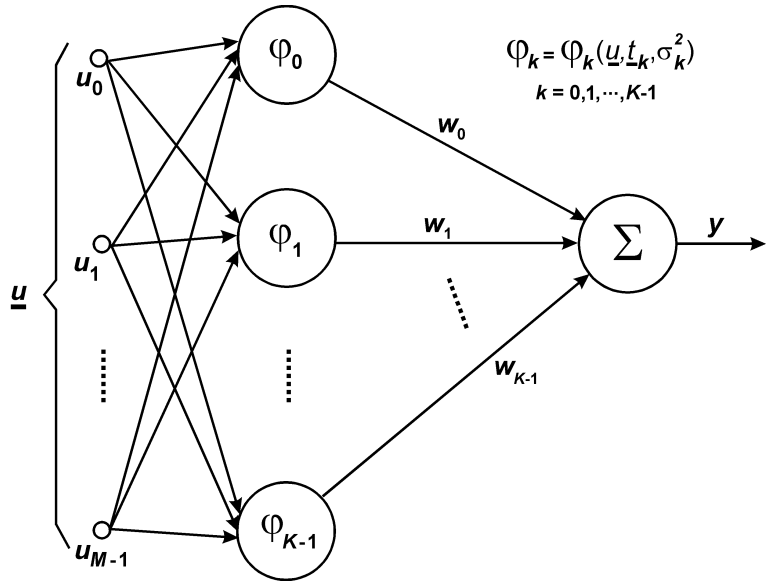

Figura 5.1: Rede neural do tipo *Radial Basis Function*.

A camada de saída da rede neural é formada por um único neurônio linear. O neurônio que compõe a camada de saída é definido como um combinador linear das funções de base radial. A saída y da rede RBF é, portanto, a soma das saídas de cada Gaussiana, ponderadas pelos respectivos pesos sinápticos w_k, de tal forma que a combinação linear é expressa por

$$y = \sum_{k=0}^{K-1} w_k \varphi_k \left(\underline{u}, \underline{t}_k, \sigma_k^2 \right) \tag{5.5}$$

Nesta equação, o termo $\varphi_k \left(\underline{u}, \underline{t}_k, \sigma_k^2 \right)$ é a *k*–ésima função de base radial. Note que φ_k computa o quadrado da distância Euclidiana $D_k{}^2 = \| \underline{u} - \underline{t}_k \|^2$ entre um vetor de entrada $\underline{u}$ e o centro $\underline{t}_k$ da *k*–ésima função de base radial. O sinal de saída produzido pelo *k*–ésimo neurônio escondido é, portanto, devido à função $\exp(\cdot)$ e ao operador $(\cdot)^2$, uma função não-linear da distância D_k . O fator de escala w_k representa o peso do caminho que conecta o *k*–ésimo neurônio escondido ao nó de saída da rede. À Equação (5.5) pode, em alguns casos, ser ainda acrescido um termo constante de polarização ou *bias* [6].

A transformação não-linear acima referida é definida pelo conjunto de funções de base radial φ_k e a transformação linear é definida pelo conjunto de pesos w_k, $k = 0, 1, \cdots, K-1$.

O mapeamento entrada/saída de uma rede RBF Gaussiana é muito semelhante à técnica estatística chamada Mistura de Modelos (*mixture models*), que são misturas de distribuição de probabilidades [4][5]. Em particular, as misturas de distribuição de probabilidades Gaussianas têm sido usadas como modelos em uma grande variedade de aplicações onde os dados de interesse provêm de duas ou mais populações misturadas entre si, com parâmetros estatísticos distintos. A resposta φ_k do neurônio *k* da camada escondida de uma rede RBF representa a densidade probabilística condicional de *u* dado o centro $\underline{t}_k$, isto é, $\varphi_k(u \mid \underline{t}_k)$. O coeficiente w_k representa a probabilidade *a priori* de *u* no contexto da densidade condicional $\varphi_k(u \mid \underline{t}_k)$. Assim, o conjunto das K densidades probabilísticas condicionais modelam a função de densidade de probabilidade representativa do mecanismo estatístico subjacente que gerou os dados, sendo o modelamento definido através de $\sum_{k=0}^{K-1} w_k \varphi_k(u \mid \underline{t}_k)$. Neste sentido, a rede RBF é muitas vezes referida como um Estimador Bayesiano [4][5].

O procedimento para a implementação de uma rede neural RBF compreende a determinação, através de um processo de aprendizagem, dos valores adequados aos parâmetros livres da RBF, que são as variâncias σ_k^2, os centros $\underline{t}_k$ e os pesos sinápticos w_k. O aprendizado ou treinamento consiste em determinar estes parâmetros de tal forma que, dado um conjunto de estímulos $\underline{u}$ na entrada, as saídas *y* se aproximem o mais possível do conjunto de valores desejado.

Diferentes algoritmos podem ser utilizados para a adaptação dos parâmetros livres das redes RBF. Por exemplo, o algoritmo *k-means* [6][7] pode ser utilizado para a inicialização e/ou atualização dos centros das funções de base radial, o algoritmo de Moore-Penrose para pseudo-inversão de matrizes [6] pode ser utilizado para a atualização dos pesos sinápticos, enquanto que o método do Gradiente Estocástico [2][6] pode ser aplicado na atualização dos pesos da rede RBF, das variâncias e dos centros das funções de base radial. A Tabela 5.2 apresenta alguns dos possíveis algoritmos de aprendizado empregados para ajuste dos parâmetros livres.

Possíveis Algoritmos de Aprendizado para Ajuste dos Parâmetros Livres		
Centros das RBF	Pesos Sinápticos	Variâncias dos centros
Constante: Por conhecimento prévio e inferência a partir do conjunto de vetores de treino.	Gradiente Estocástico (LMS). Supervisionado: usa $e(n) = d(n) - y(n)$	Constante: Por conhecimento prévio e inferência a partir do conjunto de vetores de treino.
"Clusterização" pelo algoritmo *k-means*. Não-supervisionado.	Pseudo Inversa por decomposição em valores singulares: $\underline{w}(n) = \mathbf{\Phi}^{-1}(n).\underline{d}(n)$	Gradiente Estocástico (LMS). Supervisionado: usa $e(n) = d(n) - y(n)$
Gradiente Estocástico (LMS). Supervisionado: usa $e(n) = d(n) - y(n)$		$\sigma_k^2(n) = \xi_k(n).\max_{a,b}\left\{\left\|\underline{t}_a(n) - \underline{t}_b(n)\right\|^2\right\}$ onde $\xi_k(n)$ é fixo ou ajustado pelo LMS.

Tabela 5.2: Possíveis Algoritmos de Aprendizado [6].

Diferentes modos de treinamento resultam da combinação dos algoritmos para atualização de centros, variâncias e pesos sinápticos presentes na Tabela 5.2. As redes RBF são, em geral, mais fáceis de treinar do que os *Multilayer Perceptrons* (MLPs), principalmente porque os processos de aprendizagem para os centros, as variâncias e os pesos sinápticos podem ser encadeados seqüencialmente [1], possibilitando que o aprendizado das redes RBF seja otimizado pela resultante divisão de tarefas.

Por exemplo, um algoritmo não-supervisionado pode ser utilizado para estimar os centros, uma posterior estimativa da distância do vetor de entrada com respeito a cada centro pode ser usada para especificar σ e, finalmente, já tendo sido definidos os centros e os raios σ, os pesos sinápticos podem ser calculados através do algoritmo LMS (*Least Mean Squares*)[6]. Após a estimativa inicial de parâmetros da rede, um ajuste mais fino pode ser dado a esta estimativa, utilizando técnicas de gradiente aplicadas a todos os parâmetros, ao invés de apenas aos pesos sinápticos.

Uma abordagem clássica das redes neurais RBF sob a ótica da interpolação de funções é tratada em [6]. Neste modo de treinamento, em que é utilizado o algoritmo Gradiente Estocástico, os centros das funções de base radial e todos os demais parâmetros livres são atualizados através de um processo de aprendizado supervisionado, baseado na minimização da função de custo dada pelo valor esperado do erro quadrático entre a saída fornecida pela rede RBF e a saída desejada para o processo.

Nesta abordagem, os centros das funções de base radial são primeiramente inicializados pelo algoritmo *k-means*. A cada iteração *n*, o algoritmo *k-means* determina as distâncias entre o vetor $\underline{u}(n)$ pertencente ao conjunto de entrada e cada um dos centros $\underline{t}_k(n)$. Ao centro $\tilde{k}$ que corresponder à menor distância – Equação (5.6) – é aplicada a atualização mostrada na Equação (5.7). Na Equação (5.7), η é a razão de atualização do algoritmo *k-means*.

As iterações prosseguem até que $\| \underline{t}_k(n) - \underline{t}_k(n+1) \| \to \varepsilon, \ \forall k$, onde ε é um número muito pequeno.

$$\tilde{k}(\underline{u}) = \arg\min_{k} \| \underline{u}(n) - \underline{t}_k(n) \|; \quad k = 0,1,\cdots, K-1 \tag{5.6}$$

$$\underline{t}(n+1) = \begin{cases} \underline{t}_k(n) + \eta [\underline{u}(n) - \underline{t}_k(n)]; & k = \tilde{k}(\underline{u}) \\ \underline{t}_k(n); & \text{outros casos} \end{cases} \tag{5.7}$$

Após a inicialização dos centros pelo algoritmo *k-means*, e definindo a variância inicial comum a todos os centros pela Equação (5.8), os pesos sinápticos são inicializados com zero. Na Equação (5.8), $d^2_{\max}(\underline{t}_i - \underline{t}_j)$ expressa o quadrado da maior distância Euclidiana entre os centros , isto é, $\max\left\{\left\|\underline{t}_i - \underline{t}_j\right\|^2\right\}$.

$$\sigma^2 = d^2_{\max}(\underline{t}_i - \underline{t}_j); \quad \text{com} \quad i, j = 0,1,\cdots, K-1 \tag{5.8}$$

Para cada vetor $\underline{u}(n)$ do conjunto de treino apresentado à entrada da rede, a saída $y(n)$ da rede RBF é determinada e é avaliada a diferença entre este valor de saída e aquele desejado $d(n)$, conforme

$$e(n) = d(n) - y(n) \tag{5.9}$$

O erro assim obtido é utilizado para a posterior atualização até a convergência dos centros, pesos sinápticos e variância.

As equações de atualização, derivadas do algoritmo Gradiente Estocástico [6], são expressas em (5.10), (5.11) e (5.12). Na Seção 5.1.3 deste capítulo é apresentada a derivação das equações de atualização para os pesos sinápticos, os centros e as variâncias dos centros das funções de base radial.

$$w_k(n+1) = w_k(n) + \mu_w e(n) \varphi_k(n) \tag{5.10}$$

$$\underline{t}_k(n+1) = \underline{t}_k(n) + 2\mu_t e(n) w_k(n) \varphi_k(n) \frac{\underline{u}(n) - \underline{t}_k(n)}{\sigma_k^2(n)} \tag{5.11}$$

$$\sigma_k^2(n+1) = \sigma_k^2(n) + \mu_\sigma e(n) w_k(n) \varphi_k(n) \frac{\left\| \underline{u}(n) - \underline{t}_k(n) \right\|^2}{\left(\sigma_k^2(n)\right)^2} \tag{5.12}$$

Observe que, nas equações (5.10), (5.11) e (5.12) os parâmetros μ_w, μ_t e μ_σ são, respectivamente, as razões de aprendizado da adaptação dos pesos sinápticos, dos centros das funções de base radial e das variâncias dos centros.

A apresentação de *N* vetores de dados $\underline{u}(n)$ à entrada da RBF, $n = 0,1,\cdots,N-1$, constitui uma época de treino. Ao final de cada época, o conjunto de vetores de dados é embaralhado, para evitar que a rede aprenda o padrão seqüencial de apresentação dos vetores de treino. Isto porque estamos interessados na capacidade de generalização da rede com relação ao conjunto de dados em si, e a mesma ordem de apresentação dos vetores a cada época de treino poderia prejudicar tal capacidade de generalização.

O treinamento de uma RBF através das Equações (5.10) a (5.12) é continuado até a sua convergência, situação em que o valor obtido para o erro de aproximação é menor que um valor máximo permitido ε. O erro de aproximação será definido na Seção 5.1.1.

5.1.1 Critério de Avaliação do Erro de Aproximação

Para avaliar a capacidade de aproximação das redes RBF é definido o MSEA, ou seja, o Erro Médio Quadrático de Aproximação dado por

$$\text{MSEA} = \frac{1}{N} \sum_{n=0}^{N-1} \left(d(n) - y(n)\right)^2 \tag{5.13}$$

5.1.2 Sumário da Heurística de Treino de uma RNA *Radial Basis Function* no Contexto de Aproximação de Funções

A heurística de treino de uma rede neural do tipo RBF (no contexto de aproximação de funções) proposta por Haykin em [6] é baseada no algoritmo Gradiente Estocástico. Os pesos sinápticos w_k, os centros das funções de base radiais $\underline{t}_k$ e as variâncias dos centros σ_k^2 são atualizados através de um processo de aprendizado supervisionado baseado na minimização de $J = E\{(d(n) - y(n))^2\}$.

I - Inicialização:
1. Subtrair o vetor média do conjunto de N vetores de treino.
2. Normalizar a *i*-ésima componente de cada vetor de treino pelo desvio padrão do conjunto de *N* valores formado pela *i*-ésima componente de todos os *N* vetores de treino.
3. Normalizar o conjunto de N saídas desejadas para o intervalo [-1,+1].
4. Inicializar os centros das funções de base radial: $\underline{t}_k(n=0) \rightarrow$ *k-means* $\underline{t}(n+1) = \begin{cases} \underline{t}_k(n) + \eta[\underline{u}(n) - \underline{t}_k(n)]; & k = \tilde{k}(\underline{u}) \\ \underline{t}_k(n); & \text{outros casos} \end{cases}$ $\tilde{k}(\underline{u}) = \arg\min_k \| \underline{u}(n) - \underline{t}_k(n) \|; \quad k = 0,1,\cdots,K-1$ $\| \underline{t}_k(n) - \underline{t}_k(n+1) \| \rightarrow \varepsilon, \ \forall k$
5. Inicializar os pesos sinápticos: $\underline{w}_k(n=0) \rightarrow 0; \quad k = 0,1,\cdots,K-1$

6. Inicializar as variâncias dos centros:

$$\sigma_k^2(n=0) \rightarrow d_{\max}^2\left(\underline{t}_i - \underline{t}_j\right), \forall k; \quad \text{com} \quad i, j = 0,1,\cdots,K-1, \quad \text{onde}$$

$$d_{\max}^2\left(\underline{t}_i - \underline{t}_j\right) = \max\left\{\left\|\underline{t}_i - \underline{t}_j\right\|^2\right\}$$

II - Treinamento:

1. Apresentar vetor $\underline{u}(n) \in$ ao conjunto de treino à entrada da RBF.

2. Calcular saída da rede através de

$$y(n) = \sum_{k=0}^{K-1} w_k(n)\varphi_k(n) = \sum_{k=0}^{K-1} w_k(n)\exp\left[-\frac{1}{\sigma_k^2(n)}\left\|\underline{u}(n) - \underline{t}_k(n)\right\|^2\right]$$

3. Determinar o erro instantâneo de treinamento na iteração *n* através de

$$e(n) = d(n) - y(n)$$

4. Atualizar os pesos sinápticos w_k conforme

$$w_k(n+1) = w_k(n) + \mu_w e(n)\varphi_k(n)$$

Atualizar os centros das funções de base radial $\underline{t}_k$ através de

$$\underline{t}_k(n+1) = \underline{t}_k(n) + 2\mu_t e(n) w_k(n)\varphi_k(n)\frac{\underline{u}(n) - \underline{t}_k(n)}{\sigma_k^2(n)}$$

Atualizar as variâncias dos centros das funções de base radial $\sigma_k{}^2$ de acordo com

$$\sigma_k^2(n+1) = \sigma_k^2(n) + \mu_\sigma e(n) w_k(n)\varphi_k(n)\frac{\left\|\underline{u}(n) - \underline{t}_k(n)\right\|^2}{\left(\sigma_k^2(n)\right)^2}$$

5. Incrementar n $(n = n+1)$.

   Se n é tal que todos os N vetores do conjunto de treino foram apresentados à entrada da RBF (constituindo uma época), executar o procedimento II.6, caso contrário executar o procedimento II.1.

6. Embaralhar aleatoriamente a ordem dos vetores $\in$ ao conjunto de treino.

7. Determinar e avaliar o Erro Médio Quadrático de Aproximação dado por

   $$\text{MSEA} = \frac{1}{N}\sum_{n=0}^{N-1}(d(n) - y(n))^2$$

   Se $\text{MSEA} < \varepsilon$ executar II.8, caso contrário, executar II.1.

8. Armazenar os parâmetros da rede (w_k , $\underline{t}_k$ e σ_k^2).

Observação:

Havendo *bias*, a Equação para a saída da rede (passo II.2) torna-se:

$$y(n) = \sum_{k=0}^{K-1} w_k(n)\varphi_k(n) + w_b B$$

onde w_b é a transmitância da sinapse do *bias* e B é o valor do bias.

Neste caso, a equação para atualização da sinapse do *bias* (w_b) é dada por

$$w_b(n+1) = w_b(n) + \mu_w e(n) B$$

5.1.3 Derivação das equações de atualização para w_k , $\underline{t}_k$ e σ_k^2 a partir do Gradiente Estocástico

Nesta Seção serão derivadas as equações de atualização para os pesos sinápticos, os centros e as variâncias dos centros das funções de base radial, respectivamente expressas em (5.10), (5.11) e (5.12) , a partir do algoritmo Gradiente Estocástico,

Observe que, em alguns passos do desenvolvimento das equações ao longo desta seção, tomar-se-á a liberdade de não explicitar o indexador *n* para as variáveis envolvidas (desde que seja inequivocamente definido pelo contexto). Tal medida é tomada por simplificação, para que as equações se tornem mais compactas.

5.1.3.1 Derivação de $\Delta w_k(n)$

Sabemos (do Capítulo 3) que a equação de atualização para os pesos sinápticos a partir do algoritmo Gradiente Estocástico pode ser expressa por

$$w_p(n+1) = w_p(n) - \mu_w \nabla_p \mathrm{J}(n); \quad p = 0,1,\cdots,K-1, \tag{5.14}$$

onde o parâmetro μ_w é a razão de aprendizado (ou passo de adaptação) dos pesos sinápticos e a função de custo J é expressa por

$$\mathrm{J} = \frac{1}{2} e^2 = \frac{1}{2}(d-y)^2, \tag{5.15}$$

em que foi considerada a expressão para o erro dada por (5.9).

Observa-se a partir da Equação (5.14) que, para que possamos determinar $\Delta w_p(n)$ precisamos encontrar $\nabla_p \mathrm{J}$. Desta forma, de (5.15) podemos escrever

$$\nabla_p \mathrm{J} = \frac{\partial \mathrm{J}}{\partial w_p} = \frac{1}{2}\frac{\partial}{\partial w_p}(d-y)^2 \tag{5.16}$$

em que

$$\frac{\partial}{\partial w_p}(d-y)^2 = 2(d-y)\frac{\partial}{\partial w_p}(d-y). \tag{5.17}$$

Na Equação (5.5) encontramos a expressão para a saída y da rede RBF que, substituída em (5.17) conduz a

$$\frac{\partial}{\partial w_p}(d-y)^2 = 2(d-y)\frac{\partial}{\partial w_p}\left(d - \sum_{i=0}^{K-1} w_i \varphi_i\right) \tag{5.18}$$

que pode ser expandida em

$$\frac{\partial}{\partial w_p}(d-y)^2 = 2(d-y)\frac{\partial}{\partial w_p}\left(d - w_0\varphi_0 - w_1\varphi_1 - \ldots - w_p\varphi_p - \ldots - w_{K-1}\varphi_{K-1}\right) \tag{5.19}$$

Como a saída desejada d não depende dos pesos sinápticos, e a derivada $\partial/\partial w_p$ dos termos expandidos só existirá para $w_i = w_p$, teremos

$$\frac{\partial}{\partial w_p}(d-y)^2 = 2(d-y)\left(-\varphi_p\right) \tag{5.20}$$

Substituindo (5.20) em (5.16) encontraremos

$$\nabla_p \mathrm{J} = \frac{\partial \mathrm{J}}{\partial w_p} = \frac{1}{2}\frac{\partial}{\partial w_p}(d-y)^2 = \frac{1}{2}\left[2(d-y)\left(-\varphi_p\right)\right] = (d-y)\left(-\varphi_p\right) \tag{5.21}$$

Desta forma, substituindo (5.21) em (5.14) obteremos a equação de atualização para os pesos sinápticos, que pode ser expressa por

$$w_p(n+1) = w_p(n) - \mu_w (d-y)(-\varphi_p) = w_p(n) + \mu_w\, e(n)\varphi_p(n) \tag{5.22}$$

Observe que a Equação (5.22) é igual à Equação (5.10).

5.1.3.2 Derivação de $\Delta \underline{t}_k(n)$

Passemos agora à derivação da equação de atualização para os vetores centro das funções de base radial, a partir do algoritmo Gradiente Estocástico. Sabemos, do Capítulo 3, que

$$\underline{t}_p(n+1) = \underline{t}_p(n) - \mu_t \underline{\nabla}_p \mathrm{J}(n); \quad p = 0,1,\cdots,K-1 \tag{5.23}$$

onde o parâmetro μ_t é a razão de aprendizado (ou passo de adaptação) dos vetores centro das funções de base radial e a função de custo J é, novamente, expressa pela Equação (5.15).

Observa-se, a partir da Equação (5.23) que, para que possamos determinar $\Delta \underline{t}_p(n)$ precisamos encontrar $\underline{\nabla}_p \mathrm{J}$. Desta forma, de (5.15) podemos escrever

$$\underline{\nabla}_p \mathrm{J} = \frac{\partial \mathrm{J}}{\partial \underline{t}_p} = \frac{1}{2}\frac{\partial}{\partial \underline{t}_p}(d-y)^2 \tag{5.24}$$

onde

$$\frac{\partial}{\partial \underline{t}_p}(d-y)^2 = 2(d-y)\frac{\partial}{\partial \underline{t}_p}\left[d - \sum_{i=0}^{K-1} w_i \varphi_i\right] \tag{5.25}$$

Na Equação (5.4) encontramos a expressão para as funções de base radial φ_i. A partir de (5.4) podemos escrever

$$\frac{\partial}{\partial \underline{t}_p}(d-y)^2 = 2(d-y)\frac{\partial}{\partial \underline{t}_p}\left[d - \sum_{i=0}^{K-1} w_i \exp\left[-\frac{1}{\sigma_i^2}\left\|\underline{u}-\underline{t}_i\right\|^2\right]\right] \tag{5.26}$$

Expandindo o somatório presente em (5.26), encontraremos

$$\frac{\partial}{\partial \underline{t}_p}(d-y)^2 = \tag{5.27}$$

$$= 2(d-y)\frac{\partial}{\partial \underline{t}_p}\left[d - w_0 e^{-\frac{\left\|\underline{u}-\underline{t}_0\right\|^2}{\sigma_0^2}} - w_1 e^{-\frac{\left\|\underline{u}-\underline{t}_1\right\|^2}{\sigma_1^2}} - \ldots - w_p e^{-\frac{\left\|\underline{u}-\underline{t}_p\right\|^2}{\sigma_p^2}} - \ldots - w_{K-1} e^{-\frac{\left\|\underline{u}-\underline{t}_{K-1}\right\|^2}{\sigma_{K-1}^2}}\right]$$

Como a saída desejada d não depende dos centros das funções de base radial e a derivada $\partial/\partial \underline{t}_p$ dos termos expandidos só existirá para $\underline{t}_i = \underline{t}_p$, teremos

$$\frac{\partial}{\partial \underline{t}_p}(d-y)^2 = 2(d-y)\frac{\partial}{\partial \underline{t}_p}\left[-w_p \exp\left(-\frac{\left\|\underline{u}-\underline{t}_p\right\|^2}{\sigma_p^2}\right)\right] = \tag{5.28}$$

$$= 2(d-y)(-2)w_p\left(\frac{\underline{u}-\underline{t}_p}{\sigma_p^2}\right)\exp\left[-\frac{\left\|\underline{u}-\underline{t}_p\right\|^2}{\sigma_p^2}\right]$$

Substituindo (5.28) em (5.24),

$$\underline{\nabla}_p \mathrm{J} = \frac{1}{2}\left\{2(d-y)(-2)w_p\left(\frac{\underline{u}-\underline{t}_p}{\sigma_p^2}\right)\exp\left[-\frac{\left\|\underline{u}-\underline{t}_p\right\|^2}{\sigma_p^2}\right]\right\} = \tag{5.29}$$

$$= -2w_p(d-y)\left(\frac{\underline{u}-\underline{t}_p}{\sigma_p^2}\right)\exp\left[-\frac{\left\|\underline{u}-\underline{t}_p\right\|^2}{\sigma_p^2}\right]$$

Levando o resultado de $\underline{\nabla}_p \mathrm{J}$ obtido em (5.29) à Equação (5.23), teremos

$$\underline{t}_p(n+1) = \underline{t}_p(n) + 2\,\mu_t\, w_p(n) e(n) \left(\frac{\underline{u}(n) - \underline{t}_p(n)}{\sigma_p^2(n)} \right) \varphi_p(n) \tag{5.30}$$

Observe que a Equação (5.30) é igual à Equação (5.11).

5.1.3.3 Derivação de $\Delta\sigma_k^2(n)$

Tendo derivado as equações de atualização para os pesos sinápticos e para os vetores centro das funções de base radial a partir do algoritmo Gradiente Estocástico, resta-nos derivar a equação de atualização para as variâncias dos centros das funções de base radial.

Assim, fazendo $\alpha_p = \sigma_p^2$, partiremos de

$$\alpha_p(n+1) = \alpha_p(n) - \mu_\sigma \nabla_p \mathrm{J}(n); \quad p = 0,1,\cdots,K-1 \tag{5.31}$$

onde μ_σ é o parâmetro razão de aprendizado (ou passo de adaptação) das variâncias dos centros das funções de base radial. A função de custo J é expressa pela Equação (5.15).

A partir de (5.31) observa-se que, para a determinação de $\Delta\alpha_p(n)$, precisamos encontrar $\nabla_p \mathrm{J}$. Desta forma, a partir de (5.15) podemos escrever

$$\nabla_p \mathrm{J} = \frac{\partial \mathrm{J}}{\partial \alpha_p} = \frac{1}{2} \frac{\partial}{\partial \alpha_p} (d - y)^2 \tag{5.32}$$

onde

$$\frac{\partial}{\partial \alpha_p}(d-y)^2 = 2(d-y)\frac{\partial}{\partial \alpha_p}\left[d - \sum_{i=0}^{K-1} w_i \varphi_i\right] \tag{5.33}$$

Na Equação (5.4) encontramos a expressão para as funções de base radial φ_i. A partir de (5.4) podemos escrever

$$\frac{\partial}{\partial \alpha_p}(d-y)^2 = 2(d-y)\frac{\partial}{\partial \alpha_p}\left[d - \sum_{i=0}^{K-1} w_i \exp\left[-\frac{1}{\alpha_i}\left\|\underline{u}-\underline{t}_i\right\|^2\right]\right] \tag{5.34}$$

Expandindo o somatório presente em (5.34), encontraremos

$$\frac{\partial}{\partial \alpha_p}(d-y)^2 = \tag{5.35}$$

$$= 2(d-y)\frac{\partial}{\partial \alpha_p}\left[d - w_0 e^{-\frac{\left\|\underline{u}-\underline{t}_0\right\|^2}{\alpha_0}} - w_1 e^{-\frac{\left\|\underline{u}-\underline{t}_1\right\|^2}{\alpha_1}} - \ldots - w_p e^{-\frac{\left\|\underline{u}-\underline{t}_p\right\|^2}{\alpha_p}} - \ldots - w_{K-1} e^{-\frac{\left\|\underline{u}-\underline{t}_{K-1}\right\|^2}{\alpha_{K-1}}}\right]$$

Como a saída desejada d não depende das variâncias das funções de base radial e a derivada $\partial/\partial\alpha_p$ dos termos expandidos só existirá para $\alpha_i = \alpha_p$, teremos

$$\frac{\partial}{\partial \alpha_p}(d-y)^2 = 2(d-y)\frac{\partial}{\partial \alpha_p}\left[-w_p \exp\left(-\frac{\left\|\underline{u}-\underline{t}_p\right\|^2}{\alpha_p}\right)\right] \tag{5.36}$$

Como $\frac{\partial}{\partial x}\left(\exp\left[-\frac{c}{x}\right]\right) = \frac{c}{x^2}\exp\left[-\frac{c}{x}\right]$, (5.36) pode ser escrita sob a forma

$$\frac{\partial}{\partial \alpha_p}(d-y)^2 = 2(d-y)\left(-w_p\right)\left(\frac{\left\|\underline{u}-\underline{t}_p\right\|^2}{(\alpha_p)^2}\right)\exp\left[-\frac{\left\|\underline{u}-\underline{t}_p\right\|^2}{\alpha_p}\right] \tag{5.37}$$

Levando o resultado de (5.37) à Equação (5.32), teremos

$$\nabla_p \mathrm{J} = \frac{\partial \mathrm{J}}{\partial \alpha_p} = \frac{1}{2}\left\{2(d-y)(-w_p)\left(\frac{\left\|\underline{u}-\underline{t}_p\right\|^2}{(\alpha_p)^2}\right)\exp\left[-\frac{\left\|\underline{u}-\underline{t}_p\right\|^2}{\alpha_p}\right]\right\} = \tag{5.38}$$

$$= (d-y)(-w_p)\left(\frac{\left\|\underline{u}-\underline{t}_p\right\|^2}{(\alpha_p)^2}\right)\exp\left[-\frac{\left\|\underline{u}-\underline{t}_p\right\|^2}{\alpha_p}\right]$$

Substituindo o resultado obtido para $\nabla_p \mathrm{J}$ em (5.38) na Equação (5.31), teremos

$$\alpha_p(n+1) = \alpha_p(n) - \mu_\sigma (d-y)(-w_p)\left(\frac{\left\|\underline{u}-\underline{t}_p\right\|^2}{(\alpha_p)^2}\right)\exp\left[-\frac{\left\|\underline{u}-\underline{t}_p\right\|^2}{\alpha_p}\right] \tag{5.39}$$

Considerando que $e(n) = d(n) - y(n)$ e $\alpha_p = \sigma_p^2$, teremos

$$\sigma_p^2(n+1) = \sigma_p^2(n) + \mu_\sigma\, e(n)\, w_p(n)\varphi_p(n)\left(\frac{\left\|\underline{u}(n)-\underline{t}_p(n)\right\|^2}{(\sigma_p^2)^2}\right) \tag{5.40}$$

Observe que a Equação (5.40) é igual à Equação (5.12).

5.1.4 RNAs *Radial Basis Function* no Contexto de Aproximação de Funções: Exemplo

Este item destina-se à apresentação dos resultados experimentais obtidos no procedimento de treinamento em que os centros são inicializados pelo algoritmo *k-means* e a atualização até a convergência dos centros, variâncias e pesos sinápticos é efetuada

através do Gradiente Estocástico. Este é o algoritmo clássico apresentado em [6] e cuja heurística é descrita na Seção 5.1.2.

Considere a situação hipotética em que se deseja estabelecer a relação analítica entre os oito primeiro algarismos representados em base binária e os valores que definem o quadrado de um décimo de sua representação em base decimal, conforme mostrado na Tabela 5.3.

$\underline{u} \in \Re^3$			$F(\underline{u})$
u_0	u_1	u_2	
0	0	0	0.0
0	0	1	0.01
0	1	0	0.04
0	1	1	0.09
1	0	0	0.16
1	0	1	0.25
1	1	0	0.36
1	1	1	0.49

Tabela 5.3: Mapeamento $F : \Re^3 \rightarrow \Re$ que se deseja aproximar.

Uma possível solução seria tentar expressar $F(\underline{u})$, $\underline{u} = [u_0 \;\; u_1 \;\; u_2]^T$, através do mapeamento linear $F(\underline{u}) = \underline{w}^T \underline{u} = w_0 u_0 + w_1 u_1 + w_2 u_2$, sendo $\underline{w} = [w_0 \;\; w_1 \;\; w_2]^T$ o vetor que define $F(\underline{u})$ obtido da solução do sistema de equações

$$\begin{aligned}
0w_0 + 0w_1 + 0w_2 &= 0.0 \\
0w_0 + 0w_1 + 1w_2 &= 0.01 \\
0w_0 + 1w_1 + 0w_2 &= 0.04 \\
0w_0 + 1w_1 + 1w_2 &= 0.09 \\
1w_0 + 0w_1 + 0w_2 &= 0.16 \\
1w_0 + 0w_1 + 1w_2 &= 0.25 \\
1w_0 + 1w_1 + 0w_2 &= 0.36 \\
1w_0 + 1w_1 + 1w_2 &= 0.49
\end{aligned} \tag{5.41}$$

o qual é evidentemente sobre-determinado – sem solução – pois não existe $\underline{w} = [w_0 \;\; w_1 \;\; w_2]^T$ que atenda simultaneamente todas as Equações (5.41).

No entanto, o mapeamento $F : \Re^3 \rightarrow \Re$ pode ser feito um mapeamento não-linear quando expresso pela rede neural RBF da Figura 5.1. Especificamente, o conjunto de $N = 8$ vetores $\underline{u}(n) \in \Re^3$ da Tabela 5.3, $n = 0,1,\cdots,N-1$, é considerado como o conjunto de treino da rede RBF, com saída desejada $d(n)$ dado pelo respectivo valor na coluna $F(\underline{u})$ da tabela, isto é, $d(n) = F(\underline{u}(n))$. Como $\underline{u}(n) \in \Re^3$, faz-se $M = 3$. Adotou-se $K = 3$ funções de base radial para formar a superfície de aproximação. Os centros das funções de base radial são inicializados pelo algoritmo *k-means*, através da Equação (5.7) com $\eta = 0.1$, e as variâncias são inicializadas pelo valor dado pela Equação (5.8). A atualização dos pesos sinápticos, centros e variâncias, através das Equações (5.10) a (5.12), utiliza as razões de aprendizado $\mu_w = 0.1$, $\mu_t = 0.1$ e $\mu_\sigma = 0.1$.

Para o treinamento da rede RBF, o conjunto de treino é normalizado de forma que os seus valores extremos situem-se no intervalo $[-1,1]$. Esta é uma precaução usual, que visa evitar *overflow* das variáveis de ponto flutuante ao longo da operação do algoritmo Gradiente Estocástico. Cada componente do conjunto de vetores $\underline{u}(n) \in \Re^3$ é normalizado através da transformação $\theta_u : \Re \rightarrow \Re$, $\theta_u(x) = 2x - 1$, e cada um dos valores do conjunto de N saídas desejadas $d(n)$, $n = 0,1,\cdots,N-1$, é normalizado através da transformação $\theta_d : \Re \rightarrow \Re$, $\theta_d(x) = 4.08163x - 1$.

A apresentação dos $N = 8$ vetores do conjunto de treino definido pela Tabela 5.3 constitui uma época. A Figura 5.2 mostra a evolução do NMSEA[1], à medida que as épocas de treinamento se sucedem. Após 2000 épocas de treino, a rede RBF apresenta a relação analítica dada pela Equação (5.42) como aproximação para o mapeamento da Tabela 5.3. Note que a Equação (5.42) inclui o efeito das normalizações θ_u e θ_d.

---

[1]NMSEA (Erro Médio Quadrático Normalizado de Aproximação), dado por $$\text{NMSEA} = \frac{1}{N}\sum_{n=0}^{N-1}\frac{(d(n) - y(n))^2}{(d(n))^2}$$

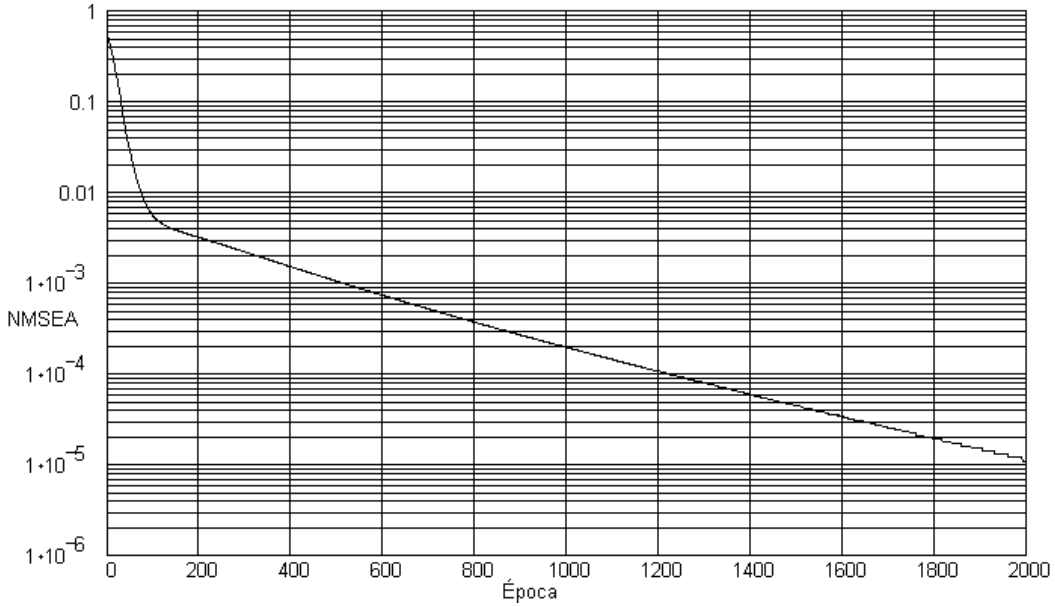

Figura 5.2: Evolução do NMSEA à medida que as épocas de treinamento se sucedem, com conjunto de treino definido pela Tabela 5.3.

$$F(\underline{u}) = \frac{\left\{ \begin{array}{l} -1.948417 \exp\left(\dfrac{-\left\| (2\underline{u} - \underline{1}) - \begin{bmatrix} -0.069259 \\ -0.962758 \\ 0.015931 \end{bmatrix} \right\|^2}{2.774243} \right) + \\ 2.111299 \exp\left(\dfrac{-\left\| (2\underline{u} - \underline{1}) - \begin{bmatrix} 1.265731 \\ 0.180856 \\ 0.316373 \end{bmatrix} \right\|^2}{2.372715} \right) + \\ -1.045209 \exp\left(\dfrac{-\left\| (2\underline{u} - \underline{1}) - \begin{bmatrix} -1.234718 \\ 1.019335 \\ -0.118491 \end{bmatrix} \right\|^2}{1.934487} \right) \end{array} \right\} + 1}{4.08163} \tag{5.42}$$

Verificando a consistência de (5.42) para representar o mapeamento da Tabela 5.3, temos que $F\left([0\,0\,0]^T\right) = -7.739\times10^{-6}$, $F\left([0\,0\,1]^T\right) = 0.01$, $F\left([0\,1\,0]^T\right) = 0.04$, $F\left([0\,1\,1]^T\right) = 0.09$, $F\left([1\,0\,0]^T\right) = 0.16$, $F\left([1\,0\,1]^T\right) = 0.25$, $F\left([1\,1\,0]^T\right) = 0.36$, $F\left([1\,1\,1]^T\right) = 0.49$. Portanto, a Equação (5.42) representa a Tabela 5.3 com boa aproximação.

5.2 Redes Neurais Artificiais RBF no Contexto de Filtragem Preditiva Não-Linear

Se conhecemos o comportamento passado de um sinal até um determinado ponto no tempo é possível fazer alguma inferência sobre seus valores futuros. Tal processo de inferência é conhecido como predição.

A predição de séries temporais é um estudo de extrema relevância, já que exemplos de séries temporais são abundantemente encontrados na natureza (em campos tais como geofísica, astrofísica e meteorologia), nas ciências sociais (em campos como a demografia), nas ciências médicas (em estudos de processos fisiológicos involuntários), nas ciências econômicas (no acompanhamento das taxas de câmbio de moedas e mercado de ações) e nas diversas engenharias (em tratamento e transmissão de sinais, sistemas dinâmicos, etc.), entre muitos outros.

Nesta seção estudaremos as RNAs RBF no contexto de Filtragem Preditiva Não-Linear. Como introdução ao estudo da predição não-linear de séries temporais através de redes neurais RBF, primeiramente apresentaremos uma abordagem de predição de séries temporais através de predição linear.

Em capítulos anteriores vimos que as redes neurais artificiais têm a capacidade de aprender padrões subjacentes presentes nos conjuntos de dados, apresentando melhor desempenho que os métodos estatísticos tradicionais quando o processo regente dos dados é desconhecido, não-linear e/ou não-estacionário – como o são a maior parte dos processos

encontrados no mundo real. Por esta razão representam uma grande contribuição ao estudo das séries temporais resultantes de tais processos [7].

Estudamos também que as redes neurais artificiais supervisionadas constituem uma classe particular de RNAs que têm a capacidade de mimetizar processos estocásticos associados a conjuntos de dados, através do aprendizado. Assim como na forma convencional de um filtro linear adaptativo, as RNAs supervisionadas têm a capacidade de, através da informação de uma resposta desejada tentar aproximar um sinal alvo durante o processo de aprendizado. Esta aproximação é obtida através do ajuste, de forma sistemática, de um conjunto de parâmetros livres, característico de cada rede neural. Na verdade, o conjunto de parâmetros livres provê um mecanismo para armazenar o conteúdo de informação subjacente presente nos dados que são apresentados à rede na fase de treinamento [7][5].

Os dois principais tipos de redes neurais artificiais supervisionadas são as redes MLP (*Multilayer Perceptrons*) treinadas pelo algoritmo *back-propagation* e as redes RBF (*Radial Basis Function*). Como sabemos, ambas as redes são aproximadoras universais. No entanto, quando se trata de aprendizado continuado, como no caso da predição de séries temporais, as redes MLP se mostram menos adequadas porque o custo computacional de treino de uma rede MLP é muito superior ao de uma rede RBF, o que impossibilita a operação de forma dinâmica [6][7].

Já as RNAs RBF possuem características especiais que as capacitam a aprender rapidamente padrões complexos e tendências presentes nos dados e a se adaptar rapidamente a mudanças. Estas características as tornam especialmente adequadas à predição de séries temporais, especialmente aquelas séries regidas por processos não-lineares e/ou não-estacionários, casos em que as técnicas lineares de modelamento têm sucesso apenas limitado em seu desempenho.

5.2.1 Predição Linear de Séries Temporais

A predição linear considera o problema de predição da amostra $u(n+1)$, subseqüente a um conjunto conhecido de amostras consecutivas prévias $\{u(n), u(n-1), \cdots\}$ pertencentes a uma série temporal discreta, problema este conhecido como predição a um passo [3].

Em predição linear, a estimativa da amostra predita, $\hat{u}(n+1)$, é expressa como uma combinação linear de *M* amostras prévias $\{u(n), u(n-1), \cdots, u(n-M+1)\}$. Os coeficientes W_k, $k = 0, 1, \cdots, M-1$ que ponderam tal combinação linear definem um filtro FIR [9] transversal.

A Figura 5.3 detalha um preditor FIR de ordem *M*, o qual é mostrado no instante *n* naquela figura [6]. Portanto, como o instante é definido, visando tornar compactas as equações no desenvolvimento que segue, não será explicitado o indexador *n* para as variáveis envolvidas, a menos que *n* não seja inequivocamente definido pelo contexto.

Um preditor linear de ordem *M* utiliza *M* amostras prévias conhecidas da série temporal para estimar $u(n+1)$, no entanto, necessita do conhecimento de todas as amostras que compõem a série para emular a matriz de correlação associada.

A função de custo J mede o erro médio quadrático entre a estimativa da predição $y(n) = \hat{u}(n+1)$ e o valor efetivamente obtido para a amostra em questão, $u(n+1)$. O vetor $\underline{W}$ que define o filtro FIR tem seus coeficientes determinados de forma a minimizar a função de custo J.

Conforme pode ser observado na Figura 5.3, a amostra predita $\hat{u}(n+1)$ é dada por

$$\hat{u}(n+1) = y(n) = \sum_{k=0}^{M-1} W_k \, u(n-k) = \underline{W}^T \, \underline{u} \tag{5.43}$$

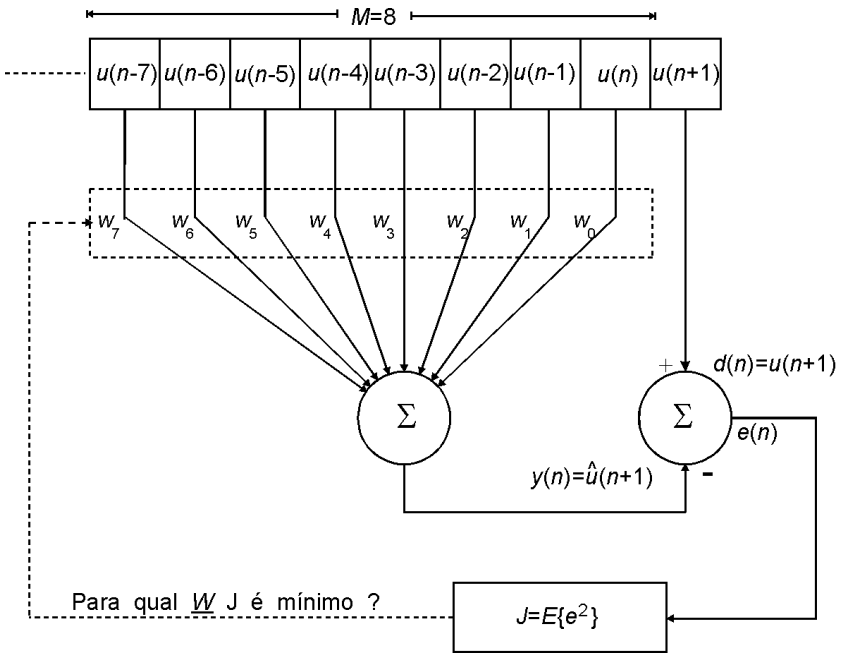

Figura 5.3: Filtro Linear Transversal utilizado como preditor de $u(n+1)$ com ordem de predição $M=8$. $E\{\cdot\}$ é o operador que resulta no valor esperado do argumento [10].

Da Figura 5.3, o erro de predição *e(n)* pode ser expresso por

$$e(n)=d(n)-y(n) \tag{5.44}$$

O operador gradiente é aplicado com o intuito de obter os valores para os pesos W_i do filtro transversal que minimizem a função de custo J, resolvendo-se a equação $\nabla J=0$. Assim, tomando a derivada parcial da função de custo J com relação a cada peso W_i,

$$\nabla_i J=\frac{\partial J}{\partial W_i}=\frac{\partial}{\partial W_i}E\left\{e^2\right\}=E\left\{2e\frac{\partial e}{\partial W_i}\right\}=E\left\{2e\frac{\partial}{\partial W_i}(d-y)\right\}=E\left\{-2e\frac{\partial y}{\partial W_i}\right\}; \tag{5.45}$$

$$i=0,1,\cdots,M-1$$

E, considerando o gradiente a partir do instante *n*, teremos

$$\nabla_i J(n) = E\left\{-2e(n)\frac{\partial y(n)}{\partial W_i}\right\} = E\left\{-2e(n)\frac{\partial}{\partial W_i}\sum_{k=0}^{M-1} W_k\, u(n-k)\right\} = -2E\{e(n)u(n-i)\} \tag{5.46}$$

Como a função de custo J é uma função quadrática, J será globalmente mínimo para $\nabla J = 0$. Assim, a partir da Equação (5.46) podemos escrever que

$$\nabla_i J(n) = -2E\{e(n)u(n-i)\} = 0 \tag{5.47}$$

Substituindo as Equações (5.43) e (5.44) na Equação (5.47), obteremos

$$E\left\{\left[d(n) - \sum_{k=0}^{M-1} W_k\, u(n-k)\right]u(n-i)\right\} = 0 \tag{5.48}$$

Distribuindo os produtos e rearranjando a Equação (5.48),

$$\sum_{k=0}^{M-1} W_k E\{u(n-k)u(n-i)\} = E\{d(n)u(n-i)\} \tag{5.49}$$

Observa-se no lado esquerdo da igualdade expressa na Equação (5.49), que

$$E\{u(n-k)u(n-i)\} = R_{uu}(k-i) \tag{5.50}$$

onde R_{uu} é a função de auto-correlação do processo aleatório *u* (= processo estocástico) para um atraso $k-i$ entre as amostras, com $k,i = 0,1,\cdots,M-1$ [10]. Da mesma forma, observando o lado direito da igualdade expressa na Equação (5.49),

$$E\{d(n)u(n-i)\} = R_{du}(-i) \tag{5.51}$$

onde R_{du} é a função de correlação cruzada entre o processo aleatório que descreve a saída desejada $d = u(n+1)$ e o processo *u*.

Considerando as Equações (5.50) e (5.51), a Equação (5.49) pode ser reescrita como

$$\sum_{k=0}^{M-1} W_k R_{uu}(k-i) = R_{du}(-i); \qquad i = 0,1,\cdots, M\text{-}1 \tag{5.52}$$

Para escrever a Equação (5.52) sob a forma matricial, consideremos que seja $\underline{u}(n)=[u(n)\ u(n-1)\ \cdots\ u(n-M+1)]^T$, tal que

$$\mathbf{R} = E\left\{\underline{u}(n)\ \underline{u}^T(n)\right\} \tag{5.53}$$

isto é

$$\mathbf{R}=\begin{bmatrix} E\{u(n)u(n)\} & E\{u(n)u(n-1)\} & \cdots & E\{u(n)u(n-M+1)\} \\ E\{u(n-1)u(n)\} & E\{u(n-1)u(n-1)\} & \cdots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ E\{u(n-M+1)u(n)\} & E\{u(n-M+1)u(n-1)\} & \cdots & E\{u(n-M+1)u(n-M+1)\} \end{bmatrix} \tag{5.54}$$

ou

$$\mathbf{R}=\begin{bmatrix} R_{uu}(0) & R_{uu}(1) & \cdots & R_{uu}(M-1) \\ R_{uu}(1) & R_{uu}(0) & \cdots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ R_{uu}(M-1) & R_{uu}(M-2) & \cdots & R_{uu}(0) \end{bmatrix} \tag{5.55}$$

Para melhor ilustrar as Equações (5.53), (5.54) e (5.55), consideremos um exemplo. Para o caso em que *M*=3, teremos $\underline{u}(n)=[u(n)\ u(n-1)\ u(n-2)]^T$ e $\mathbf{R}$ será dado por

$$\mathbf{R} = E\left[\underline{u}(n)\underline{u}^T(n)\right]= \tag{5.56}$$

$$= E\left\{\begin{bmatrix} u(n) \\ u(n-1) \\ u(n-2) \end{bmatrix}\begin{bmatrix} u(n) & u(n-1) & u(n-2) \end{bmatrix}\right\} =$$

$$= \begin{bmatrix} R_{uu}(0) & R_{uu}(-1) & R_{uu}(-2) \\ R_{uu}(1) & R_{uu}(0) & R_{uu}(-1) \\ R_{uu}(2) & R_{uu}(1) & R_{uu}(0) \end{bmatrix}$$

Mas, como $R_{uu}(x)=R_{uu}(-x)$, $\mathbf{R}$ poderá, por fim, ser expresso como

$$\mathbf{R} = \begin{bmatrix} R_{uu}(0) & R_{uu}(1) & R_{uu}(2) \\ R_{uu}(1) & R_{uu}(0) & R_{uu}(1) \\ R_{uu}(2) & R_{uu}(1) & R_{uu}(0) \end{bmatrix} \tag{5.57}$$

Seja, agora, o vetor $\underline{P}$ definido por

$$\underline{P} = E\{d(n)\underline{u}(n)\} = \tag{5.58}$$

$$= [E\{d(n)u(n)\}\ E\{d(n)u(n-1)\} \cdots\ E\{d(n)u(n-M+1)\}]^T =$$

$$= [P(0)\ P(-1) \cdots\ P(1-M)]^T$$

e seja também o vetor de pesos dado por

$$\underline{W} = [W_0\ W_1\ \cdots\ W_{M-1}]^T \tag{5.59}$$

Assim, partindo das Equações (5.52), (5.53), (5.55), (5.58) e (5.59), teremos

$$\mathbf{R}\,\underline{W} = \underline{P} \tag{5.60}$$

A Equação (5.60) é denominada Equação de Wiener-Hopf [6]. A solução de (5.60) para $\underline{W}$ define os coeficientes do filtro linear transversal mostrado na Figura 5.3. O filtro prediz com o mínimo erro quadrático médio a amostra $u(n+1)$ de uma série temporal que apresenta correlação entre as *M* prévias amostras. Se a matriz de correlação $\mathbf{R}$ da série temporal é não-singular para *M* definido, então $\underline{W}$ pode ser obtido por

$$\underline{W} = \mathbf{R}^{-1}\underline{P} \tag{5.61}$$

sendo $\underline{P}$ o vetor que define a correlação cruzada entre o vetor $\underline{u}(n)$ de entrada e a saída desejada $d(n) = u(n+1)$.

É importante observar que, para uma dada série temporal com N_t amostras totais, apresentando correlação entre *M* amostras consecutivas prévias ao instante a ser predito, a precisão com que $\mathbf{R}$ e $\underline{P}$ representam as correlações envolvidas será tanto maior quanto maior for N_t com relação a *M*.

Isto ocorre porque, na prática, não se conhece o processo aleatório subjacente que determina a série temporal em questão. Portanto, não são conhecidas as funções correlações que são realmente envolvidas no processo. Assim, o operador $E\{\cdot\}$ nas Equações (5.53) e (5.58) é substituído pela média dos vetores de *M* componentes envolvidos no cômputo de $\mathbf{R}$ e $\underline{P}$, média esta realizada sobre o intervalo de N_t amostras totais conhecidas da série temporal.

Desta maneira, a predição linear só tem sentido quando o processo aleatório subjacente é estacionário [10], pois, em caso contrário, $\mathbf{R}$ e $\underline{P}$ não são univocamente definidas, mesmo para N_t suficientemente grande. Ou seja, se a série temporal resulta de um processo aleatório não-estacionário, $\mathbf{R}$ e $\underline{P}$ variam ao longo da série, invalidando a Equação (5.61) para a obtenção do vetor de pesos $\underline{W}$. A solução algumas vezes adotada é assumir que a série temporal é estacionária em intervalos e adaptar $\mathbf{R}$ e $\underline{P}$ para cada intervalo. No entanto, o número de amostras em cada intervalo nem sempre é suficiente para expressar com fidelidade a operação $E\{\cdot\}$.

Esta é a razão do uso cada vez mais disseminado de técnicas de predição não-linear, as quais, embora apresentem custo computacional maior, contornam a necessidade do conhecimento de um número grande de amostras passadas da série a ser predita, suficientes para que o operador $E\{\cdot\}$ seja aproximado com fidelidade pela média temporal.

Objetivando reduzir a complexidade computacional envolvida no cômputo da Equação (5.61), como $\mathbf{R}$ resulta em uma matriz Töeptliz [11], a sua inversão é, em geral, realizada pelo método de Durbin-Levinson [6], muito embora a pseudo-inversão de Moore-Penrose via Decomposição em Valores Singulares [6][8] seja freqüentemente utilizada para contornar os problemas resultantes de uma matriz $\mathbf{R}$ quase singular.

5.2.1.1 Critério de Avaliação do Erro de Predição

Adotaremos como critério para avaliação da qualidade de predição (tanto linear quanto não-linear, que trataremos na próxima seção) o critério sugerido por Gershenfeld e Weigend em [12]. Este critério de avaliação é considerado referência pela comunidade de pesquisadores da área de predição.

A qualidade da predição será expressa em termos da razão entre as somas de erros quadráticos mostrada em (5.62).

$$\frac{\sum_t \left(observação_t - predição_t\right)^2}{\sum_t \left(observação_t - observação_{t-1}\right)^2} \tag{5.62}$$

Em (5.62) o denominador expressa o erro médio quadrático (MSE) de predição obtido para a chamada predição pela última amostra Tal método de predição considera que a melhor predição possível para a próxima amostra consiste simplesmente em repetir o valor efetivamente observado para a amostra atual. O valor obtido por tal critério para o MSE é tomado como normalizador para o MSE resultante das diferenças entre os valores efetivamente obtidos, após a observação da amostra em questão, e os respectivos valores obtidos pelo preditor que está sendo avaliado. Uma razão inferior a 1.0 corresponde a uma predição melhor do que aquela obtida pela simples repetição do valor efetivamente observado para a amostra anterior àquela a ser predita – limiar que qualifica um preditor que pretenda ser útil.

O erro obtido através do procedimento expresso em (5.62) é chamado Erro Médio Quadrático Normalizado (*Normalized Mean Squared Error*) e é referido na literatura por MSE.

Expressando (5.62) em forma de equação, teremos

$$\mathrm{NMSE}(n)=\frac{\sum_{i=1}^{n}(o(i)-p(i))^2}{\sum_{i=1}^{n}(o(i)-o(i-1))^2}=\frac{\sum_{i=0}^{n}(u(i+1)-\hat{u}(i+1))^2}{\sum_{i=0}^{n}(u(i+1)-u(i))^2} \tag{5.63}$$

onde $o(i)$ e $p(i)$ são respectivamente a observação (o valor efetivamente observado) e a predição no instante i. Para uma dada série temporal U com N_t amostras totais o erro ao final do processo de predição de U é dado por $\mathrm{NMSE}(N_t-1)$, onde N_t-1 é o índice do último elemento da série.

5.2.1.2 Predição Linear de Séries Temporais: Exemplo

Nesta seção apresentaremos os resultados obtidos para a predição linear da série *Chaotic_LASER*, através da heurística de predição descrita em 5.2.1.

Descrição da Série Temporal Utilizada:

A série *Chaotic_LASER* possui $N_t=1000$ amostras totais. Cada um dos 1000 pontos que a constituem corresponde à intensidade de um *Far-Infrared-LASER* em estado caótico. A série foi obtida por Udo Huebner do *Phys.-Techn. Bundesanstalt*, Braunschweig, Germany. As medidas foram feitas a partir de um LASER NH3 não-pulsado [13].

A série, cuja representação gráfica é mostrada na Figura 5.4 é descrita por Weigend e Gershenfeld em [12] e foi obtida de http://www.stern.nyu.edu/~aweigend/TimeSeries/SantaFe.html.

Observação: Em ambos os exemplos de predição de séries temporais que apresentaremos neste estudo (caso linear e caso não-linear que será apresentado na Seção 5.2.2) utilizaremos a série *Chaotic_LASER*.

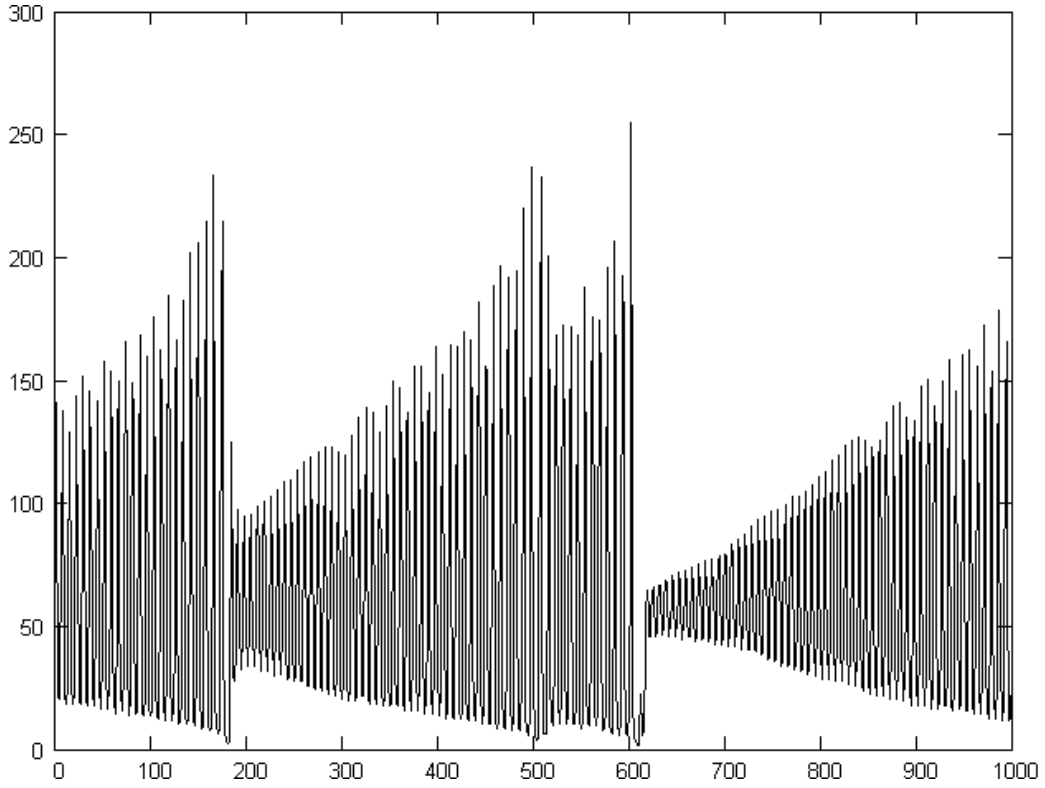

Figura 5.4: Representação gráfica da série *Chaotic_LASER*. Ordenada: Intensidade de um *Far-Infrared-LASER* em estado caótico. Abscissa: Índice da medição.

A predição estimada $\hat{u}(n+1)$ é expressa como uma combinação linear de 11 amostras prévias, equivalendo a dizer que a ordem da predição linear adotada é $M = 11$ (conforme heurística de predição descrita em 5.2.1).

Os valores para os coeficientes que ponderam tal combinação linear foram obtidos através de (5.61) e são $W_0 = \text{-}0.689614$, $W_1 = 0.545837$, $W_2 = \text{-}\,0.206001$, $W_3 = 0.144179$, $W_4 = \text{-}0.109438$, $W_5 = 0.054155$, $W_6 = \text{-}0.402716$, $W_7 = \text{-}0.334968$, $W_8 = 0.221793$, $W_9 = \text{-}0.248085$ e $W_{10} = 0.0365866$, de tal forma que a Equação (5.43) resulta em:

$$\begin{aligned}\hat{u}(n+1) = &-0.689614\,u(n) + 0.545837\,u(n-1) - 0.206001\,u(n-2) + 0.144179\,u(n-3) + \\ &- 0.109438u(n-4) + 0.054155u(n-5) - 0.402716u(n-6) - 0.334968u(n-7) + \\ &+ 0.221793u(n-8) - 0.248085u(n-9) + 0.0365866u(n-10)\end{aligned}$$

Utilizando estes coeficientes no filtro FIR da Figura 5.3, o NMSE final resultante, conforme Equação (5.63), é $\text{NMSE}(N_t - 1) = 0.213$. A Figura 5.5 apresenta as representações gráficas da série *Chaotic_LASER*., observada e predita.

É importante salientar que, apesar de a ordem de predição ser $M = 11$, a predição linear aqui efetuada necessita do conhecimento prévio de todos os $N_t = 1000$ elementos da série temporal para montar a matriz de correlação $\mathbf{R}$, na tentativa de aproximar o operador $E\{\cdot\}$ pela média temporal obtida a partir dos elementos conhecidos.

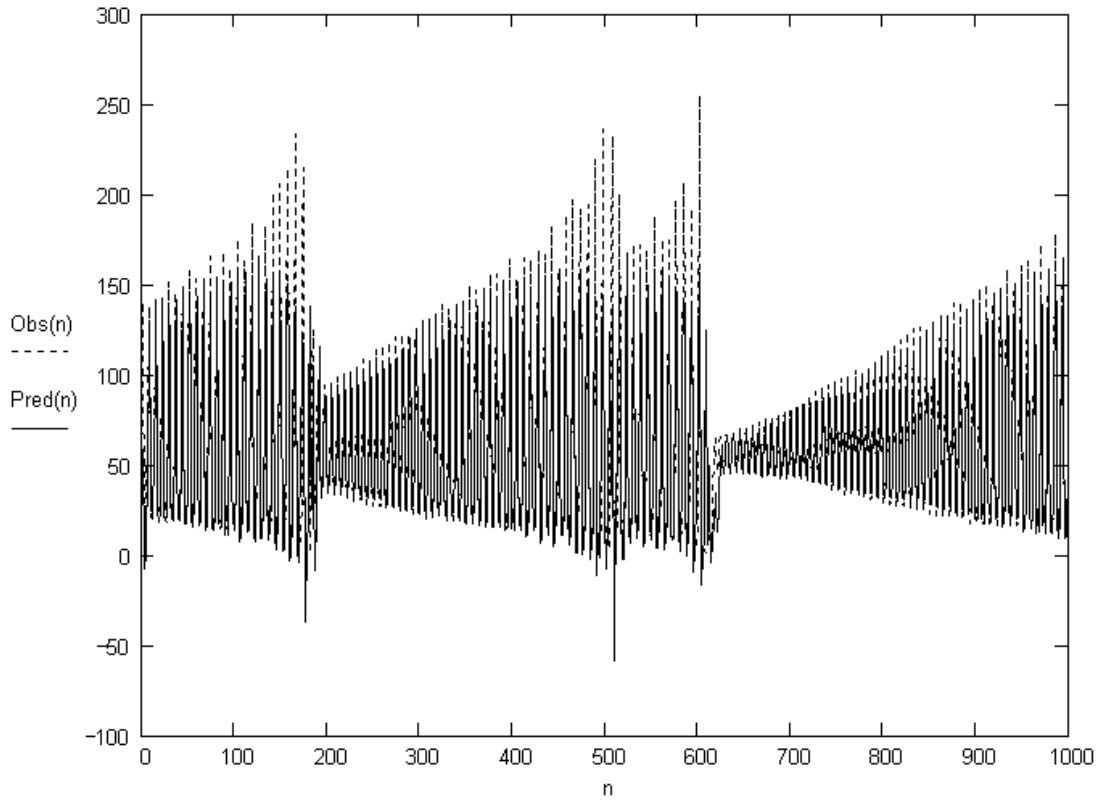

Figura 5.5: Série *Chaotic_LASER* – Predição Linear – $M = 11$. $\text{NMSE}(N_t - 1) = 0.213$

5.2.2 Predição Não-Linear de Séries Temporais através de RNAs RBF

A rede neural artificial RBF utilizada para predição não-linear de séries temporais é dita dinâmica, porque o aprendizado acontece de forma contínua com o desenrolar temporal da série [6]. A Figura 5.6 apresenta a arquitetura da rede RBF em questão.

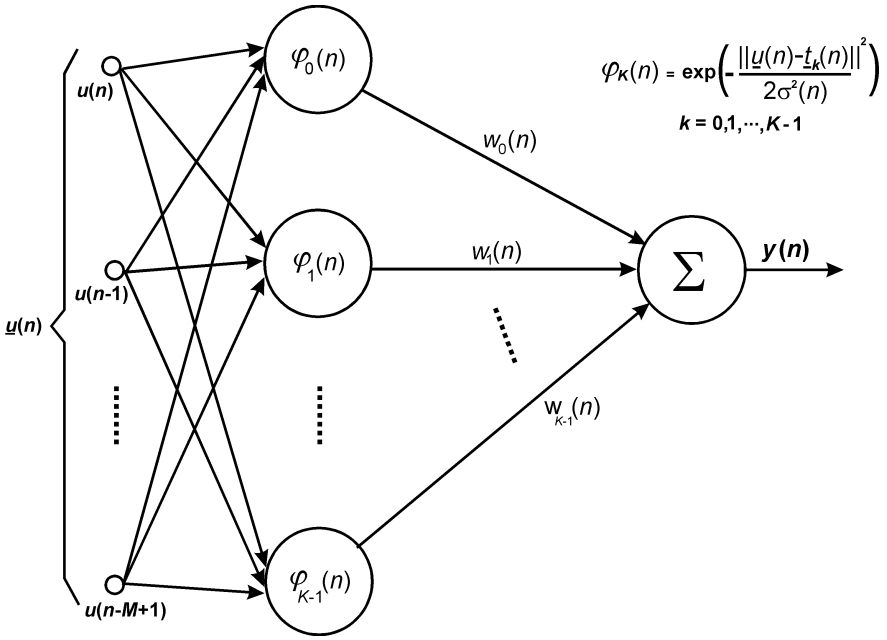

Figura 5.6: RNA RBF utilizada para predição não-linear de séries temporais.

Assim como a rede mostrada na Figura 5.1, esta rede RBF possui M nós de entrada e K centros Gaussianos. O vetor $\underline{t}_k \in \Re^M$ é o k–ésimo vetor centro da rede RBF ou k–ésimo vetor de estado do processo associado à série temporal S, w_k é o k–ésimo peso sináptico e σ^2 é a variância comum a todos os centros Gaussianos, com $k = 0, 1, \cdots, K-1$. No contexto de predição de séries temporais, K é o número de vetores de estado do processo e M é a ordem de predição [6][7].

A saída da rede neural RBF quando o *n*–ésimo vetor de entrada $\underline{u}(n) \in \Re^M$ é apresentado à sua entrada é

$$y(n) = \sum_{k=0}^{K-1} w_k(n)\varphi_k(n) = \underline{\varphi}^T(n)\underline{w}(n) \tag{5.64}$$

onde

$$\varphi_k(n) = \exp\left[-\frac{1}{2\sigma^2(n)}\left\|\underline{u}(n) - \underline{t}_k(n)\right\|^2\right] \tag{5.65}$$

é a saída do *k*–ésimo centro Gaussiano com o vetor $\underline{u}(n)$ aplicado à entrada da rede RBF.

Para utilizar a rede RBF no contexto de predição de séries temporais torna-se necessária a definição de algumas estruturas de dados.

Seja S uma série temporal definida por $S = \{u(0), u(1), \cdots, u(N_t - 1)\}$, onde N_t é o número de amostras de *S*. A um instante *n* qualquer, o objetivo é predizer a amostra $u(n+1)$ de *S*, sendo conhecidas as $N = K + M$ amostras prévias $u(n),\ u(n-1), \cdots,\ u(n-M-K+1)$ que compõem a janela de predição $p(n) = \{u(n-M-K+1), \cdots, u(n-1), u(n)\}$ definida sobre *S*.

No instante *n*, seja o processo associado ao desenrolar temporal de *S* representado pelo conjunto **U** de $K+1$ vetores $\underline{u}(n-\delta) \in \Re^M$, $\delta = 0,1,\cdots,K$, definido sobre a janela $p(n)$, de forma que dois vetores consecutivos de **U** estejam deslocados entre si da distância temporal entre duas amostras subseqüentes de *S*, conforme mostra a Figura 5.7.

Seja, ainda, o vetor de entrada da rede RBF no instante *n* dado por

$$\underline{u}(n) = [u(n)\ \ u(n-1)\ \cdots\ \ u(n-M+1)]^T \tag{5.66}$$

e seja, no instante *n*, o *k*–ésimo vetor de estado $\underline{t}_k(n)$ do processo associado ao desenrolar temporal de *S*, dado por

$$\underline{t}_k(n) = \underline{u}(n-k-1), \quad k = 0,1,\cdots,K-1 \tag{5.67}$$

A Figura 5.7 apresenta, a título ilustrativo, a janela $p(n)$ definida sobre S, a construção dos vetores $\underline{t}_k$ de estado do processo e os vetores de entrada $\underline{u}(n-j)$, $j = 0,1,\cdots,K$, para o caso em que $K = 4$ e $M = 3$.

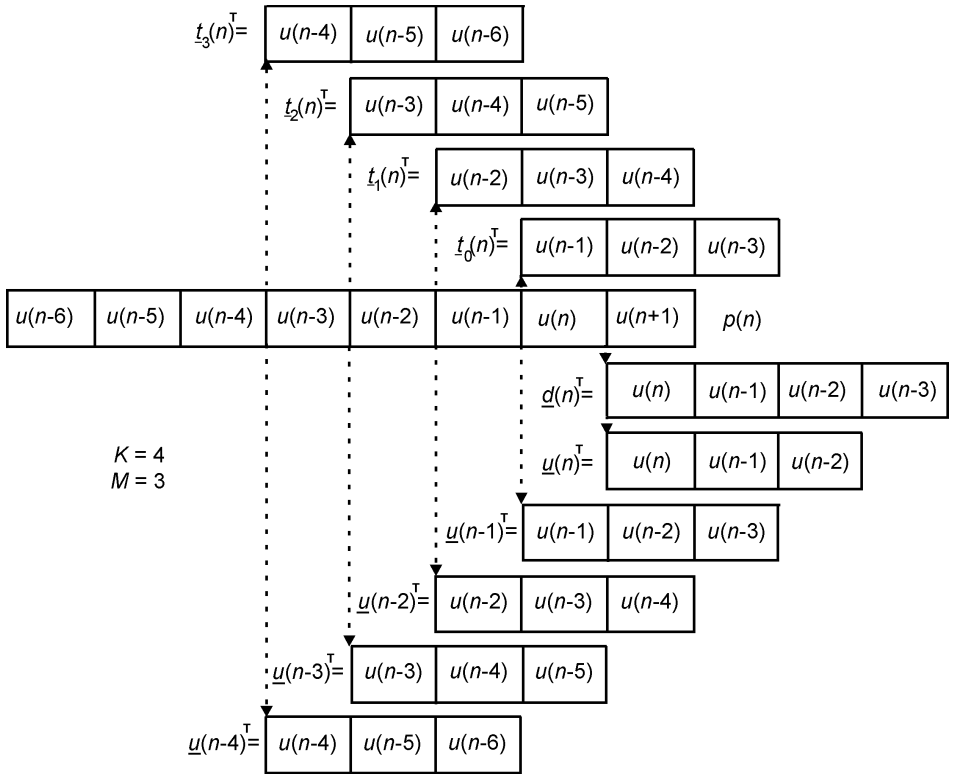

Figura 5.7: Elementos de interesse na série temporal S no instante n e construção dos vetores $\underline{t}_k$ de estado do processo associado à S e dos vetores de entrada $\underline{u}(n-j)$, $j = 0,1,\cdots,K$, $k = 0,1,\cdots,K-1$, para o caso $K = 4$ e $M = 3$. Observe que $p(n)$ é formada por $N = K + M = 7$ amostras conhecidas, prévias a $u(n+1)$.

Para que se defina a variância $\sigma^2(n)$ comum a todos os centros Gaussianos no instante n, assume-se que $\sigma^2(n)$ seja proporcional ao quadrado da máxima distância Euclidiana entre todos os vetores de estado do processo [6]:

$$\sigma^2(n) = \xi \max\left\{\left\|\underline{t}_i(n) - \underline{t}_j(n)\right\|^2\right\}, \quad i, j = 0,1,\cdots,K-1 \tag{5.68}$$

onde ξ é a constante de proporção chamada Fator de Variância, a qual absorve a constante 2 da Equação (5.65).

Assim, a saída do *k*–ésimo centro Gaussiano, quando o vetor $\underline{u}(n)$ é aplicado à entrada da rede RBF pode ser definida como

$$\varphi_k(n) = \exp\left\{-\frac{\left\|\underline{u}(n) - \underline{t}_k(n)\right\|^2}{\xi \max\left\{\left\|\underline{t}_i(n) - \underline{t}_j(n)\right\|^2\right\}}\right\}, \quad i, j = 0,1,\cdots,K-1 \tag{5.69}$$

O conjunto $\varphi_k(n), \quad k = 0,1,\cdots,K-1$, de saídas dos *K* centros Gaussianos, conjunto que resulta da aplicação do vetor de entrada $\underline{u}(n)$, pode ser colocado na forma vetorial através de

$$\underline{\varphi}(n) = \left[\varphi_0(n) \;\; \varphi_1(n) \;\; \cdots \;\; \varphi_{K-1}(n)\right]^T \tag{5.70}$$

Por exemplo, $\underline{\varphi}(n-1) \in \Re^K$ é o vetor que resulta da aplicação do vetor $\underline{u}(n-1) \in \Re^M$ à entrada da rede RBF. Os elementos do vetor $\underline{\varphi}(n-1) \in \Re^K$ são as saídas de cada centro Gaussiano ao vetor $\underline{u}(n-1)$. O *k*–ésimo centro Gaussiano é definido por seu respectivo vetor $\underline{t}_k \in \Re^M$ de estado do processo, $k = 0,1,\cdots,K-1$.

Note que, a qualquer instante arbitrário, a transformação não-linear definida pela Equação (5.69) mapeia o vetor $\underline{u}(n-\delta) \in \Re^M$ aplicado à entrada da rede RBF, δ é um atraso arbitrário qualquer, no vetor $\underline{\varphi}(n-\delta) \in \Re^K$. Portanto, a seqüência de vetores de entrada $\underline{u}(n-1), \underline{u}(n-2), \cdots, \underline{u}(n-K)$ define a seqüência de vetores $\underline{\varphi}(n-1), \underline{\varphi}(n-2), \cdots, \underline{\varphi}(n-K)$ obtidos pelo mapeamento não-linear $\Re^M \rightarrow \Re^K$ definido pela Equação (5.69).

Assim, apesar de definidos em uma dimensão diferente da dimensão original dos vetores $\underline{t}_k \in \Re^M$ de estados da série *S*, e apesar de obtidos através de uma transformação não-linear entre as dimensões $\Re^M$ e $\Re^K$, o conjunto de vetores $\underline{\varphi}(n-1), \underline{\varphi}(n-2), \cdots, \underline{\varphi}(n-K)$ definidos em $\Re^K$ ainda armazena informação sobre o desenrolar temporal da série. Portanto, os vetores em $\Re^K$ contêm informação implícita sobre os estados de *S*, definidos agora em uma outra dimensão, com *S* sendo "vista" através de um processo não-linear. Assim, o conjunto de vetores em $\Re^K$ pode ser agrupado em uma matriz de transição de estados da série temporal *S*, agora interpretada como um processo não-linear com *K* estados $\Re^K$ dimensionais.

A matriz de transição de estados da série temporal *S*, tomada como um processo não-linear com *K* estados $\Re^K$ dimensionais no instante *n*, é mostrada na Equação (5.71).

$$\Phi(n) = \begin{bmatrix} \underline{\varphi}(n-1)^T \\ \underline{\varphi}(n-2)^T \\ \vdots \\ \underline{\varphi}(n-K)^T \end{bmatrix} = \begin{bmatrix} \varphi_0(n-1) & \varphi_1(n-1) & \cdots & \varphi_{K-1}(n-1) \\ \varphi_0(n-2) & \varphi_1(n-2) & \cdots & \varphi_{K-1}(n-2) \\ \vdots & \vdots & & \vdots \\ \varphi_0(n-K) & \varphi_1(n-K) & \cdots & \varphi_{K-1}(n-K) \end{bmatrix} \tag{5.71}$$

Note que as linhas de $\mathbf{\Phi}(n)$ correspondem aos vetores de transição de estado $\underline{\varphi}(n-1)^T, \underline{\varphi}(n-2)^T, \cdots, \underline{\varphi}(n-K)^T$ do processo não-linear, os quais resultam respectivamente da aplicação dos vetores $\underline{u}(n-1), \underline{u}(n-2), \cdots, \underline{u}(n-K)$ à entrada da rede RBF.

Note também, da transformação (5.69), que o *k*-ésimo componente do vetor $\underline{\varphi}(n-\delta) \in \Re^K$, δ arbitrário, tende para o valor máximo 1.0 à medida em que o vetor $\underline{u}(n-\delta) \in \Re^M$ aplicado à entrada da rede RBF tende para o vetor de estado $\underline{t}_k$ da *k*-ésima função de base radial. Portanto, (5.69) é uma transformação não-linear de $\Re^M \rightarrow \Re^K$ que

mede o quanto o desenrolar temporal momentâneo da série S se relaciona com os K estados básicos do processo à ela associado.

Para um vetor de pesos sinápticos arbitrário $\underline{w} = \underline{w}_a$, o conjunto de saídas ou o vetor de saídas $\underline{y}(n)$ para $\boldsymbol{\Phi}(n)$ dado é obtido por

$$\underline{y}(n) = \boldsymbol{\Phi}(n).\underline{w}_a \tag{5.72}$$

com

$$\underline{y}(n) = [y(n-1)\ y(n-2)\ \cdots\ y(n-K)\ ]^T \tag{5.73}$$

onde $y(n-1), y(n-2), \cdots, y(n-K)$ são as saídas da rede RBF com respeito aos vetores de entrada $\underline{u}(n-1), \underline{u}(n-2), \cdots, \underline{u}(n-K)$ associados ao desenrolar temporal momentâneo da série S, sendo dados o vetor de pesos sinápticos arbitrário $\underline{w} = \underline{w}_a$ e a matriz de estados $\boldsymbol{\Phi}(n)$.

Vamos supor que $\underline{y}(n) = \underline{d}(n)$, onde $\underline{d}(n)$ é o vetor de saídas desejadas definido por $\underline{d}(n) = [u(n)\ \ u(n-1)\ \cdots\ \ u(n-K+1)]^T$ conforme construção mostrada na Figura 5.7, de tal forma que

$$\underline{y}(n) = \underline{d}(n) = \begin{bmatrix} u(n) \\ u(n-1) \\ \vdots \\ u(n-K+1) \end{bmatrix} = \Phi(n).\underline{w}(n) = \tag{5.74}$$

$$= \begin{bmatrix} \varphi_0(n-1) & \varphi_1(n-1) & \cdots & \varphi_{K-1}(n-1) \\ \varphi_0(n-2) & \varphi_1(n-2) & \cdots & \varphi_{K-1}(n-2) \\ \vdots & \vdots & & \vdots \\ \varphi_0(n-K) & \varphi_1(n-K) & \cdots & \varphi_{K-1}(n-K) \end{bmatrix} \begin{bmatrix} w_0(n) \\ w_1(n) \\ \vdots \\ w_{K-1}(n) \end{bmatrix}$$

Observe-se que cada elemento em $\underline{d}(n)$ é o elemento que se coloca uma posição à frente na série S, com respeito ao vetor de entrada que gerou o correspondente vetor de transição de estado não-linear em $\boldsymbol{\Phi}$. Por exemplo, a primeira linha de $\boldsymbol{\Phi}$ na Equação

(5.74) é o vetor de transição de estado não-linear $\underline{\varphi}(n-1)^T$ que resulta da aplicação do vetor $\underline{u}(n-1)$ à entrada da rede neural. A saída desejada correspondente a este vetor de entrada é o elemento $u(n)$ de $\underline{d}(n)$, que se encontra uma posição à frente na série *S*, com respeito ao vetor $\underline{u}(n-1)$, conforme pode ser observado na Figura 5.7.

Portanto, através da transformação linear definida pelo vetor de pesos sinápticos $\underline{w}(n) \in \Re^K$ e através da transformação não-linear definida pela matriz $\mathbf{\Phi}(n)$, a Equação (5.74) implicitamente relaciona cada vetor $\underline{u}(n-k-1)$ formado da janela $p(n)$, $k = 0,1,\cdots,K-1$, com o elemento $u(n-k)$ de *S*, sendo $u(n-k)$ o elemento que está localizado na série *S* uma posição à frente do vetor $\underline{u}(n-k-1)$.

Assim, uma vez determinado, $\underline{w}(n)$ conterá informação de como se efetua a transição partindo dos estados prévios do processo de *S* até o próximo elemento de *S* imediatamente adiante ao respectivo vetor $\underline{u}(n-k-1)$ representativo do desenrolar temporal momentâneo de *S*.

É importante ressaltar que a informação de transição em $\underline{w}(n)$ é resultante de uma transformação não-linear $\Re^M \rightarrow \Re^K$, e, em conseqüência disto, é uma informação que envolve as estatísticas de ordem superior do processo de *S*. Em função disto, o método de predição não-linear aqui apresentado é potencialmente mais capaz de captar as "sutilezas estatísticas" do processo estocástico subjacente em *S* do que o método de predição linear baseado em estatísticas de segunda ordem visto na Seção 5.2.1.

A obtenção de $\underline{w}(n)$ é definida pela Equação (5.75).

$$\underline{w}(n) = \mathbf{\Phi}^{-1}(n)\ \underline{d}(n) \tag{5.75}$$

Neste estudo, a inversa da matriz $\mathbf{\Phi}$ é obtida pela pseudo-inversão matricial de Moore-Penrose [6], através de decomposição em valores singulares – SVD. Embora a SVD minimize o problema da eventual singularidade de $\mathbf{\Phi}$ e, embora toda a série temporal seja normalizada para o intervalo $[-1,1]$, antes de qualquer procedimento, por

precaução, adiciona-se o valor 1×10^{-9} à diagonal principal de $\mathbf{\Phi}$, como um parâmetro de regularização [6], visando auxiliar o tratamento das singularidades da matriz $\mathbf{\Phi}$.

Vamos agora efetuar o processo de predição propriamente dito. Uma vez obtido $\underline{w}(n)$ de (5.75), aplica-se o vetor $\underline{u}(n)=[u(n),\ u(n-1),\cdots,\ u(n-M+1)]^T$ à entrada da rede neural. O vetor peso sináptico $\underline{w}(n)$ obtido de (5.75) armazena informação de como ocorre a transição "estados prévios→próximo elemento" dado o vetor de entrada que descreve o desenrolar temporal momentâneo da série *S*. Como, por definição, uma variável de estado não sofre alteração para uma variação pequena no sistema por ela descrito [3], assume-se que os vetores de estado $\underline{t}_k$ do processo de *S* não sofram uma mudança significativa uma posição à frente em *S*. Assim, a saída da rede neural $y(n)$ ao vetor de entrada $\underline{u}(n)=[u(n)\ \ u(n-1)\cdots\ u(n-M+1)]^T$ será uma estimativa $\hat{u}(n+1)$ ou predição da amostra $u(n+1)$, dada pela Equação (5.76) com base em (5.64):

$$\hat{u}(n+1)=y(n)=\sum_{k=0}^{K-1}w_k(n)\exp\left\{\frac{-\left\|\underline{u}(n)-\underline{t}_k(n)\right\|^2}{\xi\max\left\{\left\|\underline{t}_i(n)-\underline{t}_j(n)\right\|^2\right\}}\right\}= \tag{5.76}$$

$$=\sum_{k=0}^{K-1}w_k(n)\varphi_k(n)=\underline{w}^T(n)\underline{\varphi}(n)$$

Em outras palavras, se está implicitamente deslizando a matriz $\mathbf{\Phi}$ uma posição à frente ao longo de *S*, assumindo que os estados armazenados em $\mathbf{\Phi}$ permanecem inalterados e usando-se a informação de transição armazenada em $\underline{w}$ para estimar o próximo elemento em *S*, a partir dos estados definidos em $\mathbf{\Phi}$. Obviamente, *K* deve ser grande o suficiente para que $\mathbf{\Phi}$ possa armazenar todos os estados significativos. Da mesma forma, a dimensão *M* dos vetores de estado do processo de *S* deve ser suficientemente grande para representar os estados significativos do processo.

Esta técnica de predição não-linear através de redes neurais artificiais do tipo RBF com atribuição de centros definida pela Equação (5.67) doravante será referida como predição com base na Atribuição Padrão dos Centros (APC).

5.2.2.1 Sumário da Heurística APC para Predição Não-Linear de Séries Temporais via RNAs RBF

I – Inicialização:

1. Posicionar $p(n=0)$ de N elementos, sobre *S*.

2. Definir K e M, $(N = K + M)$ a partir de $p(n)$.

3. Formar vetores-centro $\underline{t}_k(n) \in \Re^M$ das funções de base radial.

4. Formar vetores $\underline{u}(n-\delta) \in \Re^M, \delta = 0,1,\cdots,K$ do conjunto de treino.

II - Treinamento:

1. Apresentar todos os vetores do conjunto de treino à entrada da rede RBF (exceto o vetor que contém as informações mais recentes sobre *S*, o vetor $\underline{u}(n)$), obtendo para cada um dos demais vetores $\underline{u}(n-\delta) \in \Re^M$, $\delta = 1,\cdots,K$ o correspondente vetor $\underline{\varphi}(n-\delta) \in \Re^K, \delta = 1,\cdots,K$.

2. Construir $\mathbf{\Phi}(n)$ a partir dos vetores $\underline{\varphi}(n-\delta)$.

3. Formar vetor saída desejada $\underline{d}(n) = [u(n)\ u(n-1)\cdots u(n-K+1)]$, atribuindo a cada vetor $\underline{\varphi}(n-\delta)$ uma saída desejada equivalente ao último componente do vetor $\underline{u}(n-\delta+1)$.

4. Determinar vetor de pesos sinápticos $\underline{w}(n) = \mathbf{\Phi}^{-1}(n)\underline{d}(n)$.

5. Aplicar vetor $\underline{u}(n)$ à entrada da RBF, obtendo na saída

$$\hat{u}(n+1) = \sum_{k=0}^{K-1} w_k(n) \exp\left[- \frac{\left\| \underline{u}(n) - \underline{t}_k(n) \right\|^2}{\xi(n) \max_{i,j} \left[\left\| \underline{t}_i(n) - \underline{t}_j(n) \right\|^2 \right]} \right] = \underline{w}^T(n) \underline{\varphi}(n)$$

6. $n = n + 1$ (deslizando $p(n)$ uma posição à frente em *S*).

7. Obtida a nova janela, voltar ao passo 3 da etapa de inicialização.

5.2.2.2 Predição Não-Linear de Séries Temporais: Exemplo

Nesta seção são apresentados os resultados obtidos para a predição não-linear através da heurística APC da mesma série utilizada na Seção 5.2.1.2 (Predição Linear de Séries Temporais: Exemplo), a série *Chaotic_LASER*, com $N_t = 1000$ amostras.

Por coerência, foi adotada a mesma ordem de predição adotada no caso da predição linear, $M = 11$, e $K = 8$ vetores centros, necessitando portanto, de $N = K + M = 19$ amostras conhecidas, anteriores à amostra $u(n+1)$ que se deseja predizer.

Para o Fator de Variância utilizou-se $\xi = 1.0$.

O resultado da predição não-linear para este caso é mostrado na Figura 5.8.

O valor obtido para o Erro Médio Quadrático Normalizado final, conforme Equação (5.63), é $\text{NMSE}(N_t - 1) = 0.096$.

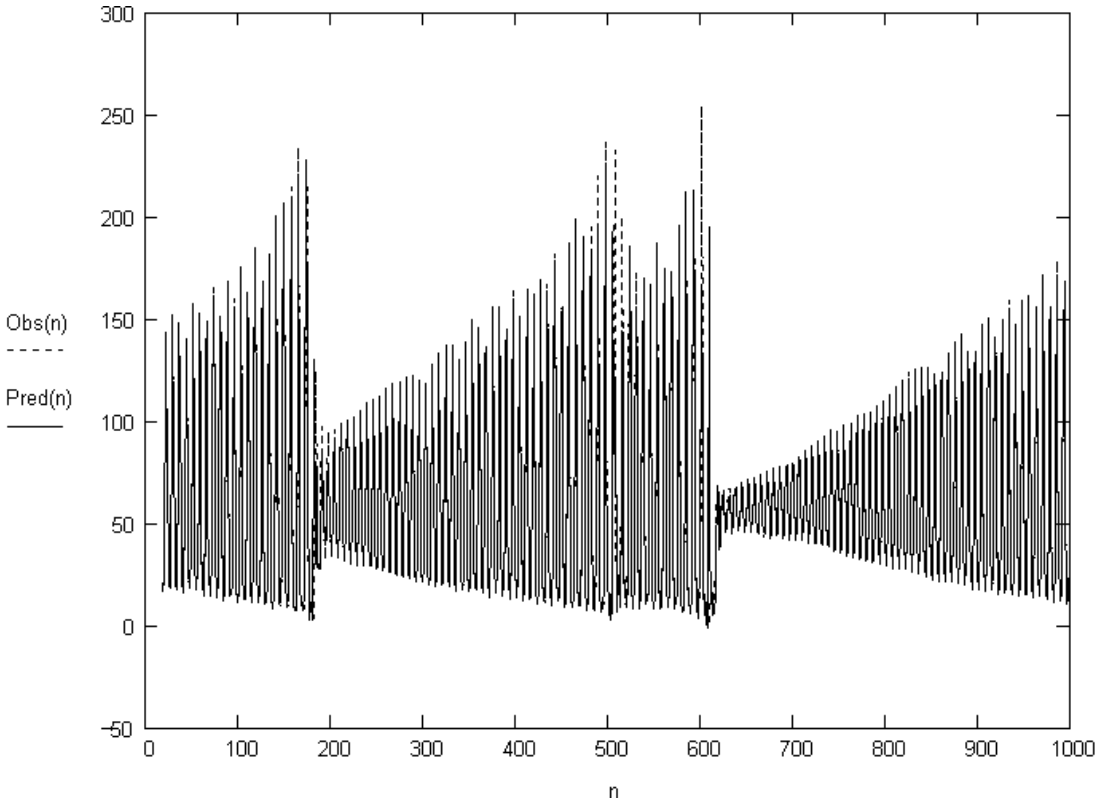

Figura 5.8: Série *Chaotic_LASER* – Predição não-linear através da heurística APC com $M = 11$, $K = 8$ e $N = 19$. NMSE($N_t - 1$) = 0.096.

Experimentou-se novamente a predição linear da série *Chaotic_LASER* (conforme descrita na Seção 5.2.1), considerando, neste novo experimento, o dobro da ordem de predição utilizada na predição por APC, isto é, $M = 22$ para o caso linear.

Os valores dos coeficientes obtidos para o filtro linear, através de (5.43) são $W_0 = -0.670762$, $W_1 = 0.545837$, $W_2 = -0.253092$, $W_3 = 0.198692$, $W_4 = -0.0704754$, $W_5 = 0.129263$, $W_6 = -0.355902$, $W_7 = -0.510382$, $W_8 = 0.277562$, $W_9 = -0.30334$, $W_{10} = 0.217955$, $W_{11} = -0.161362$, $W_{12} = 0.092362$, $W_{13} = -0.189575$, $W_{14} = 0.0408657$, $W_{15} = 0.162011$, $W_{16} = -0.109375$, $W_{17} = 0.0276988$, $W_{18} = -0.0969306$, $W_{19} = 0.0260158$, $W_{20} = -0.0871898$ e $W_{21} = 0.101258$.

O resultado da predição linear para este caso é mostrado na Figura 5.9 e o NMSE final resultante, conforme Equação (5.63), é $\text{NMSE}(N_t - 1) = 0.185$.

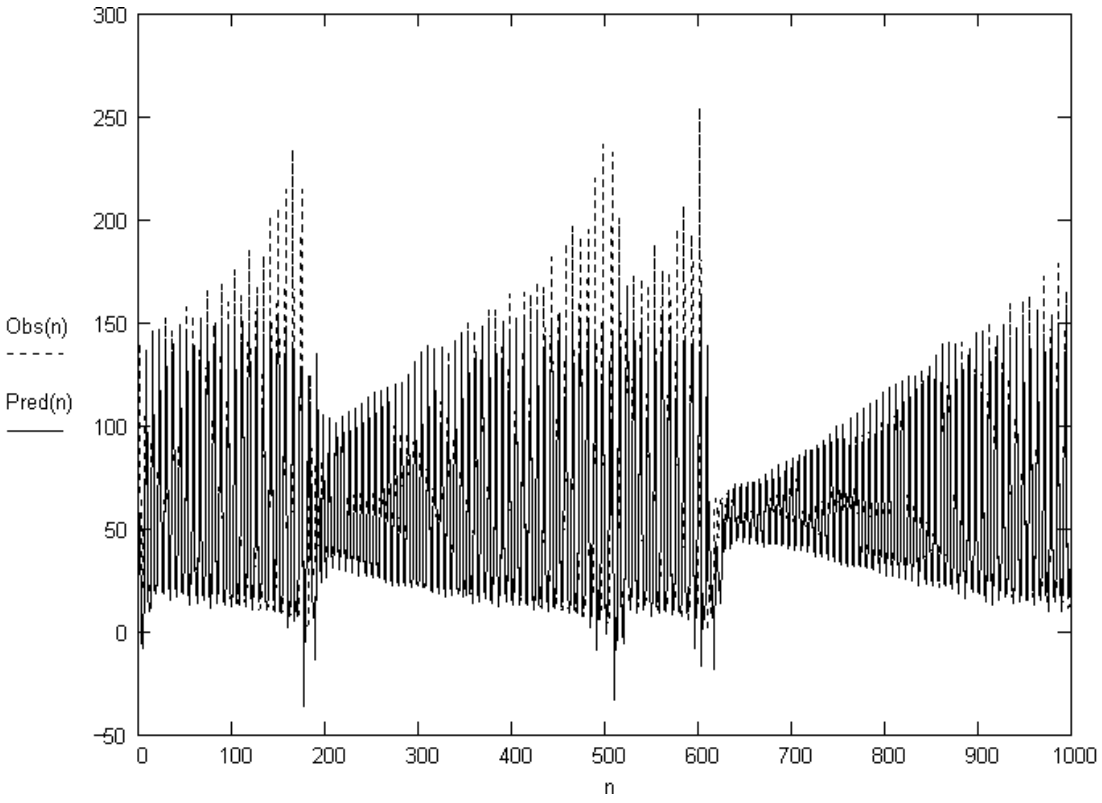

Figura 5.9: Série *Chaotic_LASER* – Predição Linear – $M = 22$. $\text{NMSE}(N_t - 1) = 0.185$.

Observe que, mesmo com o dobro da ordem de predição utilizada na heurística APC, a predição linear resulta em um NMSE final quase duas vezes maior.

A diferença de performance entre a predição linear e a heurística não-linear APC fica ainda mais evidente se lembrarmos que a heurística APC necessita, neste caso, para efetuar a predição, de apenas $N = 19$ amostras prévias conhecidas, contidas na janela de predição $p(n)$. Já a predição linear precisa conhecer todas as $N_t = 1000$ amostras da série *Chaotic_LASER* para montar a matriz de correlação $\mathbf{R}$.

Neste sentido, para uma ordem de predição $M = 11$, pode-se dizer que a heurística APC obtém um NMSE final 0.096 com uma janela de predição com apenas 19 amostras prévias conhecidas, enquanto que a predição linear obtém um NMSE final de 0.213 necessitando, para isso, uma janela de predição equivalente às 1000 amostras prévias conhecidas – todas as amostras da série.

5.3 Referências Bibliográficas do Capítulo 5:

[1] B. Mulgrew, "Applying Radial Basis Functions", *IEEE Signal Processing Magazine*, pp 50-65, 1996.

[2] B. Widrow and S. D. Stearns. Adaptive Signal Processing. Prentice-Hall, Englewood Cliff, 1985.

[3] C. T. Chen, *Linear System Theory and Design*, Harcourt Brace College Publishers, 1984.

[4] C. M. Bishop. "Mixture Density Networks". *Neural Computing Research Group*. Dept. of Computer Science and Applied Mathematics, Aston University, Birmingham, UK, February, 1994.

[5] C. M. Bishop. Neural Networks for Pattern Recognition. Oxford University Press, 1997.

[6] S. Haykin, *Adaptive Filter Theory*, 3rd ed., Prentice Hall, Upper Saddle River, New Jersey, 1996.

[7] S. Haykin, *Neural Networks*, 2nd ed., Prentice Hall, Upper Saddle River, New Jersey, 1999.

[8] W. H. Press, S. A. Teukolsky, W. T. Vetterling and B. P. Flannery. *Numerical Recipes in C*, 2nd ed., Cambridge Uniersity Press, 1992.

[9] R. D. Strum e D. E. Kirk, *First Principles of Discrete Systems and Digital Signal Processing*, Addison-Wesley, 1989.

[10] H. Taub and D.L. Schilling, *Principles of Communications Systems*, McGraw-Hill, 1986.

[11] F. R. Gantmacher, *The Theory of Matrices*, vol.1, Chelsea Publishing Company, New York, NY, 1977.

[12] A. S. Weigend and N. A. Gershenfeld. *Time Series Prediction: Forecasting the Future and Understanding the Past.* Addison-Wesley Publishing Company, 1994.

[13] U. Huebner, N. B. Abraham and C. O. Weiss. "Dimensions and entropies of chaotic intensity pulsations in a single-mode far-infrared NH3 laser." *Phys. Rev.* A 40, p. 6354, 1989.