
PREVISÃO DE VAZÕES MÉDIAS MENSAIS USANDO REDES NEURAIS NEBULOSAS

R. Ballini*

ballini@eco.unicamp.br

S. Soares†

dino@densis.fee.unicamp.br

Marinho G. Andrade‡

marinho@icmc.sc.usp.br

*Departamento de Teoria Econômica, Instituto de Economia, Universidade Estadual de Campinas
CP 6135, 13083-970, Campinas - SP - Brasil.

†Departamento de Engenharia de Sistemas, Fac. de Eng. Elétrica e de Computação, Universidade Estadual de
Campinas, CP 6101, 13083-970, Campinas - SP - Brasil.

‡Instituto de Ciências Matemática e de Computação, Universidade de São Paulo
CP 668, 13560-970, São Carlos - SP - Brasil.

ABSTRACT

This paper presents a neural fuzzy network model for seasonal streamflow forecasting. The model is based on a constructive learning method where neurons groups compete when the network receives a new input, so that it learns the fuzzy rules and membership functions essential for modelling a fuzzy system. The model was applied to the problem of seasonal streamflow forecasting using a database of average monthly inflows of three Brazilian hydroelectric plants located at different river basins. The performance of the model developed was compared with conventional approaches used to forecast streamflows. The results show that the neural fuzzy network model provides a better one-step-ahead streamflow forecasting, with forecasting errors significantly lower than the other approaches.

KEYWORDS: Streamflow forecasting, neural networks, inference fuzzy system, time series.

RESUMO

Este trabalho apresenta um modelo de rede neural nebulosa para previsão de vazões sazonais. O modelo é baseado em um método de aprendizado construtivo onde grupos de neurônios competem quando a rede recebe uma nova entrada. A rede aprende os parâmetros fundamentais para definir as regras nebulosas e funções de pertinência para cada variável de entrada. O modelo foi aplicado para o problema de previsão de vazões médias mensais de três usinas hidroelétricas situadas em diferentes regiões do Brasil. O desempenho do modelo foi comparado com métodos convencionais usados para previsão de vazões. Os resultados mostraram que a rede neural nebulosa forneceu um melhor desempenho para previsão um passo à frente, com erros significativamente menores que as outras abordagens.

PALAVRAS-CHAVE: Previsão de vazões, redes neurais, sistemas de inferência nebulosa, séries temporais.

Artigo submetido em 20/12/2000

1a. Revisão em 25/10/2002

Aceito sob recomendação do Ed. Assoc. Prof. José L. R. Pereira

1 INTRODUÇÃO

Análise e previsão de séries de vazões sazonais são de fundamental importância no planejamento da operação de sistemas de recursos hídricos. Uma das dificuldades na previsão dessas séries é a natureza não estacionária devido aos períodos de cheia e seca do ano. Estas previsões são realizadas atualmente por metodologias baseadas nos modelos de (Box et al., 1994). Como as séries de vazões mensais são séries sazonais com períodos de 12 meses, as quais, geralmente, apresentam correlações periódicas, modelos de séries temporais que vêm sendo sugeridos são os modelos autoregressivos periódicos (MecLeod, 1994), (Vecchia, 1985).

Recentemente, muitas pesquisas vêm sendo desenvolvidas para aplicação de modelos de redes neurais artificiais em previsões de séries temporais por sua habilidade de aprendizado e capacidade de generalização, associação e computação paralela. Estas qualidades as tornam capazes de identificar e assimilar as características das séries, tais como sazonalidade, periodicidade e tendência, na maioria das vezes camufladas por ruídos, sem necessitar da laboriosa formulação teórica, imprescindível para os procedimentos estatísticos (Weigend et al., 1990). Na maioria dos estudos realizados o modelo empregado é a rede multicamadas com algoritmo back-propagation, proposto por (Rumelhart et al., 1986a). Este modelo vem sendo aplicado para previsão de vazões mostrando resultados satisfatórios (Atiya et al., 1999), (Zealand et al., 1999).

A capacidade de mapeamentos complexos das redes neurais cresce com o número de camadas e neurônios. Contudo, este aumento acarreta maior tempo de processamento bem como, uma considerável soma de dados. Na prática, os parâmetros devem ser ajustados rapidamente e somente uma pequena quantidade de dados é disponível (Lachtermacher and Fuller, 1995). Frequentemente, dados do mundo real apresentam ruídos, podendo conter contradições e imperfeições. Assim, modelos de análise de dados baseados em conjuntos nebulosos vêm sendo especialmente utilizados para tratar estas hipóteses (Kacprzyk and Fedrizzi, 1992).

Em anos recentes, surgiu um método promissor unindo os benefícios das redes neurais e da lógica nebulosa, resultando em um sistema integrado, onde a aprendizagem e o poder computacional das redes neurais, e a capacidade de representação e raciocínio da lógica nebulosa são combinados. Estes sistemas são chamados de redes neurais nebulosas (Pedrycz and Gomide, 1998). Esta abordagem vêm sendo usada em modelos de previsão de carga (Bakirtzis et al., 1995) e previsão de vazões

(Ballini et al., 1999), mostrando resultados promissores.

Mais recentemente, um modelo de rede neural nebulosa adaptativa foi proposto por (Figueiredo and Gomide, 1997). Este modelo apresenta um método de aprendizado construtivo no qual grupos de neurônios são adicionados à estrutura da rede quando novo conhecimento é necessário. A rede aprende os parâmetros da função de pertinência para cada variável de entrada dos dados de treinamento, codificando um conjunto de regras nebulosas em sua estrutura, uma propriedade que fornece uma relação dual entre a rede e um sistema nebuloso associado.

Neste trabalho, uma análise comparativa é feita para avaliar o desempenho da rede neural nebulosa, rede multicamadas com algoritmo back-propagation e modelo autoregressivo periódico para previsão de vazões médias mensais de três usinas hidroelétricas brasileiras localizadas em diferentes regiões. Os resultados revelam que o modelo de rede neural nebulosa fornece melhores resultados para previsão 1 passo à frente quando comparados com as outras abordagens.

Na próxima seção é descrito o modelo de rede neural nebulosa. A Seção 3 apresenta os métodos de aprendizagem. A Seção 4 descreve a aplicação da metodologia proposta para previsão de vazões médias mensais. Finalmente, na Seção 5 são apresentadas as conclusões.

1.1 Notação

\mathbf{x}	vetor de entrada
A_j, A_j^i	conjuntos nebulosos
w_i	pesos sinápticos
\hat{y}	saída prevista
P_j^i	medida de possibilidade
H^i	nível de ativação
c_j^i	valor modal
r^i	dispersão
N	número de regras
λ	fator de ampliação
β	taxa de aprendizagem
ϵ^i	fator de aprendizagem
D	distância Euclidiana

γ	fator de redução
y	vazão média mensal
μ_m	média no mês m
σ_m	desvio padrão no mês m
$\phi_{i,m}$	função periódica

2 REDE NEURAL NEBULOSA

Esta seção apresenta a estrutura da rede neural nebulosa (RNN) proposta por (Figueiredo and Gomide, 1997). Esta estrutura foi desenvolvida com base em duas características essenciais: o mapeamento das regras nebulosas na estrutura é direto e o processamento neural e a inferência nebulosa devem concordar totalmente. Essas propriedades são condições necessárias para garantir que o sistema proposto tenha uma natureza dual, ou seja, o sistema pode ser visto ora como um sistema baseado em regras nebulosas, ora como uma rede neural nebulosa (Pedrycz and Gomide, 1998).

A rede é baseada no mecanismo de inferência nebulosa, codificando uma base de regras na forma de “Se um conjunto de condições é satisfeito Então um conjunto de conseqüentes é inferido”, o qual pode ser definido da seguinte forma (Gomide et al., 1995):

Entrada:	$x_1 \text{ é } A_1 \text{ e } \dots x_M \text{ é } A_M$
<hr/>	
Regras:	Se $x_1 \text{ é } A_1^1 \text{ e } \dots x_M \text{ é } A_M^1$ Então $y \text{ é } w_1$

	Se $x_1 \text{ é } A_1^i \text{ e } \dots x_M \text{ é } A_M^i$ Então $y \text{ é } w_i$

	Se $x_1 \text{ é } A_1^N \text{ e } \dots x_M \text{ é } A_M^N$ Então $y \text{ é } w_N$
<hr/>	
Saída:	$\hat{y} = y(\mathbf{x})$

onde x_j , $j = 1, \dots, M$ é uma variável nebulosa e M o número de entradas; A_j e A_j^i , $i = 1, \dots, N$ são conjuntos nebulosos representados por funções de pertinência onde N o número de regras nebulosas; y é um número real definido no espaço de saída. Os valores de $w_i(\mathbf{x}) = w_i$ são números reais e representam os pesos sinápticos correspondentes aos conseqüentes das regras nebulosas, sendo que para cada regra nebulosa está associado um único peso sináptico.

Para efeito de cálculo e representação de conhecimento na RNN, todas as funções de pertinência são discretizadas, ou seja, para cada conjunto nebuloso Z define-se uma função de pertinência $Z(\cdot) : U \rightarrow [0, 1]$, onde

U é o universo de discurso, tal que $Z(x_j) = z_j$, se $x_j \in I_{j_k} = (x_{j_{k_I}}, x_{j_{k_F}}]$, onde I_{j_k} é o k -ésimo intervalo de discretização em relação à entrada x_j , $k = 1, \dots, Q$.

Desta forma, define-se a_{j_k} como sendo o valor da função de pertinência do conjunto nebuloso A_j , para todo $x_j \in I_{j_k}$, ou seja, $A_j(x_j) = a_{j_k}$. O valor de $a_{j_k}^i$ é também definido como sendo o valor da função de pertinência do conjunto nebuloso A_j^i , isto é, $A_j^i(x_j) = a_{j_k}^i$, se $x_j \in I_{j_k}$.

A implementação do mecanismo de inferência nebulosa pode ser obtida de várias formas (Pedrycz and Gomide, 1998). Em (Figueiredo and Gomide, 1997) foi utilizado um dos métodos mais conhecidos, proposto por (Yager and Filev, 1994), onde a saída y é determinada a partir de três estágios, como segue:

1. *Comparação*: Para cada regra i , $i = 1, \dots, N$, e para cada antecedente j , $j = 1, \dots, M$, calcula-se a medida de possibilidade P_j^i entre os conjuntos nebulosos A_j e A_j^i , da seguinte forma:

$$P_j^i(\mathbf{x}) = S_k \{T(A_j(\mathbf{x}), A_j^i(\mathbf{x}))\} \quad (1)$$

onde S e T são operadores, correspondentes a uma s -norma e uma t -norma, respectivamente, e $\mathbf{x} = (x_1, \dots, x_j, \dots, x_M)'$ é o vetor de entrada. A notação $S_k(\cdot, \cdot)$ é a aplicação da s -norma para todo intervalo de discretização $k = 1, \dots, Q$ relativo à variável do vetor de entrada x_j .

2. *Agregação dos Antecedentes*: Para cada regra i , $i = 1, \dots, N$, agregam-se os valores de cada um dos antecedentes através de uma t -norma. O resultado desta agregação é chamado de nível de ativação H^i , definido por:

$$H^i(\mathbf{x}) = T_j \{P_j^i(\mathbf{x})\} \quad (2)$$

onde a notação $T_j(\cdot, \cdot)$ é a aplicação da t -norma para todo $j = 1, \dots, M$ relativo ao vetor de entrada \mathbf{x} .

3. *Agregação das Regras*: A saída \hat{y} do sistema de inferência, ou seja, a etapa de defuzzificação dos dados, é calculada através do método do centro de gravidade, da seguinte forma:

$$\hat{y}(\mathbf{x}) = \frac{\sum_{i=1}^N H^i(\mathbf{x}) w_i}{\sum_{i=1}^N H^i(\mathbf{x})} \quad (3)$$

A RNN é constituída por unidades de processamento, chamadas de neurônios nebulosos. Um modelo geral do neurônio nebuloso é mostrado na Figura 1. Matematicamente, este neurônio pode ser representado da seguinte forma:

$$y(\mathbf{x}) = \varphi(\phi(\Psi(\mathbf{x}))) = \varphi(\phi(\psi_1(x_1), \dots, \psi_M(x_M)))$$

onde $\Psi : \mathfrak{R}^M \rightarrow \mathfrak{R}^M$ é o operador sináptico, $\phi : \mathfrak{R}^M \rightarrow \mathfrak{R}$ é o operador de agregação e $\varphi : \mathfrak{R} \rightarrow \mathfrak{R}$ é a função de ativação.

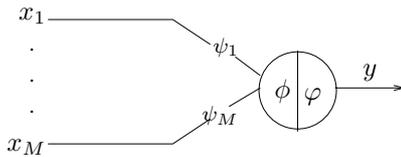


Figura 1: Modelo geral do neurônio.

A RNN apresenta uma arquitetura não recorrente com 5 camadas (Figura 2). Pode-se verificar que o mecanismo de inferência descrito através dos procedimentos de comparação, agregação dos antecedentes e agregação das regras é isomórfico à estrutura da rede.

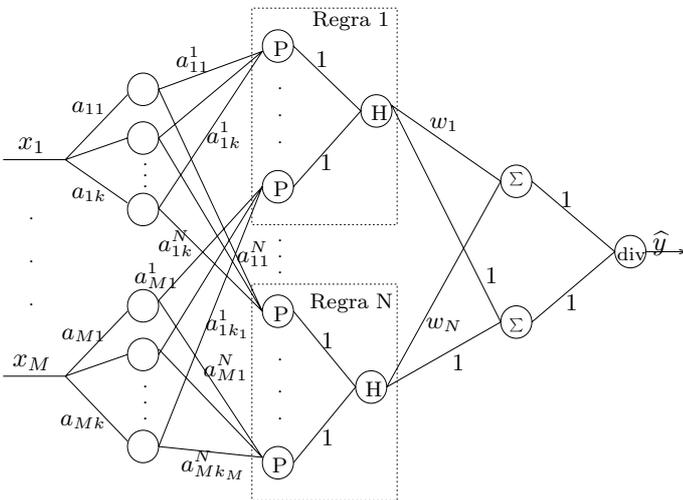


Figura 2: Rede neural nebulosa.

A primeira camada é dividida em M grupos de neurônios, cada um associado a uma variável nebulosa de entrada. Cada neurônio nesta camada representa um intervalo de discretização I_{jk} , correspondente ao espaço de entrada. Ou seja, há tantos neurônios nesta camada quantos forem os intervalos de discretização. Cada grupo de neurônios é responsável por transformar as entradas não nebulosas para que as camadas seguintes possam tratar a informação adequadamente. Estes grupos

geram os valores a_{jk} referentes à função de pertinência do conjunto nebuloso A_j . Assim, um neurônio de entrada recebe um sinal simples, decodifica e o transmite para a segunda camada. O sinal a_{jk} é transmitido pelo k -ésimo neurônio localizado no j -ésimo grupo.

Definindo-se ψ e ϕ como funções identidade, a saída $a_{jk} \in \{0, 1\}$ é dada por $a_{jk} = \varphi(x_j)$. Para um intervalo $I_{jk} = (x_{jk_I}, x_{jk_F}]$, a função decodificada $\varphi(\cdot)$ para este neurônio, através do k -ésimo intervalo, mostrada na Figura 3, é dada por:

$$a_{jk} = \varphi(x_j) = \begin{cases} 1, & \text{se } x_j \in (x_{jk_I}, x_{jk_F}] \\ 0, & \text{caso contrário} \end{cases}$$

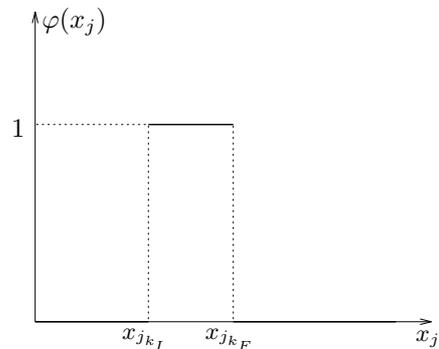


Figura 3: Função de decodificação.

A segunda camada contém N grupos (número de regras), cada qual contendo M neurônios (número de antecedentes de cada regra). Esta camada representa o primeiro estágio de inferência chamado comparação. O j -ésimo neurônio do i -ésimo grupo representa, através dos pesos sinápticos a_{jk}^i , a função de pertinência do conjunto nebuloso A_j^i . Portanto, o i -ésimo grupo é composto por neurônios que representam as funções de pertinência de todos os conjuntos nebulosos correspondentes aos antecedentes da i -ésima regra. Além disso, cada neurônio da i -ésima regra calcula a mediada de possibilidade P_j^i entre os conjuntos nebulosos A_j e A_j^i . Assim, o k -ésimo neurônio do j -ésimo grupo da primeira camada, cuja saída é dada por a_{jk} , se conecta com o i -ésimo grupo da segunda camada através de pesos sinápticos a_{jk}^i . Considerando-se que $\psi = t$ -norma, $\phi = s$ -norma e φ é a função identidade, a saída do j -ésimo neurônio é a medida de possibilidade P_j^i dada pela Equação (1).

A terceira camada contém N neurônios, cada um com M entradas. Para cada grupo i da segunda camada, um neurônio na terceira camada realiza agregação dos antecedentes (segundo estágio de inferência). Todas as conexões possuem peso sináptico unitário. A saída desta

camada é dada pela Equação (2), assumindo $\psi =$ função identidade, $\phi = t$ -norma e $\varphi =$ função identidade.

Cada neurônio i da terceira camada está conectado com os dois neurônios da quarta camada. Ou seja, a quarta camada possui dois neurônios, ambos com N entradas. O operador de agregação associado com estes neurônios é a soma algébrica. Um dos neurônios conecta-se com o i -ésimo neurônio da camada anterior através das sinapses, com pesos w_i . Os operadores ψ e ϕ são, respectivamente, o operador algébrico e a soma algébrica, e φ é a função identidade. As saídas são constituídas pelo numerador da Equação (3).

O outro neurônio também é conectado com todos os neurônios da camada anterior com pesos sinápticos unitários. Considerando $\psi =$ função identidade, $\phi =$ soma algébrica e $\varphi =$ função identidade, a saída equivale ao denominador da inferência nebulosa definida em (3).

Finalmente, a última camada consiste de um único neurônio para calcular o quociente dos sinais de entrada, ou seja, para calcular $\hat{y}(\mathbf{x})$ (Equação (3)). Portanto, as conexões possuem pesos unitários e os operadores ψ , ϕ e φ são função identidade, operador de divisão e a função identidade, respectivamente.

Cada uma das camadas da RNN possui uma associação com os diversos estágios do mecanismo de inferência nebulosa, descritos de (1) a (3). Além disso, os pesos sinápticos representam tanto o conhecimento da rede neural como também representam as funções de pertinência dos antecedentes e conseqüentes das regras nebulosas. Assim, a RNN codifica um conjunto de N regras em sua estrutura, processa as informações através de neurocomputação, executando raciocínio nebuloso.

3 MÉTODOS DE APRENDIZAGEM

A estratégia de aprendizagem do modelo RNN se divide em duas fases. A primeira fase corresponde à aprendizagem não supervisionada, ou seja, a aprendizagem se processa sem que as saídas sejam fornecidas, envolvendo somente uma parte da arquitetura, mais especificamente a segunda camada. Nesta fase, a rede aprende as funções de pertinência dos antecedentes das regras nebulosas enquanto ajusta os pesos sinápticos associados a esta camada. O método utilizado durante esta etapa é do tipo competitivo, baseado nas redes do tipo Kohonen. A segunda fase adota um método supervisionado, baseado no método do gradiente para aprendizagem dos conseqüentes das regras, e envolve apenas o ajuste dos pesos da quarta camada. Assim, a RNN deve aprender os pesos sinápticos a_{jk}^i relacionados aos neurônios

da segunda camada, bem como os pesos sinápticos w_i da quarta camada.

O algoritmo de aprendizagem consiste na apresentação dos pares de entrada/saída desejada à rede, sendo o t -ésimo par dado por $(\mathbf{x}(t), y(t))$, onde $\mathbf{x}(t) = (x_1(t), \dots, x_M(t))'$ é o vetor de entrada com M componentes e y é a saída desejada. Assume-se que os conjuntos nebulosos, $A_j^i(\cdot)$, são representados por funções de pertinência do mesmo tipo, simétricas e completamente definidas pelos valores modais c_j^i e de dispersão r^i . Estes parâmetros devem ser aprendidos durante o treinamento, segundo o método de aprendizagem competitivo, para determinar os pesos sinápticos $a_{jk}^i = A_j^i(x_j) = F_j^i(x_j)$, $x_j \in I_{jk} = (x_{jk_I}, x_{jk_F}]$ da segunda camada. Os pesos sinápticos w_i , pertencentes a quarta camada, são ajustados através do método do gradiente. Dois algoritmos de aprendizagem baseados no paradigma de aprendizado competitivo, chamados de *offline* e *adaptativo* são descritos a seguir.

3.1 Aprendizado Offline

No método de aprendizagem *offline* o número de regras nebulosas N da segunda camada é definido no início do treinamento. Os pares de entrada/saída desejada são apresentados aleatoriamente à rede e o nível de ativação $H^i(\mathbf{x}(t))$ para cada regra i é calculado, determinando-se, assim, a regra vencedora. Esta regra vencedora guia o procedimento de adaptação dos parâmetros da seguinte forma:

a) O vetor de entrada $\mathbf{x}(t)$ é caracterizado por pelo menos uma regra, ou seja, $\exists i \mid H^i(\mathbf{x}(t)) \neq 0$. O grupo $\xi(t)$ vence a competição se $H^{\xi(t)}(\mathbf{x}(t)) \geq H^i(\mathbf{x}(t))$, $\forall i \in \{1, \dots, N\}$ e as equações abaixo são usadas para ajustar os parâmetros do modelo:

$$c_j^{\xi(t)}(t+1) = c_j^{\xi(t)}(t) + \epsilon^{\xi(t)}(t) \left(x_j(t) - c_j^{\xi(t)}(t) \right),$$

$$j = 1, \dots, M \quad (4)$$

$$\eta^{\xi(t)}(t+1) = \eta^{\xi(t)}(t) + 1 \quad (5)$$

$$w_i(t+1) = w_i(t) + \beta \left(\frac{H^i(\mathbf{x}(t)) (y(t) - y(t))}{\sum_{h=1}^N H^h(\mathbf{x}(t))} \right),$$

$$i = 1, \dots, N \quad (6)$$

$$r^i(t+1) = r^i(t), \quad i = 1, \dots, N \quad (7)$$

b) A entrada $\mathbf{x}(t)$ não pertence a nenhuma regra, ou seja, $H^i(\mathbf{x}(t)) = 0, \forall i \in \{1, \dots, N\}$. Neste caso, determinam-se as distâncias $D(\mathbf{c}^i(t), \mathbf{x}(t))$ entre os vetores de entrada $\mathbf{x}(t)$ e modal $\mathbf{c}^i(t) = (c_1^i(t), \dots, c_M^i(t))'$, para $i \in \{1, \dots, N\}$. O grupo vencedor $\xi(t)$ é tal que $D(\mathbf{c}^\xi(t), \mathbf{x}(t)) \leq D(\mathbf{c}^i(t), \mathbf{x}(t)), \forall i \in \{1, \dots, N\}$. As equações de ajuste dos parâmetros da rede são dadas por:

$$c_j^{\xi(t)}(t+1) = c_j^{\xi(t)}(t) + \epsilon^{\xi(t)}(t) (x_j(t) - c_j^{\xi(t)}(t)),$$

$$j = 1, \dots, M \quad (8)$$

$$w_i(t+1) = w_i(t) + \beta \left(\frac{H^i(\mathbf{x}(t))(y(t) - y(t))}{\sum_{h=1}^N H^h(\mathbf{x}(t))} \right),$$

$$i = 1, \dots, N \quad (9)$$

Faça

$$r^{\xi(t)}(t) = \lambda r^{\xi(t)}(t) \quad (10)$$

Enquanto

$$H^{\xi(t)}(\mathbf{x}(t)) = 0 \quad (11)$$

$$\eta^{\xi(t)}(t+1) = \eta^{\xi(t)}(t) + 1 \quad (12)$$

onde, $\eta^i(t)$ é o número de vezes que o valor modal c_j^i é ajustado, $\epsilon^i(t) = \alpha/(\eta^i(t) + 1)$ é o fator de aprendizagem, definido como sendo uma função monotônica decrescente positiva, tal que $0 < \epsilon^i(t) < 1$ e $\alpha \in (0, 1]$. O fator de aprendizagem β é constante e pertence ao intervalo $(0, 1)$ e $\lambda > 1$ é um fator de ampliação.

Esta estratégia de aprendizagem apresenta duas fases. Na primeira fase os pares de entrada/saída são apresentados aleatoriamente à rede. Os valores modais e de dispersão são ajustados tal que todo o espaço de entrada seja coberto pelos grupos de neurônios. Assim, para qualquer entrada apresentada à RNN há sempre um grupo de neurônios i pertencente à segunda camada tal que $H^i(\mathbf{x}(t)) \neq 0$. A partir desta fase, os pesos $w_i(t)$ da quarta camada são ajustados a cada par entrada/saída apresentada à rede, correspondendo à segunda fase do algoritmo.

3.2 Aprendizado Adaptativo

O método de aprendizagem *adaptativo* capacita a RNN a adquirir novos conhecimentos codificando novas regras nebulosas em sua estrutura, alterando a arquitetura da

rede. Portanto, o número de regras nebulosas codificadas pela rede, correspondente ao número de grupos de neurônios na segunda camada, não é constante, ou seja, $N = N(t)$.

O algoritmo *adaptativo* utiliza fatores de aprendizagem análogos aos do algoritmo *offline*. Ou seja, para ajuste dos valores modais $c_j^i(t)$ das funções de pertinências $F_j^i(\cdot)$ e conseqüentes das regras nebulosas $w_i(t)$, utilizam-se os fatores de aprendizagem $\epsilon^i(t)$ e $\beta \in (0, 1)$, respectivamente. O fator $\epsilon^i(t)$ é definido da mesma maneira que no método *offline*, ou seja, $\epsilon^i(t) = \alpha/(\eta^i(t) + 1)$, onde $\eta^i(t)$ é o número de vezes que o valor modal foi ajustado até a t -ésima entrada e $\alpha \in (0, 1]$. Ao contrário do método *offline*, que possui um fator de ampliação, o método *adaptativo* utiliza um fator de redução $\gamma \in (0, 1)$ para ajuste da dispersão dos conjuntos nebulosos.

Quando a rede recebe uma nova entrada os grupos de neurônios competem entre si. Se um dos grupos vence e o desempenho é satisfatório então os parâmetros são ajustados. Se o desempenho não é satisfatório ou nenhum grupo vence uma nova regra é adicionada a estrutura da rede. Assim, duas situações são possíveis:

a) A entrada $\mathbf{x}(t)$ é caracterizada por pelo menos uma regra, isto é, $\exists i | H^i(\mathbf{x}(t)) \neq 0$. O grupo $\xi(t)$ vence a competição se $H^{\xi(t)}(\mathbf{x}(t)) \geq H^i(\mathbf{x}(t)), \forall i \in \{1, \dots, N(t)\}$. Se o modelo alcança o desempenho desejado, ou seja, $\|y(t) - y(t)\| \leq \delta$, onde δ é a precisão desejada, então os parâmetros são ajustados de acordo com:

$$c_j^{\xi(t)}(t+1) = c_j^{\xi(t)}(t) + \epsilon^{\xi(t)}(t)(x_j(t) - c_j^{\xi(t)}(t)),$$

$$j = 1, \dots, M \quad (13)$$

$$\eta^{\xi(t)}(t+1) = \eta^{\xi(t)}(t) + 1 \quad (14)$$

$$w_i(t+1) = w_i(t) + \beta \left(\frac{H^i(\mathbf{x}(t))(y(t) - y(t))}{\sum_{h=1}^{N(t)} H^h(\mathbf{x}(t))} \right),$$

$$i = 1, \dots, N(t) \quad (15)$$

$$r^i(t+1) = r^i(t), \quad i = 1, \dots, N(t) \quad (16)$$

$$N(t+1) = N(t) \quad (17)$$

Por outro lado, se o desempenho desejado não foi alcançado pelo modelo, uma nova regra é adicionada à estrutura da rede e os parâmetros são ajustados como segue:

Para $h = 1$ a $N(t)$ Faça:

Se $H^h(\mathbf{x}(t)) \neq 0$ Então:

$$\rho^h(t+1) = \gamma \rho^h(t) \quad (18)$$

Realiza-se a inserção de uma nova regra à estrutura da rede. Os novos valores modais são iguais a cada componente do vetor de entrada, a dispersão é igual à distância $D(\mathbf{c}^{\xi(t)}(t), \mathbf{x}(t))$ e o conseqüente é igual à saída desejada $y(t)$:

$$N(t+1) = N(t) + 1 \quad (19)$$

Para $i = N(t+1)$ Faça:

$$c_j^i(t+1) = x_j(t), \quad j = 1, \dots, M \quad (20)$$

$$r^i(t+1) = D(\mathbf{c}^{\xi(t)}(t), \mathbf{x}(t)) \quad (21)$$

$$w_i(t+1) = y(t) \quad (22)$$

$$\eta^i(t+1) = 0 \quad (23)$$

b) A entrada $\mathbf{x}(t)$ não é classificada por nenhuma regra, isto é, $H^i(\mathbf{x}(t)) = 0, \forall i \in \{1, \dots, N(t)\}$. Neste caso, a distância $D(\mathbf{c}^i(t), \mathbf{x}(t))$ é determinada entre o vetor de entrada e o vetor de valores modais $\mathbf{c}^i(t) = (c_1^i(t), \dots, c_M^i(t))'$ e o grupo vencedor $\xi(t)$ é tal que $D(\mathbf{c}^{\xi(t)}(t), \mathbf{x}(t)) \leq D(\mathbf{c}^i(t), \mathbf{x}(t)), \forall i \in \{1, \dots, N(t)\}$. Os ajustes das equações são:

$$N(t+1) = N(t) + 1 \quad (24)$$

Para $i = N(t+1)$ Faça:

$$c_j^i(t+1) = x_j(t), \quad j = 1, \dots, M \quad (25)$$

$$r^i(t+1) = D(\mathbf{c}^{\xi(t)}(t), \mathbf{x}(t)) \quad (26)$$

$$w_i(t+1) = y(t) \quad (27)$$

$$\eta^i(t+1) = 0 \quad (28)$$

Os métodos de aprendizagem *offline* e *adaptativo*, descritos nesta seção, capacitam a rede a adquirir novos conhecimentos quando necessário, fornecendo uma forma automática de aprender os parâmetros para um modelo nebuloso sem a interferência ou participação de um especialista. A próxima seção apresenta a aplicação dessas estratégias de aprendizagem no problema de previsão de vazões afluentes médias mensais 1 passo à frente.

4 PREVISÃO DE VAZÕES

Este trabalho tem como objetivo analisar o desempenho do modelo RNN proposto por (Figueiredo and Gomide, 1997) aplicado ao problema de previsão de vazões afluentes médias mensais. Para tanto, foi realizada uma análise comparativa entre os modelos de séries temporais, rede neural multi-camadas com algoritmo de retropropagação e a rede neural nebulosa com os algoritmos *offline* e *adaptativo*.

Para efeito de estudo de casos, foram utilizados os dados de duas usinas hidroelétricas Furnas e Emborcação, com reservatórios do sistema Sudeste Brasileiro localizados na bacia do rio Paraná, e a usina hidroelétrica de Sobradinho, com reservatório do sistema Nordeste localizada na bacia do rio São Francisco.

Os dados utilizados neste trabalho consistem em séries históricas de vazões naturais afluentes médias mensais que abrangem o período de 1931 a 1990, medidos em postos de medições de vazões, onde existe aproveitamento hidroelétrico. Esses conjuntos de dados compõem um banco de vazões históricas mantido pela Eletrobrás.

Para avaliar o desempenho dos modelos foram selecionados os últimos 5 anos (60 meses) do histórico de vazões, compreendendo o período de 1986 a 1990. Os modelos de séries temporais, rede neural artificial e rede neural nebulosa foram ajustados utilizando o período de janeiro de 1931 a dezembro de 1985.

As séries de vazões naturais médias mensais têm como características o comportamento periódico, onde os dados oscilam entre um valor mínimo, que ocorre no meses secos, e um máximo, que ocorre nos meses de cheia, podendo ser observado uma variação sazonal, com período aproximado de 12 meses. Este comportamento pode ser observado na Tabela 1 onde são apresentadas a média e o desvio padrão das séries de vazões, para as três usinas hidroelétricas.

A análise destes tipos de séries pode ser feita através do uso de formulações periódicas cujos parâmetros apresentam um comportamento periódico. A esta classe de modelos denomina-se auto-regressivos periódicos (Salas et al., 1980). Estes modelos são referenciados por modelos $PAR(p)$, onde p é a ordem do modelo. Em geral, p é um vetor, $p = (p_1, p_2, \dots, p_{12})$, onde cada elemento fornece a ordem de cada mês. Matematicamente, o modelo $PAR(p_1, p_2, \dots, p_{12})$, pode ser descrito da seguinte forma:

$$\hat{y}_{t(r,m)} = \mu_m + \sum_{i=1}^{p_m} \phi_{i,m} (y_{t(r,m)-i} - \mu_{m-i}) + a_{t(r,m)} \quad (29)$$

Tabela 1: Média e Desvio Padrão (DP) para as Séries de Vazões das Usinas Hidroelétricas de Furnas, Emborcação e Sobradinho (m^3/s). (M=1 : Janeiro, ..., M=12 : Dezembro)

M	Furnas		Emborcação		Sobradinho	
	Média	DP	Média	DP	Média	DP
1	1719.3	687.98	885.9	348.9	4858.9	1430.8
2	1669.2	671.34	907.0	413.1	5174.5	2062.7
3	1518.0	645.98	845.6	363.5	4937.5	2538.4
4	1019.6	358.67	625.8	236.7	3973.1	1787.8
5	743.5	238.96	403.3	125.3	2485.2	1228.7
6	623.0	278.65	308.8	92.6	1697.5	559.4
7	515.5	164.77	244.2	70.9	1401.6	384.2
8	431.2	135.05	189.9	57.2	1201.5	299.3
9	447.5	252.08	161.8	49.9	1063.2	263.9
10	532.4	242.19	206.9	67.9	1179.0	298.6
11	747.8	314.34	356.7	165.9	1909.2	667.8
12	1268.2	499.49	655.0	310.7	3477.9	1233.2

onde μ_m é a média no mês m , $\phi_{i,m}$ é a função periódica com período s , $a_{t(r,m)}$ é um ruído branco com distribuição normal $N(0, \sigma_m^2)$ sendo σ_m^2 a variância no mês m e $y_{t(r,m)}$ representa a série padronizada:

$$y_{t(r,m)} = \frac{x_{t(r,m)} - \mu_m}{\sigma_m} \quad (30)$$

sendo $x_{t(r,m)}$ a série sazonal e t é o índice de tempo escrito como função do número de anos $r = 1, 2, \dots, n$ e dos meses $m = 1, 2, \dots, s$.

Para o ajuste dos modelos PAR para os dados de vazões das três usinas hidroelétricas foram adotados duas estratégias. Na primeira, foi realizada a análise dos histogramas e das funções de autocorrelação e autocorrelação parcial para selecionar a ordem do modelo a ser ajustado. Neste caso, um modelo auto-regressivo periódico de mesma ordem p (PAR(p)) foi selecionado para todos os meses do ano e, para a estimação dos parâmetros, foi empregado o método de máxima verossimilhança.

A Figura 4 mostra as funções de autocorrelação e autocorrelação parcial para o posto de Furnas. A análise destas funções indica o ajuste de um modelo de ordem 2, ou seja, o ajuste de um modelo PAR(2). Prosseguindo da mesma forma, as análises dos histogramas e das funções de autocorrelação e autocorrelação parcial para os dados de vazões dos postos de Emborcação e Sobradinho indicaram o ajuste de um modelo de ordem 1 (PAR(1)).

A segunda estratégia adotada foi escolher a ordem do modelo mais adequada para cada mês (PAR(p_m)), onde p_m denota a ordem do modelo no mês m . Neste caso, foi utilizado inferência Bayesiana, sendo que os parâmetros

foram estimados através da densidade *a priori* e para a seleção da ordem do modelo foi usado a densidade preditiva (Ballini, 2000). A Tabela 2 mostra a ordem selecionada para cada mês e para os dados de vazões das três usinas hidroelétricas. Pode-se notar que, para os meses secos (Abril a Outubro) a ordem dos modelos selecionados, em geral, foi maior que para os meses de cheia (Novembro a Março) indicando que, durante o período seco a correlação com os meses anteriores é mais forte que para os meses úmidos.

Tabela 2: Ordem do modelo PAR(p_m).

Meses	Furnas	Emborcação	Sobradinho
Janeiro	1	1	1
Fevereiro	2	2	1
Março	1	1	1
Abril	3	2	1
Maio	3	3	1
Junho	1	1	5
Julho	2	2	2
Agosto	1	3	1
Setembro	4	1	1
Outubro	6	1	1
Novembro	1	2	1
Dezembro	2	1	1

Uma outra metodologia utilizada para comparação com o modelo RNN e os modelos de séries temporais, foi a rede neural multicamadas (*MultiLayer Perceptron-MLP*), com algoritmo de retro-propagação, proposto por (Rumelhart et al., 1986a). A rede MLP é uma das

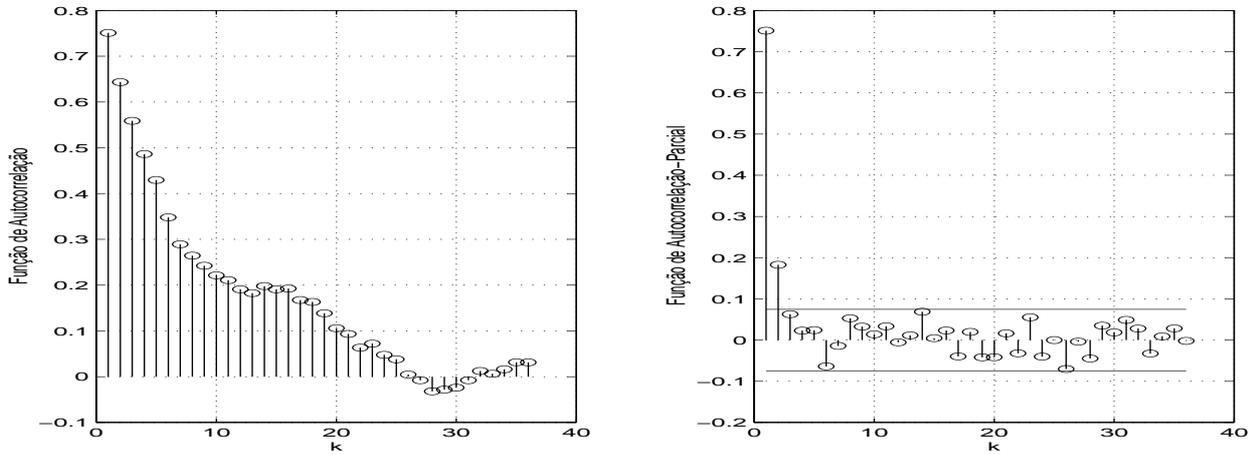


Figura 4: Funções de autocorrelação e autocorrelação parcial para a série de vazões do posto de Furnas.

mais conhecidas e aplicadas arquiteturas de redes neurais. Sua arquitetura (ou topologia) consiste de uma camada de entrada, uma ou mais camadas intermediárias e uma camada de saída. Os neurônios são conectados por arcos e a cada arco está associado um peso. A comunicação entre os neurônios é realizada através destas interconexões ponderadas. Cada neurônio da rede tem uma função não-decrescente e diferenciável, conhecida como função de ativação. Uma das funções de ativação mais utilizada em aplicações é a função logística, definida como segue:

$$y_j(t) = f_j(\nu_j(t)) = \frac{1}{1 + \exp^{-\nu_j(t)}} \quad (31)$$

onde $\nu_j(t)$ é o nível de ativação do neurônio j no instante t .

Para o treinamento da rede MLP, o método de aprendizagem mais utilizado é o algoritmo de retro-propagação do erro. Este algoritmo consiste em alterar os pesos da rede a fim de minimizar a diferença entre as saídas desejadas e as saídas fornecidas pelo modelo. Se o erro na camada de saída não é menor que uma dada tolerância, então o erro é retro-propagado através da rede, tendo como base para a atualização dos pesos a Regra Delta, a qual implementa o Método do Gradiente Descendente (Haykin, 1994). Um método simples para aumentar a velocidade de aprendizagem é modificar a Regra Delta, adicionando um termo *momentum*, como proposto por (Rumelhart et al., 1986a):

$$\Delta w_{ji}(n) = \alpha \Delta w_{ji}(n-1) + \eta \delta_j(n) y_i(n) \quad (32)$$

onde α é um número positivo, restrito ao intervalo

$[0,1)$, chamado constante *momentum* e η é a taxa de aprendizado. Algumas heurísticas foram sugeridas por (Jacobs, 1988) para acelerar a convergência do algoritmo através da taxa de aprendizado. Uma delas é utilizar um parâmetro diferente e variável para cada peso sináptico. Isto é realizado usando o Método do Gradiente Descendente para a pesquisa da direção para a atualização dos pesos (Haykin, 1994).

Neste trabalho, uma estrutura com 6 entradas e uma saída foi selecionada para os modelos RNN e MLP, dadas pelas vazões dos meses:

$$\begin{aligned} x_1(t) &= \text{vazão}(t-1, m-13); \\ x_2(t) &= \text{vazão}(t-1, m-12); \\ x_3(t) &= \text{vazão}(t-1, m-11); \\ x_4(t) &= \text{vazão}(t, m-3); \\ x_5(t) &= \text{vazão}(t, m-2); \\ x_6(t) &= \text{vazão}(t, m-1); \\ y(t) &= \text{vazão}(t, m); \end{aligned}$$

onde, $t = 1, 2, \dots$ é o ano, $m = 1, \dots, 12$ é o mês e $\text{vazão}(t, m)$ é a vazão média mensal do ano t e mês m . A seleção dessas entradas foi devido ao fato de fornecer ao modelo a tendência da série dada pelos três meses imediatamente anteriores ($m-1, m-2, m-3$) e também informações sobre a sazonalidade representada pelos três meses $m-11, m-12$ e $m-13$.

As seguintes inicializações foram adotadas para a rede MLP: para os pesos foram atribuídos valores aleatórios com distribuição uniforme e de magnitude pequena; termo *momentum* $\alpha = 0.9$; número de iterações 1000; e taxa de aprendizado, η , diferente para diferentes partes da rede, dependendo da conexão. Para avaliar o desempenho dos modelos MLP foi fixado uma tolerância $\delta = 0.001$. A Tabela 3 mostra a topologia obtida para as três usinas hidroelétricas. O número de neurônios foi maior para o período de cheia (Novembro a Abril),

pois os desvios padrões (Tabela 1) durante estes meses são maiores que nos meses de seca, resultando em uma maior variação nos dados.

Tabela 3: Configuração da rede MLP para cada mês.

Meses	Furnas	Emborcação	Sobradinho
Janeiro	6/19/1	6/17/1	6/18/1
Fevereiro	6/19/1	6/22/1	6/21/1
Março	6/19/1	6/17/1	6/21/1
Abril	6/9/1	6/14/1	6/19/1
Maió	6/9/1	6/15/1	6/15/1
Junho	6/9/1	6/8/1	6/9/1
Julho	6/9/1	6/4/1	6/9/1
Agosto	6/9/1	6/4/1	6/9/1
Setembro	6/9/1	6/5/1	6/6/1
Outubro	6/12/1	6/5/1	6/12/1
Novembro	6/12/1	6/12/1	6/22/1
Dezembro	6/18/1	6/23/1	6/18/1

Para as funções de pertinência das regras nebulosas codificadas na RNN foram selecionadas as funções Gaussianas, definidas como segue:

$$F_j^i(x_j) = \begin{cases} 0, & \text{se } |x_j - c_j^i| > 2r^i \\ \exp(-|x_j - c_j^i|/r^i), & \text{caso contrário} \end{cases} \quad (33)$$

A taxa de aprendizagem para atualização dos pesos w_i foi fixada em $\beta = 0.5$ e o parâmetro $\alpha = 1.0$ foi utilizado para atualização do fator de aprendizagem $\epsilon^i(t)$, para os dois métodos de aprendizagem. Os neurônios da segunda camada possuem como operador sináptico a t -norma mínimo e como operador de agregação a s -norma máximo. Os neurônios na terceira camada possuem como operador de agregação a t -norma produto.

Um fator de ampliação $\lambda = 1.1$ para o ajuste das dispersões das regras nebulosas foi adotado para o algoritmo *offline*. Os valores modais c_j^i e de dispersão r^i são inicializados aleatoriamente e o número de regras para cada mês está apresentado na Tabela 4.

Para a RNN com o algoritmo *adaptativo* o fator de redução $\gamma = 0.9$ foi considerado e a inicialização da dispersão das funções de pertinência foi adotada $r^i(0) = 1.0$. O número de regras nebulosas iniciais foi $N(0)=1$. Os grupos de neurônios foram adicionados à estrutura da rede para um desempenho escolhido como $\delta = 0.09$, ou seja, novas regras foram adicionadas quando

$$\|y(t) - \hat{y}(t)\| > 0.09$$

Tabela 4: Número de regras para o modelo *Offline*.

Meses	Furnas	Emborcação	Sobradinho
Janeiro	15	18	18
Fevereiro	18	33	22
Março	12	12	12
Abril	9	9	12
Maió	6	9	9
Junho	6	9	9
Julho	6	9	9
Agosto	6	9	9
Setembro	18	18	18
Outubro	15	18	18
Novembro	12	22	22
Dezembro	12	12	12

A Tabela 6 mostra o número de regras que foram ajustados para cada mês e para cada usina hidrelétrica. Novamente, pode-se observar que o número de regras é maior durante os meses de cheia do ano (Novembro a Abril) porque durante este período as vazões apresentam grandes variações. Este mesmo comportamento foi observado quando foi ajustado o modelo MLP e PAR.

Tabela 5: Número de regras para o modelo *Adaptativo*.

Meses	Furnas	Emborcação	Sobradinho
Janeiro	36	38	28
Fevereiro	39	33	36
Março	31	35	27
Abril	28	37	39
Maió	25	27	20
Junho	12	28	19
Julho	16	25	24
Agosto	14	16	22
Setembro	18	26	21
Outubro	21	24	28
Novembro	25	29	39
Dezembro	36	35	37

Para avaliar o desempenho dos modelos ajustados neste trabalho são analisados os erros quadrático médio (*EQM*), absoluto médio (*EAM*) e relativo percentual médio (*ERM*), definidos como segue:

$$EQM = \frac{1}{P} \sum_{j=1}^P (y_j - \hat{y}_j)^2 \quad (34)$$

$$EAM = \frac{1}{P} \sum_{j=1}^P |y_j - \hat{y}_j| \quad (35)$$

$$ERM(\%) = \frac{100}{P} \sum_{j=1}^P \frac{|y_j - \hat{y}_j|}{y_j} \quad (36)$$

onde $P = 1, \dots, 60$ é o número de padrões, y_j é o valor real e \hat{y}_j é o valor previsto em relação ao mês j .

As Tabelas 6, 7, e 8 mostram os erros da aplicação dos modelos para as usinas hidroelétricas de Furnas, Emborcação e Sobradinho, respectivamente, para previsão 1 passo à frente. Em todos os casos e para todos os critérios, a RNN resultou em um melhor resultado que a MLP para as topologias escolhidas e os modelos PAR(p_m) e PAR(2) fornecendo uma redução significativa nos erros de previsão. Também pode-se notar que o ajuste de um modelo PAR(p_m) fornece melhores resultados que o ajuste de um modelo de ordem fixa para todo mês m (PAR(2)). Nota-se que tanto os modelos MLP como o modelos RNN apresentaram melhor desempenho que os modelos de séries temporais.

Tabela 6: Estatísticas para a usina hidroelétrica de Furnas.

Modelos	EQM	EAM	ERM(%)
<i>Offline</i>	$3,53 \times 10^4$	125,54	16,89
<i>Adaptativo</i>	$2,26 \times 10^4$	91,51	13,02
<i>RNA</i>	$6,67 \times 10^4$	169,36	20,49
<i>PAR(2)</i>	$6,12 \times 10^4$	173,8	22,7
<i>PAR(p_m)</i>	$5,93 \times 10^4$	168,09	21,84

Tabela 7: Estatísticas para a usina hidroelétrica de Emborcação.

Modelos	EQM	EAM	ERM(%)
<i>Offline</i>	$1,55 \times 10^4$	80,91	16,09
<i>Adaptativo</i>	$1,41 \times 10^4$	68,76	15,68
<i>RNA</i>	$2,29 \times 10^4$	92,16	19,52
<i>PAR(1)</i>	$5,52 \times 10^4$	138,5	27,3
<i>PAR(p_m)</i>	$3,02 \times 10^4$	98,18	19,54

Tabela 8: Estatísticas para a usina hidroelétrica de Sobradinho.

Modelos	EQM	EAM	ERM(%)
<i>Offline</i>	$5,81 \times 10^5$	461,97	18,91
<i>Adaptativo</i>	$5,58 \times 10^5$	401,60	16,87
<i>RNA</i>	$1,06 \times 10^6$	603,10	21,10
<i>PAR(1)</i>	$1,99 \times 10^6$	808,5	31,1
<i>PAR(p_m)</i>	$1,33 \times 10^6$	601,76	22,09

5 CONCLUSÕES

Este trabalho apresentou uma classe de rede neural nebulosa para previsão de vazões naturais afluentes médias mensais. Dois algoritmos de treinamento baseados no paradigma de aprendizado competitivo foram utilizados pela rede neural nebulosa. No primeiro, o número de regras foi definido no início do treinamento; no segundo, grupos de neurônios são adicionados à estrutura da rede quando novos conhecimentos são necessários. Estes modelos aprendem os parâmetros fundamentais para modelar estes sistemas. Os resultados da aplicação destes modelos para previsão de vazões médias mensais 1 passo à frente foram comparados com os resultados obtidos pelos modelos de rede neural artificial e modelos de séries temporais, fornecendo uma melhor previsão 1 passo à frente, com erros em torno de 40% menores que os modelos periódicos auto-regressivos. Quando comparados os dois métodos de aprendizado, o algoritmo *adaptativo* apresentou melhores resultados, porém o número de regras nebulosas ajustadas foi maior que para o algoritmo *offline*.

AGRADECIMENTOS

Esta pesquisa teve o suporte financeiro da Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP), e do Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq).

REFERÊNCIAS

- Atiya, A. F., El-Shoura, S. M., Shaheen, S. I. and El-Sherif, M. S. (1999). A comparison between neural-network forecasting techniques - case study: River flow forecasting, *IEEE Transactions on Neural Networks* **10**(2): 402–409.
- Bakirtzis, A. G., Theocharis, J. B., Kiartzis, S. J. and Satsios, K. J. (1995). Short term load forecasting using fuzzy neural networks, *IEEE Transactions on Power Systems* **10**(3): 1518–1524.
- Ballini, R. (2000). *Análise e Previsão de Vazões Utilizando Modelos de Séries Temporais, Redes Neurais Artificiais e Redes Neurais Nebulosas*, Tese de Doutorado, Faculdade de Engenharia Elétrica e de Computação, Unicamp.
- Ballini, R., Soares, S. and Andrade, M. G. (1999). Seasonal streamflow forecasting via a neural fuzzy system, *Proceedings of International Federation of Automatic Control - IFAC'99*, Vol. k, pp. 81–86.

- Box, G. E. P., Jenkins, G. M. and Reinsel, G. C. (1994). *Time Series Analysis: Forecasting and Control*, Holden Day, 3th Edição.
- Figueiredo, M. and Gomide, F. (1997). Adaptive neuro fuzzy modelling, *Proceedings of the 6th IEEE Conference on Fuzzy Systems*, Barcelona – Espanha, pp. 1567–1572.
- Gomide, F., Figueiredo, M. and Pedrycz, W. (1995). A neural fuzzy network: Structure and learning, *Fuzzy Logic and Its Applications, Information Sciences and Intelligent Systems*, Bien, Z. and Min, K., Kluwer Academic Publishers, Netherlands, pp. 177–186.
- Haykin, S. (1994). *Neural Networks - A comprehensive Foundation*, IEEE Press, New York.
- Jacobs, R. A. (1988). Increased rates of convergence through learning rate adaptation, *Neural Networks*, Vol. 1, pp. 295–307.
- Kacprzyk, J. and Fedrizzi, M. (1992). *Fuzzy regression Analysis. Studies in Fuzziness*, Vol. 1, Physica-Verlag, Heidelberg.
- Lachtermacher, G. and Fuller, J. D. (1995). Backpropagation in time-series forecasting, *Journal of Forecasting* **14**(4): 381–393.
- MecLeod, A. I. (1994). Diagnostic checking of periodic autoregression, *Journal of Time Series Analysis* **15**(2): 221–223.
- Pedrycz, W. and Gomide, F. (1998). *An Introduction to Fuzzy Sets: Analysis and Design*, MIT Press, Cambridge, MA.
- Rumelhart, D., Hinton, G. E. and Williams, R. (1986a). Learning representations by back-propagating errors, *Nature (London)* **323**: 533–536.
- Salas, J. D., Delleur, J. W., Yevjevich, V. and Lane, W. L. (1980). Applied modeling of hydrologic time series, *Water Resources Publications* .
- Vecchia, A. V. (1985). Maximum likelihood estimation for periodic autoregressive-moving average models, *Technometrics* **27**(4): 375–384.
- Weigend, A. S., Rumelhart, D. E. and Huberman, B. A. (1990). Predicting the future: A connectionist approach, *International Journal Neural System* **1**: 3.
- Yager, R. and Filev, D. (1994). *Essentials of Fuzzy Modeling and Control*, Wiley Interscience, New York.
- Zealand, C. M., Burn, D. H. and Simonovic, S. P. (1999). Short term streamflow forecasting using artificial neural networks, *Journal of Hydrology* **214**: 32–48.