

Previsão de Vazões Naturais Médias Mensais Usando uma Rede Neural Nebulosa Adaptativa

Rosângela Ballini¹, Secundino Soares¹, Marinho Gomes Andrade²

¹Universidade Estadual de Campinas
Faculdade de Engenharia Elétrica e de Computação
13083-970 Campinas - SP - Brasil
E-mails: ballini@dca.fee.unicamp.br, dino@denis.fee.unicamp.br

²Instituto de Ciências Matemáticas e de Computação
Universidade de São Paulo
13560-970 São Carlos - SP- Brasil
E-mail: marinho@icmc.sc.usp.br

Abstract

Analysis and forecast of seasonal streamflow series are of utmost importance in the operation planning of water resources systems. One of the greatest difficulties in forecasting of those series is the seasonality nature of streamflow series due to dry periods of the year. This work suggests the application of neurofuzzy network model to seasonal streamflow forecasting. This model learns membership functions parameters for each input variable from training data, processes data automatically adjusted to cover the whole input space. The problem of seasonal streamflow forecasting is solved using a database of average monthly inflows of one Brazilian hydroelectric plant located at Grande River. Comparison of the neurofuzzy model with multilayer neural network and periodic autoregressive models are also included to illustrate the performance of the approach. The results show that the models here proposed provide a better performance than the other ones considering one-step-ahead forecasting and multi-step-ahead forecasting, with forecasting errors significantly lower than the other approaches.

1. Introdução

A produção energética de um sistema hidroelétrico depende da série de vazões afluentes às diversas usinas do sistema. Por exemplo, quando se adota um aproveitamento hídrico, a série histórica de vazões no local é usualmente utilizada como dado de entrada para um modelo de simulação e/ou otimização, com intervalos de discretização mensal.

As séries de vazões naturais médias mensais têm como características o comportamento periódico, como por exemplo a média, a variância, a assimetria e a estrutura de auto-correlação. A análise destes tipos de séries pode ser feita pelo uso de formulações auto-regressivas cujos parâmetros apresentam um comportamento periódico. A esta classe de modelos costuma-se denominar mo-

delos auto-regressivos periódicos [1]. O modelo auto-regressivo periódico é proposto para ser utilizado no Modelo Estratégico de Geração Hidrotérmica a Subsistemas Equivalentes Interligados (NEWAVE) [2].

Recentemente, redes neurais artificiais tornaram-se extremamente conhecidas para previsão em várias áreas, incluindo finanças, carga elétrica e recursos hídricos. A utilização de redes neurais para previsão de séries temporais vem sendo proposta por diversos autores, entre eles [3], mostrando a viabilidade de utilizar estes modelos.

A comparação de redes neurais com outras abordagens ainda são contraditórias. As explicações para tais contradições podem ser descritas por diferentes fatores, tais como: a estrutura da rede neural, o tipo de série (estacionária, não-estacionária) usada nos estudos e a relação do tamanho da rede e o número de entradas das séries temporais.

Freqüentemente, dados do mundo real apresentam ruídos, podendo conter contradições e imperfeições. Tolerância a imprecisão e incertezas é também exigido para considerar tratabilidade e robustez. Sob essas hipóteses, modelos de análise de dados baseados em conjuntos nebulosos vêm sendo empregados atualmente em diversas áreas [4].

Os modelos de redes neurais nebulosas fornecem estas características, pois, diferente das abordagens clássicas, estes modelos abstraem o conhecimento físico dos sistemas a ser modelado, já que o modelo pode ser gerado a partir dos dados de entrada e saída, e ao se manter ativo o processo de geração do modelo, criam-se condições para eliminar deficiências causadas por variações dos parâmetros. Estes sistemas vêm sendo aplicados sucessivamente para sistemas de identificação não-linear [5], controle de processos [6], reconhecimento de padrões [7], previsão de carga [8], previsão de vazões [9].

Este trabalho propõe a aplicação de um modelo de rede neural nebulosa com estrutura adaptativa, proposto em [10]. Este modelo apresenta um método de apren-

dizagem construtivo, onde regras nebulosas são adicionadas à estrutura da rede quando o desempenho desta não é satisfatório. Para avaliar o desempenho do modelo, uma análise comparativa foi realizada com os modelos de redes neurais multi-camadas com algoritmo de retro-propagação e auto-regressivo periódico para previsão um passo à frente e vários passos à frente para os dados de vazões afluentes da usina hidroelétrica de Furnas. Os resultados mostraram que o modelo proposto apresentou melhor performance que as outras abordagens tanto para previsão um passo à frente como para previsão vários passos à frente.

2. Estrutura da Rede Neural Nebulosa

A estrutura da Rede Neural Nebulosa (RNN), proposta em [10], foi desenvolvida com base em duas características essenciais: o mapeamento das regras nebulosas na estrutura é direto e o processamento neural deve equivalever totalmente a um mecanismo de inferência nebulosa. 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 [11].

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:

Entrada: x_1 é A_1 e ... x_M é A_M

Se x_1 é A_1^1 e ... x_M é A_M^1 Então y é w_1

Regras: Se x_1 é A_1^i e ... x_M é A_M^i Então y é w_i

Se x_1 é A_1^N e ... x_M é A_M^N Então y é w_N

Saída: $y = y(\mathbf{x})$

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 sendo 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 vetor de entrada $\mathbf{x} = x_1, \dots, x_j, \dots, x_M$, está associado um único peso sináptico w_i .

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 $x_j \in I_{jk} = (x_{jk_l}, x_{jk_f}]$, onde I_{jk} é o k -ésimo intervalo de discretização em relação à entrada $x_j, k = 1, \dots, Q$.

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

definido como sendo o valor da função de pertinência do conjunto nebuloso A_j^i , isto é, $A_j^i(x_j) = a_{jk}^i$, se $x_j \in I_{jk}$.

A implementação do mecanismo de inferência nebulosa pode ser obtida de várias formas [11]. Em [10] foi utilizado um dos métodos mais conhecidos, proposto por [12], 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:

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

onde S e T são operadores, correspondentes a uma s -norma e uma t -norma, respectivamente.

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:

$$H^i(\mathbf{x}) = T \left\{ P_j^i(\mathbf{x}) \right\} \quad (2)$$

3. *Agregação das Regras*: A saída y do sistema de inferência, ou seja, a etapa de defuzzificação dos dados, é realizada através de uma média ponderada entre os conseqüentes e os níveis de ativação calculados em (2) para cada regra, dada por:

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

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.

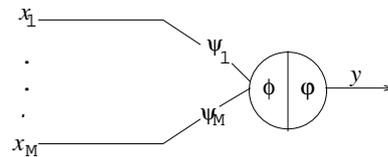


Figura 1: Modelo Geral do Neurônio Nebuloso

O modelo RNN apresenta uma arquitetura não recorrente com 5 camadas, como mostra a figura (2). 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.

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 $a_{jk} = \phi(x_j)$. Para um intervalo $I_{jk} = (x_{jk_I}, x_{jk_F}]$, a função decodificada $\phi(\cdot)$ para este neurônio, através do k -ésimo intervalo, mostrada na figura (3), é dada por:

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

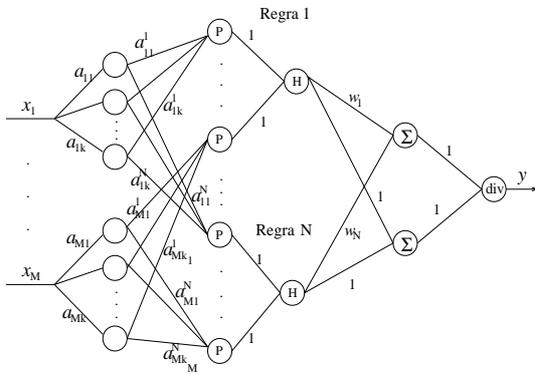


Figura 2: Rede Neural Nebulosa.

A segunda camada contém N grupos (número de regras pré-definido), 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^i_{jk} , a função de pertinência do conjunto nebuloso A^i_j . 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 comparação P^i_j entre os conjuntos nebulosos A_j e A^i_j . 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^i_{jk} . Considerando-se que $\psi = t\text{-norma}$, $\phi = s\text{-norma}$ e φ é a função identidade, a saída do j -ésimo neurônio é a medida de possibilidade P^i_j 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 que $\psi =$ função identidade, $\phi = t\text{-norm}$ e $\varphi =$ função identidade.

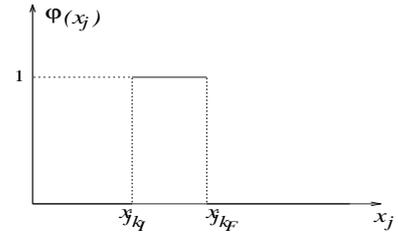


Figura 3: Função de Decodificação.

A quarta camada possui dois neurônios, ambos com N entradas. Ou seja, cada neurônio i da terceira camada está conectado com os dois neurônios da quarta camada. 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 $y(\mathbf{x})$ (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 da 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. Método de Aprendizagem

A estratégia de aprendizagem da 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 a^i_{jk} 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 w_i da

quarta camada.

O algoritmo de aprendizagem consiste na apresentação dos pares de entrada/saída desejada a rede, sendo o t -ésimo par dado por $(\mathbf{x}(t), y_d(t))$, onde $\mathbf{x}(t) = (x_1(t), \dots, x_M(t))'$ é o vetor de entrada com M coordenadas e $y_d(t)$ é a saída desejada.

Considera-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 ρ^i . Determinando-se o valor modal e de dispersão de cada função de pertinência $F_j^i(\cdot)$ e definidos os intervalos de discretização I_{jk} , é possível determinar os pesos sinápticos a_{jk}^i na forma $a_{jk}^i = F_j^i(x_j)$, tal que $x_j \in I_{jk} = [x_{jk_L}, x_{jk_F}]$.

Para o ajuste dos valores modais $c_j^i(t)$ das funções de pertinências $F_j^i(\cdot)$ e conseqüentes das regras nebulosas $w_i(t)$ são utilizados os fatores de aprendizagem $\beta^i(t)$ e $\eta \in (0, 1)$, respectivamente. O fator $\beta^i(t)$ é definido como sendo uma função monotônica decrescente positiva, tal que $0 < \beta^i(t) < 1$. Para ajuste da dispersão ρ^i é empregada uma taxa de redução $\gamma \in (0, 1)$.

Este método de aprendizagem capacita a RNN a adquirir novos conhecimentos codificando novas regras nebulosas em sua estrutura. Portanto, o número de regras nebulosas codificadas pela rede, correspondente ao número de grupos de neurônios na segunda camada, não é constante, ou seja, $N = N(t)$.

Os grupos de neurônios da segunda camada competem entre si sempre que a rede recebe uma nova entrada $\mathbf{x}(t)$. O grupo do neurônio vencedor guia os procedimentos de ajuste dos pesos sinápticos, podendo ocorrer duas situações:

(a) $\exists i \mid H^i(\mathbf{x}(t)) \neq 0$. A regra $\xi(t)$ vence a competição se $H^{\xi(t)}(\mathbf{x}(t)) \geq H^i(\mathbf{x}(t)), \forall i \in \{1, \dots, N(t)\}$. Se a saída da rede satisfizer o desempenho desejado, ou seja $\|y_d(t) - y(t)\| \leq \delta$, então os parâmetros são ajustados de acordo com as seguintes equações:

$$c_j^{\xi(t)}(t+1) = c_j^{\xi(t)}(t) + \beta^{\xi(t)}(x_j(t) - c_j^{\xi(t)}(t)) \quad (4)$$

$$\rho^i(t+1) = \rho^i(t), \quad i = 1, \dots, N(t) \quad (5)$$

$$w_i(t+1) = w_i(t) + \eta \left(\frac{H^i(\mathbf{x}(t))(y_d(t) - y(t))}{\sum_{h=1}^{N(t)} H^h(t)(\mathbf{x}(t))} \right) \quad (6)$$

para todo $i = 1, \dots, N(t)$

$$N(t+1) = N(t) \quad (7)$$

Por outro lado, se a saída não satisfizer o desempenho desejado, $\|y_d(t) - y(t)\| > \delta$, uma nova regra é adicionada a estrutura da rede e os parâmetros são ajustados como segue:

$$c_j^i(t+1) = c_j^i(t), \quad i = 1, \dots, N(t), \quad j = 1, \dots, M \quad (8)$$

A dispersão de cada função $F_j^i(\cdot)$ é reduzida para que as regras existentes deixem de influenciar no resultado da entrada $\mathbf{x}(t)$, ou seja:

Para $i = 1$ até $N(t)$ faça:

Se $H^i(\mathbf{x}(t)) \neq 0$ Então:

$$\rho^i(t) = \gamma \rho^i \quad (9)$$

$$\rho^i(t+1) = \rho^i(t) \quad (10)$$

$$w_i(t+1) = w_i(t) \quad (11)$$

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:

$$N(t+1) = N(t) + 1 \quad (12)$$

Para $i = N(t+1)$ faça:

$$c_j^i(t+1) = x_j(t), \quad j = 1, \dots, M \quad (13)$$

$$\rho^i(t+1) = D(\mathbf{c}^{\xi(t)}(t), \mathbf{x}(t)) \quad (14)$$

$$w_i(t+1) = y_d(t) \quad (15)$$

(b) $H^i(\mathbf{x}(t)) = 0, \forall i \in \{1, \dots, N(t)\}$. 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 e o grupo vencedor $\xi(t)$ é tal que $D(\mathbf{c}^{\xi(t)}(t), \mathbf{x}(t)) \leq D(\mathbf{c}^i(t), \mathbf{x}(t)), \forall i \in \{1, \dots, N(t)\}$. A estrutura da rede é alterada, acrescentado uma nova regra capaz de codificar o padrão $(\mathbf{x}(t), y_d(t))$. Os ajustes são realizados da seguinte forma:

$$N(t+1) = N(t) + 1 \quad (16)$$

Para $i = N(t+1)$ faça:

$$c_j^i(t+1) = x_j^i(t), \quad j = 1, \dots, M \quad (17)$$

$$\rho^i(t+1) = D(\mathbf{c}^{\xi(t)}(t), \mathbf{x}(t)) \quad (18)$$

$$w_i(t+1) = y_d(t) \quad (19)$$

O método de aprendizagem descrito nesta seção, capacita 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 dessa estratégia de aprendizagem no problema de previsão de vazões médias mensais, realizando um análise comparativa com a rede neural artificial (RNA) e os modelos periódicos auto-regressivos (PAR).

4. Previsão de Vazões

Neste trabalho, foram utilizados os dados de vazões naturais médias mensais da usina hidroelétrica de Furnas, com reservatório no sistema Sudeste brasileiro, localizada no Rio Grande, pertencente à empresa FURNAS - Centrais Elétricas S.A..

Os dados de vazões foram ajustados aos modelos PAR, RNA multi-camadas com algoritmo de retropropagação e ao modelo RNN apresentado neste trabalho. Para todos os modelos foram utilizados dados de vazões naturais médias mensais do histórico de 1931 a 1990. Para avaliar o desempenho dos modelos foi escolhido um período úmido, correspondendo ao período de Janeiro/1991 a Dezembro/1985, ou seja, foram considerados 60 meses para teste. Este período foi retirado do histórico e os modelos foram ajustados para o restante dos dados. Após o ajuste dos modelos, foram realizadas previsões 1, 3, 6 e 12 passos à frente para o período de 1981 a 1985.

Os modelos PAR(p_m), onde p_m é a ordem do modelo para cada mês m , são expressos da seguinte forma:

$$y_{t(r,m)} = \hat{\mu}_m + \sum_{i=1}^{p_m} \phi_{i,m} (y_{t(r,m)-i} - \hat{\mu}_{m-i}) + a_{t(r,m)}$$

onde $y_{t(r,m)}$ é a vazão no tempo $t(r,m) = 12(r-1) + m$, sendo r o número de anos e m o mês; $\hat{\mu}_m$ é a estimativa da vazão média no mês m . Tanto o parâmetro $\hat{\mu}_m$ e $\phi_{i,m}$ são funções periódicas e $a_{t(r,m)}$ é um ruído branco com distribuição normal $N(0, \tau_m^{-1})$, onde $\tau_m^{-1} = \sigma_m^2$ é o inverso da variância no mês m .

A ordem do modelo p_m foi determinada através da análise da densidade preditiva e os parâmetros $\phi_{i,m}$ e τ_m^{-1} foram estimados utilizando inferência Bayesiana [13]. A tabela (1) apresenta a ordem do modelo e os valores das estimativas dos parâmetros $\hat{\phi}_{i,m}$ e τ_m . Pode-se verificar que a ordem do modelo é maior para os meses de seca (Abril a Outubro), devido à forte correlação entre estes meses.

Tabela 1: Ordem e Estimativas dos Parâmetros Usando Inferência Bayesiana para a Série de Vazões de Furnas (M 1 = Janeiro; M 12 = Dezembro)

M	p_m	$\hat{\phi}_{1,m}$	$\hat{\phi}_{2,k}$	$\hat{\phi}_{3,k}$	$\hat{\phi}_{4,k}$	τ_m
1	1	0.415	-	-	-	1.095
2	2	0.495	0.036	-	-	1.479
3	2	0.516	0.232	-	-	1.529
4	3	0.572	0.189	0.243	-	2.736
5	3	0.532	0.211	0.263	-	5.158
6	3	0.713	0.267	-0.101	-	3.981
7	3	0.579	0.357	0.026	-	8.825
8	4	0.914	-0.038	0.244	0.199	9.394
9	4	0.622	0.172	0.636	-0.565	4.733
10	3	0.484	-0.223	0.580	-	3.313
11	1	0.883	-	-	-	2.487
12	2	0.349	0.386	-	-	1.803

Os parâmetros dos modelos RNA e RNN foram ajustados utilizando uma estrutura com 6 entradas, dadas pelas vazões dos meses $t-1, t-2, t-3, t-11, t-12, t-13$. A seleção dessas entradas foi devido ao fato de fornecer ao modelo a tendência da série dada pelos três meses imediatamente anteriores ($t-1, t-2, t-3$) e também informações sobre a sazonalidade representada pelos três meses $t-11, t-12$ e $t-13$.

Para a seleção adequada dos parâmetros livres da RNA para um determinado arquivo de dados, foi utilizado um conjunto de arquiteturas candidatas e através da técnica de validação cruzada os parâmetros foram estimados, verificando o desempenho dos modelos sob o conjunto de validação, sendo escolhido o modelo que apresentar melhor desempenho em relação ao erro quadrático médio (EQM).

Um modelo de RNA para cada mês foi ajustado pois, assim, o modelo torna-se mais específico para prever a vazão do mês m . A tabela (2) mostra o número de neurônios intermediários para cada mês para o período de teste de 1981 a 1985. Neste caso, os parâmetros taxa de aprendizado foram obtidos pela regra delta-bar-delta e o termo *momentum* foi fixado em $\alpha = 0.5$ [13].

Tabela 2: Número de Neurônios Intermediários da RNA para cada Mês

Meses	Neurônios	Meses	Neurônios
Janeiro	6	Julho	4
Fevereiro	3	Agosto	3
Março	3	Setembro	4
Abril	3	Outubro	3
Mai	4	Novembro	4
Junho	5	Dezembro	4

Para o ajuste da RNN os dados de vazões foram padronizados para o intervalo $[0, 1]$. Esta padronização é realizada para simplificar a entrada na rede pois, desta forma, pode-se considerar que a primeira camada da RNN contém M grupos e cada grupo contém 1 neurônio.

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

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

A taxa de aprendizagem para atualização dos pesos w_i foi fixada em $\eta = 0.5$ e foi utilizado como fator de aprendizagem a função $\beta^i(t) = (1/(\alpha^i + 1))$, onde α^i é número de vezes que os valores modais foram atualizados, para cada regra i . 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. O fator de redução $\gamma = 0.9$ foi considerado e a inicialização da dispersão das funções de pertinência foi adotado $\Theta = 2.0$. O número de regras

nebulosas iniciais foi $N(0)=1$. Os grupos de neurônios foram adicionados à estrutura da rede para um desempenho $\delta = 0.09$. Ao final do treinamento foram ajustadas 27 regras à estrutura da rede.

Para avaliar o desempenho dos modelos estudados neste trabalho são analisados os erros quadrático médio (*EQM*), absoluto médio (*EAM*) e relativo percentual médio (*EPM*). A tabela (3) mostra os erros para as previsões 1, 3, 6 e 12 passos à frente.

Tabela 3: Estatísticas para os Modelos de Previsão

Modelos	Passos	<i>EQM</i> ($\times 10^5$)	<i>EAM</i>	<i>EPM</i>
PAR(p_m)	1	2.15	302.11	21.28
	3	5.11	466.92	30.83
	6	5.65	540.75	31.56
	12	5.44	516.95	30.81
RNA	1	3.73	429.51	30.56
	3	4.79	539.67	39.62
	6	5.21	559.67	38.16
	12	6.18	574.80	38.17
RNN	1	0.63	172.91	14.63
	3	1.21	236.57	15.83
	6	2.09	373.87	21.99
	12	3.16	409.78	27.75

Analisando a tabela (3), pode-se observar que o modelo RNN apresentou melhor desempenho tanto para previsão 1 passo à frente como para previsão vários passos à frente. Para previsão 1 passo à frente o *EQM* para o modelo RNN é 3 vezes menor em relação ao modelo PAR(p_m) e 5 vezes menor em relação a RNA ajustada.

A figura (4) mostra os erros *EPM* para os três modelos ajustados e para os vários passos à frente, onde pode-se observar que a variação no *EPM* quando o horizonte de previsão cresce.

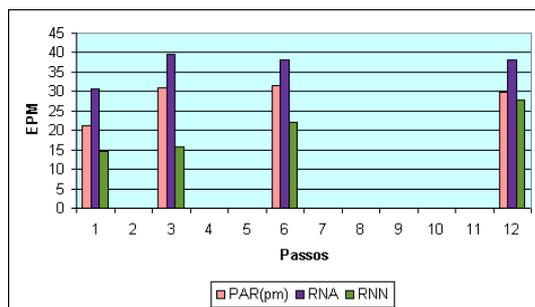


Figura 4: Erro Percentual Médio.

5. Conclusões

Este trabalho apresentou uma classe de rede neural nebulosa para previsão de vazões naturais médias mensais. A estrutura do modelo utilizado é adaptativa e um algoritmo de treinamento baseado no paradigma de aprendizado competitivo foi utilizado, onde grupos de neurônios são adicionados à estrutura da rede quando

novos conhecimentos são necessários. Foram realizadas previsões 1, 3, 6 e 12 passos à frente e os resultados da aplicação do modelo de rede neural nebulosa foram comparados com os resultados obtidos pelos modelos de rede neural artificial e auto-regressivo periódico. O modelo de rede neural nebulosa apresentou um desempenho superior em relação aos outros modelos com erros significativamente menores tanto para previsão um passo à frente como para previsão vários passos à frente.

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

- [1] J. D. Salas, J. W. Delleur, V. Yevjevich, and W. L. Lane. Applied modeling of hydrologic time series. *Water Resources Publications*, 1980.
- [2] CEPEL. *Centro de Pesquisas de Energia Elétrica. Modelo de Séries Hidrológicas. Manual de Metodologia.*, Relatório Técnico 113-81-A, 1983.
- [3] A. S. Weigend, D. E. Rumelhart, and B. A. Huberman. Predicting the future: A connectionist approach. *International Journal Neural System*, 1:3, 1990.
- [4] J. Kacprzyk and M. Fedrizzzi. *Fuzzy regression Analysis. Studies in Fuzziness*, volume 1. Physica-Verlag, Heidelberg, 1992.
- [5] L. X. Wang and J. M. Mendel. Back-propagation fuzzy system as non-linear dynamic system identifiers. In *Proceedings of FUZZ-IEEE'92*, pages 1409–1418, 1992.
- [6] C. C. Lee. Fuzzy logic control in controller - part i and ii. *IEEE Transactions on Systems, Man and Cybernetics*, 20(2):404–435, December 1990.
- [7] W. Caminhas, H. Tavares, F. Gomide, and W. Pedrycz. Fuzzy set based neural networks: Structure, learning and application. *Journal of Advanced Computational Intelligence*, 3(3):151–157, 1999.
- [8] A. G. Bakirtzis, J. B. Theocharis, S. J. Kiartzis, and K. J. Satsios. Short term load forecasting using fuzzy neural networks. *IEEE Transactions on Power Systems*, 10(3):1518–1524, August 1995.
- [9] R. Ballini, M. Figueiredo, S. Soares, M. Andrade, and F. Gomide. A seasonal streamflow forecasting model using neurofuzzy network. In *Information, Uncertainty and Fusion*, pages 257–276, Kluwer Academic Publishers, 2000. B. Bouchon- Meunier and R. R. Yager and L. Zadeh, Eds.
- [10] M. Figueiredo and F. Gomide. Fuzzy system design using neurofuzzy networks. In *Proceedings of the 7th Information Processing and Management of Uncertainty on Knowledge - Based Systems - IPMU'98*, volume 2, pages 1416–1422, 1998.
- [11] W. Pedrycz and F. Gomide. *An Introduction to Fuzzy Sets: Analysis and Design*. MIT Press, Cambridge, MA, 1998.
- [12] R. Yager and D. Filev. *Essentials of Fuzzy Modeling and Control*. Wiley Interscience, New York, 1994.
- [13] R. Ballini. *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, 2000.