

# Redes Neurais Construtivas uma Alternativa para Previsão de Vazões Médias Mensais

Mêuser Valença  
Chesf/UNIVERSO  
Rua Gaspar Peres 427/104  
CEP 50670-350, CDU  
[meuserv@yahoo.com.br](mailto:meuserv@yahoo.com.br)

## Resumo

*O Setor Elétrico Brasileiro tem mais de 90% de sua energia proveniente de usinas hidroelétricas que se encontram distribuídas por 12 bacias hidrográficas do país. O Operador Nacional do Sistema Elétrico (ONS) com o objetivo de otimizar o despacho centralizado das usinas possui dentre outras atribuições a de realizar de forma mensal a previsão de vazões para os locais dos aproveitamentos hidrelétricos. A previsão para o primeiro mês é realizada utilizando-se o modelo PREVIVAZ, que a partir de dados em base semanal, seleciona o melhor modelo. Para os meses entre o segundo e duodécimo à frente, utiliza-se o modelo PREVAZ que trabalha em base mensal, com a metodologia combinada de Box & Jenkins e Yevjehich.*

*O principal objetivo deste artigo é apresentar o modelo NeuroInflow que é baseado em redes neurais construtivas e comparar os resultados obtidos com a metodologia tradicional do modelo PREVAZ atualmente em uso no setor.*

## 1. Introdução

O Setor Elétrico Brasileiro tem mais de 90% de sua energia proveniente de usinas hidroelétricas que se encontram distribuídas por 12 bacias hidrográficas do país. O Operador Nacional do Sistema Elétrico (ONS) com o objetivo de otimizar o despacho centralizado das usinas, possui dentre outras atribuições a de realizar de forma mensal a previsão de vazões para os locais dos aproveitamentos hidrelétricos.

A previsão para o primeiro mês é realizada utilizando-se o modelo PREVIVAZ, que a partir de dados em base semanal, seleciona o melhor modelo dentre as 94 diferentes combinações de séries

temporais, estacionária ou periódica, métodos de estimação de parâmetros e diferentes transformações e consolida os resultados em base mensal.

Para os meses entre o segundo e duodécimo à frente, utiliza-se o modelo PREVAZ que trabalha em base mensal, com a metodologia combinada de Box/Jenkins e Yevjehich [1].

O uso de modelos lineares na predição de vazões e de séries temporais, como os modelos clássicos de Box-Jenkins [1], geralmente está intimamente relacionado à simplicidade desses modelos, o que acarreta um fácil projeto e implementação. Entretanto, existem muitas situações do mundo real nas quais se faz necessário um mapeamento não-linear entre as variáveis de entrada e o domínio de resposta do sistema, daí a necessidade de novas técnicas que permitam este mapeamento de forma eficiente. Com o advento do algoritmo *backpropagation*, uma grande atenção tem-se dado às redes neurais, especialmente às redes MLP (*multi-layer perceptron*), em função da sua capacidade de poder aproximar uma função arbitrária não-linear em muitas variáveis. A capacidade de aproximação delas tem sido provada por vários pesquisadores como Cybenko [2] e Stinchcomb-White [3]. Cybenko [2], mostrou que, uma rede MLP com uma única camada escondida é capaz de aproximar qualquer função contínua definida sobre um conjunto compacto limitado.

Portanto, as redes MLP podem ser consideradas uma técnica extremamente poderosa para realizar um mapeamento não-linear. Entretanto, esses pesquisadores não fornecem procedimentos para determinar o número de neurônios necessários na camada escondida, para realizar a aproximação de uma dada função. Esse aspecto é de fundamental importância, haja vista que se o número de neurônios na camada escondida for muito grande, além de se ter um modelo não-parcimonioso, a rede poderá

memorizar os dados de treinamento e ficar com uma pobre capacidade de generalização. Por outro lado, se o número de neurônios na camada escondida for pequeno, a rede poderá ficar subajustada e, como consequência, não ser capaz de extrair dos dados todas as informações desejadas.

Este artigo apresenta o modelo NeuroInflow, para previsão de vazões mensais afluentes as usinas do Sistema Hidroelétrico Brasileiro e compara os resultados obtidos com os do modelo PREVAZ.

## 2. Redes construtivas

A arquitetura de uma rede neural, de acordo com a possibilidade de mudanças no seu tamanho durante o processo de treinamento, pode ser classificada em estática ou dinâmica. Uma rede é dita de arquitetura estática quando o número de camadas e o número de unidades de processamento permanecem constantes desde a concepção da rede até a finalização do seu treinamento. Por outro lado, uma rede de arquitetura dinâmica tem como característica principal um processo de mudanças no tamanho durante o treinamento.

A proposta do modelo NSRBN (non-linear sigmoidal regression blocks network) é de se construir a rede por blocos de polinômios homogêneos, utilizando uma função de ativação adequada para os neurônios escondidos. Quanto aos neurônios de saída, estes têm uma ativação mais complexa, uma vez que realizam uma regressão logística não-linear com relação à saída dos neurônios escondidos. Desta forma, essas redes NSRBN podem aproximar uniformemente qualquer função contínua definida em um conjunto compacto limitado e também lidar com a realização de funções *booleanas*, o que não é possível com os algoritmos tradicionais do grupo GMDH – Group Method Of Data Handling [4].

O primeiro passo para a construção de um modelo combinatorial consiste em dividir a função  $f(x)$  em blocos compostos de termos de igual grau, como apresentado na Figura 1.

$$f(x) = \tilde{f}_1(x) + \tilde{f}_2(x) + \dots + \tilde{f}_d(x) \quad (1)$$

O bloco de grau  $p$  ( $p=1,2,\dots,d$ ), de acordo com o polinômio de Kolmogorov-Gabor [5], pode ser expresso como uma soma de produtos das entradas, isto é:

$$f_p(x) = \sum_{i_1=1}^N \sum_{i_2=i_1}^N \dots \sum_{i_N=i_1}^N a_p(i_1, i_2, i_3, \dots, i_N) x_1^{qp1(i_1)} \dots x_2^{qp2(i_2)} \dots x_N^{qpN(i_N)} \quad (2)$$

em que,  $q_{p1}(i_1) + q_{p2}(i_2) + q_{p3}(i_3) + \dots + q_{pN}(i_N) = p$  e  $p \geq q_{pm}(\cdot) \geq 0$  para  $m=1,2,3,\dots,N$ .

Tem-se como objetivo realizar o mapeamento de cada um dos polinômios homogêneos de grau  $p$ .

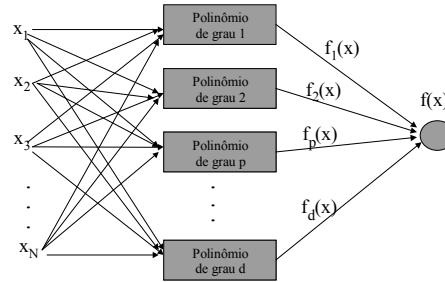


Figura 1 – Arquitetura combinatorial

Uma forma simples seria realizar todos os produtos das variáveis de entrada procurando aproximar diretamente a hypersuperfície não-linear, entretanto isto não é prático, a menos que  $d$  e  $N$  sejam relativamente pequenos (modelo combinatorial). Logo, para se ter um modelo eficiente, faz-se necessário que ele possua funções de ativação que sejam capazes de realizar vários termos dos polinômios homogêneos de forma otimizada, isto é, sem que ocorra um crescimento explosivo no número de parâmetros necessários para ajustar a função desejada. Com esse objetivo, propõe-se uma nova classe de redes neurais que tenha por base o algoritmo combinatorial com uma estrutura de blocos similar a uma rede MLP, sendo que, diferentemente destas, os neurônios de saída realizam a técnica estatística de regressão logística não-linear com relação aos neurônios escondidos [5].

**Definição 1:** uma rede NSRBN é uma rede combinatorial composta pela soma de  $p$  blocos ( $p=1,\dots,d$ ) com estrutura similar a uma MLP, na qual as unidades escondidas desses blocos realizam uma regressão sigmóide linear das entradas (tal qual uma MLP), e as unidades de saída realizam uma regressão do tipo sigmóide não-linear das unidades escondidas (tal qual uma regressão logística não-linear). Essa nova estrutura formada é um aproximador universal com forte capacidade de mapeamento não-linear [5].

A estrutura proposta para cada bloco é composta de apenas uma camada escondida, tendo  $N_h$  unidades escondidas em cada bloco com função de ativação do tipo:

$$f(x) = (\sigma_{net(h)} + \theta_h) \quad (3)$$

em que:  $net(h)$  é a entrada líquida para cada neurônio escondido,  $h$  é um número inteiro que representa as unidades escondidas em cada bloco (variando de 1

até a ordem  $p$  do bloco),  $\theta_h$  é a tendência e  $\sigma_{net(h)}$  é, por exemplo, a função tangente hiperbólica.

O valor de  $p$  correspondente ao grau do bloco é quem vai definir o número de neurônios em cada bloco, isto é: o bloco de grau 1 ( $p=1$ ) tem  $h=1$ , logo um neurônio escondido; o bloco de grau 2 ( $p=2$ ) terá 2 neurônios escondidos ( $h=1$  e  $h=2$ ) e assim sucessivamente, de tal forma que o bloco de grau  $p$  terá  $h=p$  e portanto  $p$  neurônios escondidos.

A estrutura proposta para cada bloco é similar à da Figura 2, na qual apresenta-se o mapeamento para um bloco de grau  $p$  (a ordem dos neurônios escondidos  $h$  variará de 1 até  $p$ ), para  $i$  variáveis de entrada (onde  $i = 1, \dots, N$ ) e uma única unidade de saída.

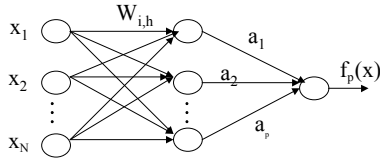


Figura 2 – Arquitetura de um bloco polinomial

Esse bloco é uma rede *feedforward* completamente conectada com uma única camada escondida. A diferença básica desse bloco de grau  $p$  para uma rede MLP é que os neurônios de saída são expressos como uma regressão sigmóide não-linear das unidades escondidas.

Seja  $\mathbf{x} = [1, x_1, x_2, x_3, \dots, x_N]^T$  o vetor coluna das variáveis de entrada, no qual o elemento  $x_i$  denota um elemento qualquer dele. As entradas são ponderadas pelos vetores de pesos  $\mathbf{w} = [w_{0h}, w_{1h}, w_{2h}, \dots, w_{Nh}]^T$ ,  $h=1, \dots, p$ , nos quais  $p$  é a ordem do bloco. Portanto, utilizando-se a função de ativação estabelecida na Equação 3, a saída de cada unidade escondida,  $f(net(h))$ , desse bloco de grau  $p$  pode ser escrito como:

$$f(net(h)) = (\sigma_{net(h)} + \theta_h) \quad (4)$$

em que:  $net(h) = \mathbf{w}_h^T \mathbf{x}$ , ou seja,  $net(h) = \sum_{i=1}^N w_{ih} x_i + w_{0h}$ ,  $h=1, 2, \dots, p$ .

Logo, a unidade de saída (que realiza uma regressão logística não-linear) desse bloco pode ser representada pela seguinte equação:

$$f_p(x) = a_1 (f(net(1)))^1 + a_2 (f(net(2)))^2 + a_3 (f(net(3)))^3 + \dots + a_p (f(net(p)))^p \quad (5)$$

sendo  $\mathbf{a} = [a_1, a_2, a_3, \dots, a_p]^T$  o vetor dos pesos das unidades escondidas para a unidade de saída.

Fazendo-se uso da Equação 3, a função  $f_p(x)$  pode ser escrita da seguinte forma:

$$f_p(x) = \sum_{h=1}^p a_h (\sigma_{net(h)} + \theta_h)^h \quad (6)$$

em que:  $h$  representa a ordem dos neurônios escondidos, ( $h=1, \dots, p$ ), do bloco de grau  $p$ .

De acordo com a Figura 2, a arquitetura de uma rede NSRBN é composta pela soma de blocos com grau variando de 1 a  $d$ . Aqui, está sendo utilizado o bloco  $p$  (Figura 2) para representar qualquer um desses blocos.

Portanto, usando-se esta arquitetura, uma função desconhecida  $f$  em  $R^N$  pode ser aproximada por uma rede NSRBN, através da seguinte expressão:

$$f(x) = \sigma_{net(o)} \left( \sum_{p=1}^d f_p(x) \right) \quad (7)$$

em que:  $\sigma_{net(o)}$  uma função de ativação adequada para a unidade de saída.

Por outro lado, a construção das redes NSRBN é feita de forma incremental, de maneira que se pode representá-la em uma forma similar a de um bloco (Equação 6), conforme apresenta-se a seguir. Considere o passo do algoritmo para o qual se adiciona-se à rede o bloco de ordem  $d$ ,  $f_d$ , tendo-se anteriormente adicionado os blocos  $f_0, f_1, \dots, f_{d-1}$ . Fazendo-se  $f_0 \equiv 0$ , uma função  $f$  pode ser representada por:

$$f(x) = \sigma_{net(o)} \left( \sum_{p=0}^{d-1} f_p(x) + f_d(x) \right) \quad (8)$$

na qual os pesos dos blocos  $f_0, f_1, \dots, f_{d-1}$  são congelados ao adicionar-se o bloco de ordem  $d$ ,  $f_d$ .

Isso posto, pode-se escrever  $f(x)$  da seguinte forma:

$$f(x) = \sigma_{net(o)} \{ c_0 + c_1 (\sigma_{net(i)} + \theta_1) + c_2 (\sigma_{net(2)} + \theta_2)^2 + \dots + c_d (\sigma_{net(d)} + \theta_d)^d \} \quad (9)$$

na qual o vetor  $\mathbf{c} = [c_1, c_2, \dots, c_d]$  representa os pesos das unidades escondidas para as unidades de saída e a constante  $c_0$  está representando todos os blocos congelados até a ordem  $d-1$ .

A Equação 6 é, portanto, idêntica a uma regressão logística não-linear quando a função de ativação utilizada é a sigmóide logística tradicional.

A potencialidade dessas redes está fundamentada nas características impostas pela regressão sigmóide não-linear na unidade de saída.

## 2.1 Algoritmo construtivo para as NSRBN

Esse algoritmo construtivo tem como base um método de aprendizagem construtiva [5]. Portanto,

usando-se essa arquitetura proposta uma função qualquer  $f(x)$  pode ser aproximada pelo uso direto de uma rede NSRBN por dividir  $f(x)$  em blocos homogêneos de ordem 1 até o bloco de ordem  $d$ . Dessa forma, a função total  $f(x)$  (polinômio completo) pode ser aproximada pela soma das funções parciais  $f_p(x)$ ,  $p=1, \dots, d$ , de maneira incremental podendo ser representada matematicamente por:

$$f(x) = g(\hat{f}_1(x) + \dots + \hat{f}_p(x) + \dots + f_d(x)) \quad (10)$$

O modelo proposto tem como princípio manter congelados os pesos de todos os blocos anteriores ao atualmente otimizado, de tal forma que o número de pesos a ser otimizado a cada grau de complexidade não aumente explosivamente [5].

### 3. Modelo Prevaz

O modelo estocástico de previsão de vazões, PREVAVZ, foi desenvolvido na década de 1970, baseia-se numa metodologia combinada de Yevjevich e Box & Jenkins [1].

O modelo PREVAVZ pode utilizar modelos autoregressivos, modelos AR(p), onde o valor atual do processo estocástico se expressa como agregado linear de valores anteriores, ou seja, o próximo valor possui uma forte dependência dos “p” últimos valores ocorridos. Também pode utilizar modelos de médias móveis MA(q) no qual o valor do processo estocástico é linearmente dependente de um número finito de ruídos brancos “q” defasados no tempo. Por último, também pode utilizar um modelo misto autoregressivo e médias móveis ARMA(p,q) que possui as características dos dois anteriores.

O sistema PREVAVZ baseia-se em cinco programas.

O primeiro, denominado “Transfor”, aplica a família de transformações de Box & Cox (logarítmica) aos dados básicos  $X(t)$ , de vazões naturais médias mensais, obtendo uma distribuição normal  $Y(t)$ , que contém a melhor transformação a ser aplicada.

O segundo programa “Comptest”, realiza uma padronização dos dados, a partir de sua média e desvio padrão.

O terceiro programa “Identifi”, fornece subsídios para a identificação dos modelos ARIMA (autoregressivos e médias móveis) a serem ajustados, calculando as funções de autocorrelação e autocorrelação parcial.

O quarto programa “Estima”, através do critério de Akaike estima os coeficientes dos modelos

ARMA(p,q), calcula os respectivos erros padrão e seleciona o melhor deles.

O quinto programa “Prevaz”, a partir do modelo ARMA(p,q) selecionado, realiza uma despadronização e uma transformação inversa para se obter a previsão e seus intervalos de confiança.

### 4. Aplicação

A nossa aplicação está baseada num histórico de vazões médias mensais no período de 1931 a 1998 para todas as usinas do Sistema Interligado Nacional. Para realizar nosso experimento com o horizonte de 1 (um) mês à frente utilizamos estes dados para 36 usinas hidroelétricas onde o período de 1931 a 1990 foi utilizado para treinamento da rede neural e calibração do modelo estatístico e o período de 1991 a 1998 foi utilizado para validação dos modelos.

Como elemento de avaliação do desempenho dos modelos estatísticos e os modelos de rede neural foram utilizados o erro relativo percentual médio absoluto (EPMA), Equação 11, o erro médio absoluto (EMA), Equação 12 e o erro padrão de predição (EP), definido na Equação 13.

$$EPMA = \frac{1}{N} \left[ \sum_{p=1}^N |Z_p - Z_o| / Z_o \right] \cdot 100 \quad (11)$$

$$EMA = \frac{1}{N} \left[ \sum_{p=1}^N |Z_p - Z_o| \right] \quad (12)$$

$$EP = \left[ \frac{1}{N} \sum_{p=1}^N (Z_p - Z_o)^2 \right]^{0,5} \quad (13)$$

Por fim, o coeficiente U de Theil também é utilizado. Ele é definido pela fórmula:

$$U = \frac{\sqrt{(Z_p - Z_o)}}{\sqrt{(Z_{o,t} - Z_{o,t-1})}} \quad (14)$$

em que:  $Z_p$  - é o valor previsto;  $Z_o$  - é o valor ocorrido e  $N$  - é o número de valores de verificação.

#### 4.1 O modelo NEUROINFLOW

A versão 1.0 corresponde ao modelo de previsão de vazões médias mensais com o objetivo de permitir uma análise energética de curto e longo prazo [6]. Este modelo foi utilizado para treinamento de todos os pontos de interesse ao programa mensal de operação, tendo sido dividido em quatro módulos para maior facilidade de uso e manutenção. O NORDESTE (3 usinas), o NORTE (3 usinas), o SUL (15 usinas) e o

SUDESTE (74 usinas). A figura 3 apresenta uma visão da tela inicial deste software.

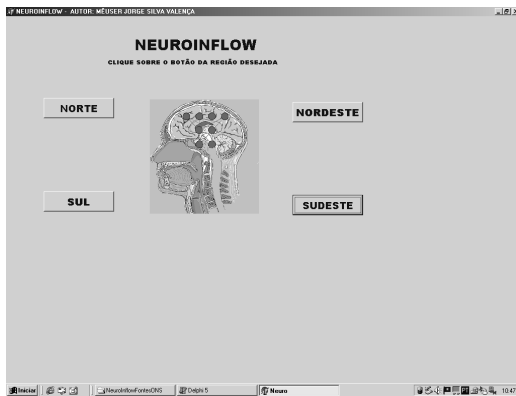


Figura 3 – Tela de abertura do modelo NeuroInflow

#### 4.1.1 Seleção da região

O modelo tem como premissa básica sua facilidade de uso uma vez que ele pode gerar tanto previsões de longo prazo individualizadas como um relatório geral compreendendo todos os pontos de cada região. O acesso aos sub-sistemas do sistema é feito através de um clique sobre o botão que possui o nome da região desejada. Por exemplo: após um clique na região Sudeste aparece à tela da Figura 4.

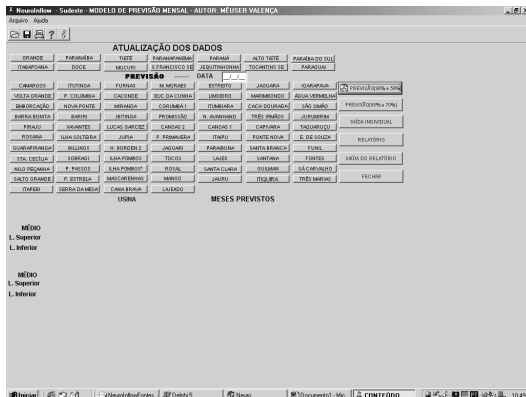


Figura 4 – Tela para a região Sudeste (74 usinas)

#### 4.1.2 Seleção da usina

Selecione o botão que possui o nome da usina e clique sobre este. Imediatamente será apresentada na tela a previsão realizada para 12 meses à frente e também será gerado um arquivo texto para impressão.

#### 4.1.3 Seleção do Intervalo de Confiança

Selecione o botão para qual deseja definir os intervalos de confiança e clique sobre este. Por Exemplo: selecionando o botão Previsão (95% e 70%)

e dando um clique sobre este serão mostrados em tela os novos valores. A tela fica da seguinte forma:

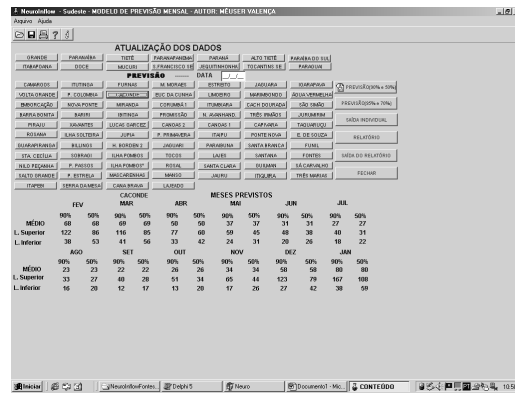


Figura 5 – Tela após seleção da usina para a região Sudeste (Usina de Caconde)

Este software permite a geração de um relatório único com a previsão para todas as usinas de cada sub-sistema com os intervalos de confiança (Figura 6).

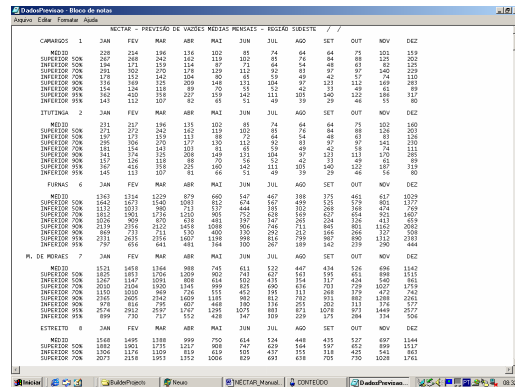


Figura 6 – Tela do arquivo relatório geral para a região Sudeste

Para uma análise comparativa entre os resultados obtidos com as redes neurais construtivas e os modelos estatísticos tradicionais foram selecionadas 36 usinas localizadas em diferentes regiões do país. Analisando-se os resultados obtém-se que o menor coeficiente de Theil ocorreu para o aproveitamento de Tucuruí, 0,36 e o maior a usina de Limoeiro, 0,83.

Entretanto em todos os casos este coeficiente foi inferior a 1, mostrando que as previsões foram melhores que a previsão trivial.

A tabela 1 apresenta para as outras métricas, como exemplo, os resultados para 5 aproveitamentos.

USINA	ESTATÍSTICO			NEUROINFLOW		
	EPMA	EMA	EP	EPMA	EMA	EP
TUCURUI	15,4	2180	4180	13,5	1780	3604
FOZ DO AREIA	47,6	270	461	38,0	240	410
EMBORCAÇÃO	21,7	187	331	16,77	144	241
ITUMBIARA	19,0	450	640	15,0	376	582
ILHA SOLTEIRA	21,0	1333	1560	17,6	1205	1580

Tabela 1– Resultados obtidos – 1 mês à frente

Foi realizado também um estudo comparativo para alguns aproveitamentos no que diz respeito às previsões de longo prazo para o ano de 2000/2001. Para efeito de análise de desempenho dos modelos utilizou-se o erro relativo percentual médio absoluto (EPMA). A tabela 2 apresenta os resultados obtidos para 5 aproveitamentos. Analisando-se a tabela 2 verificam-se os enormes ganhos obtidos na previsão.

APROVEITAMENTO	Out/2000 a Set/2001	
	Prevaz	Rede
TUCURUI	43%	23%
ILHA SOLTEIRA	60%	44,4%
FURNAS	100%	77,5%
SOBRADINHO	71%	55%
FOZ DO AREIA	35%	33%

Tabela 2 – Resultados obtidos – longo prazo

As figuras 7 e 8 apresentam estes resultados, como exemplo, para as usinas de Tucuruí e Sobradinho.

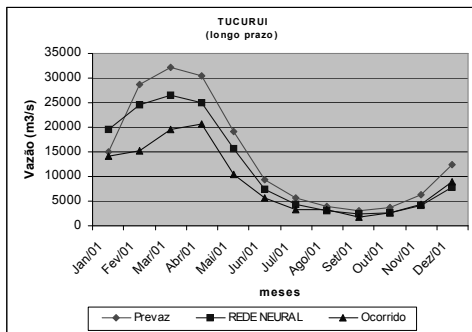


Figura 7 – Comparação entre os resultados para Tucuruí

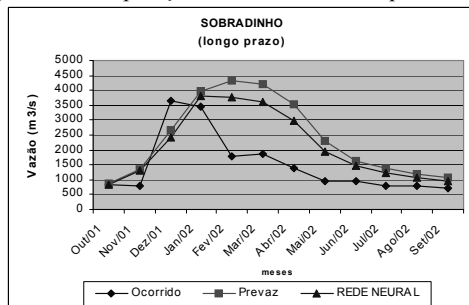


Figura 8 – Comparação entre os resultados para Sobradinho

## 5. Conclusões

No que diz respeito às previsões de vazões mensais, os resultados comprovam a eficácia das redes neurais construtivas, tendo o modelo NeuroInflow considerado aqui, apresentado resultados superiores aos modelos estocásticos para as unidades métricas de erro utilizadas para avaliar o desempenho desses modelos.

No caso de previsões apresentadas neste artigo, uma única rede foi capaz de fornecer melhores resultados do que os diversos melhores modelos de Box-Jenkins, calibrados para cada mês do ano, com um tempo de treinamento inferior a 2 minutos, fornecendo-se apenas a janela de entrada (valores passados de vazões).

Os resultados obtidos com a metodologia proposta além da facilidade de uso, uma vez que o usuário não precisa ser especialista em redes neurais, fornece ganhos substanciais chegando em alguns aproveitamentos a fornecer uma redução de 50% no erro cometido para a previsão de longo prazo. Em média temos uma redução no erro superior a 30% .

## 6. Referências

- [1] G.E.P Box & G.M. Jenkins, Time Series Analysis - Forecasting and Control. Holden-Day Inc, California, 1976.
- [2] G. Cybenko. Approximation by superpositions of a sigmoid function. Mathematics of Control, Signal and Systems, 2:303-314, 1989.
- [3] M. Stinchcomb and H. White. Universal approximation using feedforward networks with non-sigmoid hidden layer activation functions. In proceedings of the International Joint Conference on Neural Networks (IJCNN), 1:613-617, Washington D.C., June 1989.
- [4] M. J. S. Valença and T. B. Ludermitr. Multiplicative-Additive Neural Networks with Active Neurons. International Joint Conference on Neural Networks (IJCNN), IEEE, Book of Summaries (2073), Washington, DC, July 1999.
- [5] M. J. S. Valença. Analysis and Design of the constructive neural networks for complex systems modeling (in portuguese). Ph.D These, UFPE, Brazil, 1999.
- [6] Valença, Mêuser Jorge Silva e Ludermitr, Teresa Bernarda. Multivariate Modelling of Water Resources Time Series Using Constructive Neural Networks. V Brazilian Conference on Neural Networks - V Congresso Brasileiro de Redes Neurais, pages 163-168, PUC, Rio de Janeiro, Abril 2001.