

MODELOS NÃO LINEARES RECORRENTES COM CAPACIDADE DE EXTRAPOLAÇÃO - UM ESTUDO PARA PREDIÇÃO DO INDICE DE PREÇOS AO CONSUMIDOR (IPC-BR)

LIDIO MAURO LIMA DE CAMPOS
limadecampos@gmail.com
UFPA

Resumo: Presents a comparative study for prediction of time series of the Consumer Price Index-CPI using recurrent neural network (RNN). For this, three models are designed for networks with recurrent and are given the changes in

Palavras Chave: Neural Network - time series predicti - - -



MODELOS NÃO LINEARES RECORRENTES COM CAPACIDADE DE EXTRAPOLAÇÃO - UM ESTUDO PARA PREDIÇÃO DO ÍNDICE DE PREÇOS AO CONSUMIDOR (IPC-BR)

Lídio Mauro Lima de Campos

Universidade Federal do Pará em Castanhal

lidio@ufpa.br

Abstract - Presents a comparative study for prediction of time series of the Consumer Price Index-CPI using recurrent neural network (RNN). For this, three models are designed for networks with recurrent and are given the changes in "back propagation" to allow them to incorporate the models ARX (Auto-Regressive with external input) and NARX (Nonlinear Auto Regressive with external input). Furthermore, we present a third architecture, referred with the intermediate layer, nicknamed ARXI, which is a special case of the Elman Network. Is carried out training for all networks and tests the ability to generalize them (identification stage), in order to select the best architectures of recurrent networks to prediction of the IPC. After this stage, it makes the models validation, by means of the test the extrapolation capacity of the networks, i.e., presented data were not used during the training phase and gets the responses that indicate the capacity to predict future CPI for various times (validation phase). We conclude that NARX networks are those with best performance and that the hybrid system proposed by [5] constitutes an excellent tool when you want to get minimal networks that make a series of prediction satisfactorily.

Keywords - Recurrent Neural Networks, Time Series Prediction, NARX networks, hybrid systems.

1 INTRODUÇÃO

Atualmente, observa-se um aumento na demanda por predições e análise de tendência de variáveis em qualquer área de atuação profissional. Muitas empresas e órgãos do governo buscam desenvolver ferramentas computacionais que possibilitem fornecer predições automáticas acerca de valores de variáveis que estão sendo monitorados, o que auxilia na definição de políticas estratégicas, no processo de tomada de decisão e no planejamento a curto e médio prazo dessas organizações. Os métodos para previsão variam muito e dependem da disponibilidade de dados, da qualidade dos modelos disponíveis e dos tipos de suposições feitas, dentre outras coisas. Falando de maneira geral as predições, não são fáceis, dessa forma tem atraído muitos pesquisadores a explorá-las.

As Redes Neurais Artificiais (RNA) tem sido utilizadas de forma crescente para predição de séries temporais em vários ramos de negócio, econômico-financeiro [1][2], predição de séries caóticas [3], consumo de energia[11] dentre outras. As RNAs podem aprender a partir de exemplos, reconhecer um padrão escondido em observações históricas e usá-los para prever valores futuros. Além disso, são capazes de lidar com informação incompleta ou dados ruidosos e podem ser muito eficaz, especialmente em situações onde não é possível definir as regras ou etapas que levam à solução de um problema.

A tarefa de projetar uma arquitetura de uma RNA para simular problemas de predição de séries temporais não é uma tarefa simples, dado que a inclusão de recorrências no modelo aumentam a complexidade dos algoritmos de aprendizado utilizados nessas redes. Dessa forma, as Redes Neurais Recorrentes ainda não foram totalmente exploradas devido ao seu tedioso treinamento e por suas estruturas complexas, [4] discute que vários enfoques buscam métodos heurísticos para limitar a complexidade do projeto, tempo de computação, a parametrização e a seleção de RNAs, o que causa problemas de decisão adicionais e uma abordagem de tentativa e erro para a modelagem da rede.

Alguns pesquisadores estão buscando métodos de projeto automático de RNAs usando estratégias evolucionárias, entretanto ainda são poucos os trabalhos que buscam a geração de arquiteturas de Redes Neurais Artificiais Recorrentes (RNARs). [6] Propôs um método que usa Algoritmos Evolucionários para projeto automático de arquiteturas de RNAs, o artigo apresenta uma avaliação de oito combinações de estratégias evolucionárias em 16 conjunto de dados de domínio público, a metodologia apresenta excelentes resultados para Redes Neurais Diretas. Os métodos de construção automática de (RNARs) tem sido prejudicadas pelo fato de que os algoritmos de treinamento dessas redes são mais complexos do que os utilizados para Redes Diretas.

[5] Propõe uma metodologia plausível biologicamente que é capaz de gerar Redes Neurais Artificiais Recorrentes com o número ideal de neurônios e topologia adequada. Com esse objetivo, três metáforas biológicas foram utilizadas: Algoritmos Genéticos (AG), Sistemas de Lindenmayer (L-Systems) e RNAs. A metodologia tenta imitar o processo natural de crescimento e evolução do sistema nervoso, usando os L-Systems como uma receita para o desenvolvimento dos neurônios e suas conexões o AG para evoluir e otimizar a arquitetura do sistema nervoso adequado para uma específica tarefa. A técnica foi testada em três problemas simples conhecidos (XOR, Paridade e Linguagens de Tomita[9]) onde topologias de redes recorrentes devem ser utilizadas. Um problema mais complexo, envolvendo o aprendizado de séries temporais foi também



estudado. Os resultados mostram que a proposta é bastante promissora e podendo gerar arquiteturas de redes neurais diretas e recorrentes com um número ideal de neurônios e conexões, boa capacidade de generalização, menor erro e tolerância ao ruído grande.

Atualmente buscam-se melhorias para a metodologia proposta em [5], por meio de um aprofundamento no aspecto teórico-metodológico com três focos: primeiro o estudo de mecanismos que possam imitar o processo de desenvolvimento biológico dos neurônios, considerando aspectos de plausibilidade biológica no que diz respeito a codificação cromossômica [7], segundo estratégias evolucionárias que possam selecionar e evoluir as melhores arquiteturas de conexão das Redes Neurais Artificiais geradas, bem como a possibilidade de explorar (RNARs) com arquiteturas variadas e nesse aspecto um estudo detalhado de várias possibilidades de conexões recorrentes é necessário, dada a complexidade dos algoritmos de treinamento para essas arquiteturas.

Para que os objetivos da pesquisa sejam alcançados, na seção 2 apresentam-se três arquiteturas utilizadas na predição do Índice de Preços ao Consumidor: a rede ARX (Auto-Regressiva com entrada externas) a NARX (Auto Regressiva não linear com entrada externas). Além disso, considera-se uma terceira arquitetura, com a camada intermediária realimentada, apelidada de ARXI. Na seção 3, apresentam-se os resultados de simulação para predição de uma série temporal que prevê o Índice de Preços ao Consumidor –IPC e finalmente as conclusões na seção 4 e na seção 5 as referências bibliográficas consultadas.

2- REPRESENTAÇÕES NÃO LINEARES

A predição de séries temporais vem sendo realizada com o uso dos tradicionais modelos Auto Regressivos e/ou médias móveis (AR e ARMA). Trata-se de modelos paramétricos através dos quais a previsão das futuras observações é obtida a partir da combinação linear (linear combination) de valores passados e, quando for o caso, com as componentes de ruído da série de interesse ponderadas por um conjunto de parâmetros. Os sistemas dinâmicos (dynamics systems) são não lineares, dessa forma nessas aplicações devem-se escolher modelos com não linearidade pelo fato de produzirem certos regimes dinâmicos que os lineares não conseguem representar.

As RNAs são um exemplo bem difundido de representação linear que atualmente são cada vez mais usadas para predição de séries temporais, devido à sua capacidade de agregar conhecimento em sua estrutura, a partir de exemplos, um preditor baseado em rede neural tem condições de estimar o comportamento futuro de uma série temporal apenas a partir de suas amostras passadas. Um modelo de predição deste tipo é chamado de não paramétrico, uma vez que não há necessidade de se conhecer os parâmetros do processo que gera o sinal. O modelo do processo é estimado através de um algoritmo de aprendizagem onde os exemplos são apresentados à rede neural e seus pesos são atualizados de acordo com o erro de predição. A seguir apresentam-se algumas representações não lineares baseadas em RNAs, bem como as alterações no algoritmo “backpropagation” para cada caso.

2.1 – REPRESENTAÇÃO 1 : REDE RECORRENTE COM SAÍDA REALIMENTADA

Na presente seção apresenta-se uma arquitetura de Rede Neural Recorrente com a saída realimentada, Figura 1. Essa RNAR foi baseada no modelo ARX (AGUIRRE,2007), que nada mais é do que uma rede MLP cuja entrada consiste da própria saída realimentada com atrasos. Esta rede neural é equivalente ao Modelo **ARX (Autoregressive with exogenous inputs)**, dado pela equação (1). Onde $x(n)$ é a entrada do sistema e $y(n)$ a saída, em que a função $f(\cdot)$ é uma função não linear, geralmente desconhecida e $x(n)$ e $y(n)$ correspondem à entrada e saída no tempo n , enquanto $dy > 0$, é a ordens da memória de entrada. Quando esta função é aproximada por uma rede “perceptron” de múltiplas camadas, a topologia resultante é chamada rede recorrente **ARX**, sendo um caso particular da rede apresentada na seção 2.3.

$$y(n) = f[y(n-1) + a_2y(n-dy) + x(n)] \quad (1)$$

Para as deduções dos modelos apresentados nessa seção e nas posteriores, considere que A seja o número de unidades da camada de entrada, conforme determinado pelo comprimento dos vetores de entrada de treinamento, C o número de unidades da camada de saída e B o número de unidades da camada oculta. As camadas de entrada e oculta, têm, cada uma, uma unidade extra usada como limite; portanto as unidades dessas camadas às vezes serão indexadas pelos intervalos $(0, \dots, A)$ e $(0, \dots, B)$. Denotam-se os níveis de ativação das unidades da camada de entrada por x_j da camada oculta por h_j e da camada de saída por o_j . Os pesos que conectam a camada de entrada a camada oculta são denotados por w_{1ij} , onde i indexa as unidades de entrada e j , as unidades ocultas. Da mesma forma, os pesos que conectam a camada oculta à camada de saída são denotados por w_{2ij} com i indexando as unidades ocultas e j as unidades de saída.

As modificações realizadas no “backpropagation” para aproximação do modelo ARX pelo modelo neural recorrente foram as seguintes: as saídas na camada intermediária são agora dadas pela equação (2). Observa-se a inclusão das contribuições das recorrências dadas pelo somatório de $m=0$ a C para os termos $om(t-1)$ na mesma.



$$h_k(t) = \frac{1}{1 + e^{-\sum_{m=0}^C o_m(t-1) \cdot w_3(t)_{mk} + \sum_{j=0}^A X_{nj}(t) \cdot w_1(t)_{jk}}}, \quad (2)$$

para $t > 0$, $h_0 = 1$, que é o valor do bias. O termo $o_m(t-1)$ refere-se a cada saída que é realimentada e C é o número de neurônios da camada de saída, para $t=0$ o seu valor é $o_m(t-1) = 0$. A atualização dos pesos das realimentações são dadas pela equação 3 :

$$\Delta w_{3_{mk}}(t+1) = \Delta w_{3_{mk}}(t) - \eta \frac{\partial \varepsilon_n}{\partial w_{3_{mk}}(t)} \quad (3)$$

Após deduções matemática chegou-se a conclusão que as atualizações dos pesos da camada intermediária e da camada de saída são dadas pelas equações (4) e (5). O desenvolvimento da equação (3) é dado pelas equações (6) a (21).

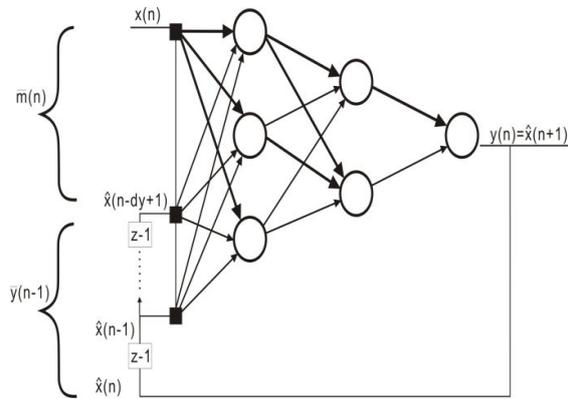


Figura 1 – Rede ARX aproximação do Modelo ARX (1)

$\Delta w_{2_{km}}(t+1) = \Delta w_{2_{km}}(t) + \eta [(y_{nm} - o_m)(1 - o_m) \cdot o_m] h_k \quad (4)$	$\Delta w_{1_{jk}}(t+1) = \Delta w_{1_{jk}}(t) + \eta [(1 - h_k) h_k \cdot \sum_{m=0}^C (y_{nm} - o_m)(1 - o_m) \cdot o_m \cdot w_{2_{km}}] x_{nj} \quad (5)$
$\frac{\partial \varepsilon_n}{\partial w_{3_{mk}}(t)} = \frac{1}{2} \sum_{m=0}^C \frac{\partial \varepsilon_{nm}}{\partial o_m} \frac{\partial o_m}{\partial h_k} \frac{\partial h_k}{\partial w_{3_{mk}}} \quad (6)$	$\frac{\partial \varepsilon_{nm}}{\partial o_m} = \frac{\partial \varepsilon_{nm}}{\partial o_m} = -2(y_{nm}(t) - o_m(t)) \quad (7)$
$\frac{\partial o_m}{\partial h_k} = (1 - o_m) \cdot o_m \cdot w_{2_{km}} \quad (8)$	$\frac{\partial h_k}{\partial w_{3_{mk}}} = \frac{\partial}{\partial w_{3_{mk}}} \left[\frac{1}{1 + e^{-\sum_{m=0}^C o_m(t-1) \cdot w_3(t)_{mk} + \sum_{j=0}^A X_{nj}(t) \cdot w_1(t)_{jk}}} \right] \quad (9)$
$u_k(t) = 1 + e^{-\sum_{m=0}^C o_m(t-1) \cdot w_3(t)_{mk} + \sum_{j=0}^A X_{nj}(t) \cdot w_1(t)_{jk}} \quad (10)$	$u_k(t) - 1 = e^{-\sum_{m=0}^C o_m(t-1) \cdot w_3(t)_{mk} + \sum_{j=0}^A X_{nj}(t) \cdot w_1(t)_{jk}} \quad (11)$
$h_k(t) = \frac{1}{u_k(t)}, \quad \frac{\partial h_k(t)}{\partial u_k(t)} = -\frac{1}{u_k(t)^2}, \quad (12), (13)$	$\frac{\partial u_k(t)}{\partial w_{3_{mk}}(t)} = \frac{\partial h_k(t)}{\partial u_k(t)} \frac{\partial u_k(t)}{\partial w_{3_{mk}}(t)} \quad (14)$
$\frac{\partial u_k(t)}{\partial w_{3_{mk}}(t)} = (u_k(t) - 1)(-o_m(t-1)) \quad (15)$	$\frac{\partial h_k(t)}{\partial w_{3_{mk}}(t)} = -\frac{1}{u_k(t)^2} (u_k(t) - 1)(o_m(t-1)) \quad (16)$



$\frac{\partial h_k(t)}{\partial w_{pk}(t)} = \frac{\left(\frac{1}{h_k(t)} - 1\right)}{\left(\frac{1}{h_k(t)}\right)^2} \cdot om(t-1) \quad (17)$	$\frac{\partial h_k(t)}{\partial w_{mk}(t)} = (1 - h_k(t)) h_k(t) \quad (18)$
$\frac{\partial \varepsilon_n}{\partial w_{mk}(t)} = (1 - h_k(t)) h_k(t) * P \quad (19)$ <p>Sendo P dado pela equação (20)</p>	$P = \sum_{m=0}^C (o_m(t) - y_{nm}(t))(1 - o_m(t)) \cdot o_m(t) \cdot om(t-1) \cdot w_{km}(t) \quad (20)$
<p>Logo a equação 3 será dada por:</p> $\Delta w_{mk}(t+1) = \Delta w_{mk}(t) + \eta (1 - h_k(t)) h_k(t) \sum_{n=0}^C (y_{nm}(t) - o_m(t))(1 - o_m(t)) \cdot o_m(t) \cdot om(t-1) w_{km}(t) \quad (21)$	

2.2- REPRESENTAÇÃO 2 : REDE RECORRENTE COM CAMADA INTERMEDIÁRIA REALIMENTADA

O que se pretende na presente seção é apresentar a formulação de um algoritmo baseado na regra delta generalizada para redes recorrentes com a camada intermediária realimentada, a arquitetura da rede é mostrada na Figura 2. Essa rede é um caso particular da Rede de Elman, no sentido de que entrada exógena, não apresenta atrasos $x(n)$, mas apresenta a unidade de contexto realimentada com atrasos $vq(n-1)$. As saídas na camada intermediária são agora dadas pela equação (22).

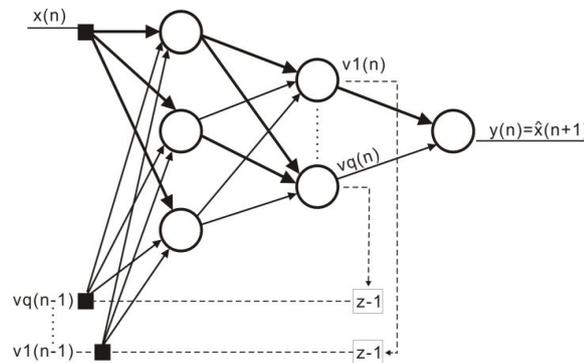


Figura 2

$$hk(t) = \frac{1}{1 + e^{-\sum_{p=1}^B hp(t-1) \cdot w3(t)_{pk} + \sum_{j=0}^A X(t) \cdot w1(t)_{jk}}} \quad (22)$$

para $t > 0$, $h(0) = 1$, que é o valor do bias. Os termos $hp(t-1)$ são as realimentações iguais a $hk(t-1)$, para $t=0$ o seu valor é $hp(t-1) = 0$. A atualização dos pesos é dada por:

$$\Delta w_{pk}(t+1) = \Delta w_{pk}(t) - \eta \frac{\partial \varepsilon_n}{\partial w3(t)_{pk}} \quad (23)$$

a atualização dos pesos da camada intermediária e da camada de saída são dadas pelas equações (4) e (5). A atualização dos pesos da realimentação é calculado pela equação (24). O desenvolvimento da equação 23 é mostrado pelas equações (24) a (38).

2.3 – REPRESENTAÇÃO 3 – REDE NARX

O modelo estudado nessa seção é o modelo NARX que nada mais é que uma rede MLP cuja entrada consiste na própria saída realimentada com atrasos de tempo e uma entrada exógena, também com atrasos. Nessa arquitetura, as saídas estimadas da rede são introduzidas novamente às entradas, essa representação é equivalente ao modelo estatístico **NARX (Nonlinear Autoregressive model with exogenous input)**, dado por :



$$y(n) = f[y(n-1) \dots y(n-dy), u(n-1) \dots u(n-du+1)] \quad (39)$$

Em que a função $f(\cdot)$ é uma função não linear, geralmente desconhecida e $u(n)$ e $y(n)$ correspondem à entrada e saída no tempo n , enquanto $du > 0$ e $dy > 0$, $du \leq dy$ são as ordens da memória de entrada e memória de saída. Quando esta função é aproximada por uma rede perceptron de múltiplas camadas, a topologia resultante é chamada rede recorrente NARX. A Figura 3 ilustra uma rede NARX, com uma camada oculta e um laço de realimentação global. Na presente pesquisa considerou-se a utilização da rede NARX no modo de identificação paralelo.

$\frac{\partial \varepsilon_n}{\partial w_{3_{pk}}(t)} = \frac{1}{2} \sum_{m=0}^C \frac{\partial \varepsilon_{nm}}{\partial o_m} \frac{\partial o_m}{\partial h_k} \frac{\partial h_k}{\partial w_{3_{pk}}} \quad (24)$	$\frac{\partial \varepsilon_{nm}}{\partial o_m} = \frac{\partial \varepsilon_{nm}}{\partial o_m} = -2(y_{nm}(t) - o_m(t)) \quad (25)$
$\frac{\partial o_m}{\partial h_k} = (1 - o_m) \cdot o_m \cdot w_{2_{km}} \quad (26)$	$\frac{\partial h_k}{\partial w_{3_{pk}}} = \frac{\partial}{\partial w_{3_{pk}}} \left[\frac{1}{1 + e^{-\sum_{p=1}^B hp(t-1) \cdot w_{3(t)_{pk}} + \sum_{j=0}^A X(t) \cdot w_{1(t)_{jk}}}} \right] \quad (27)$
$u_k(t) = 1 + e^{-\sum_{p=1}^B hp(t-1) \cdot w_{3(t)_{pk}} + \sum_{j=0}^A X(t) \cdot w_{1(t)_{jk}}} \quad (28)$	$u_k(t) - 1 = e^{-\sum_{p=1}^B hp(t-1) \cdot w_{3(t)_{pk}} + \sum_{j=0}^A X(t) \cdot w_{1(t)_{jk}}} \quad (29)$
$h_k(t) = \frac{1}{u_k(t)}, \frac{\partial h_k(t)}{\partial u_k(t)} = -\frac{1}{u_k(t)^2} \quad (30), (31)$	$\frac{\partial u_k(t)}{\partial w_{3(t)_{pk}}} = \frac{\partial h_k(t)}{\partial u_k(t)} \frac{\partial u_k(t)}{\partial w_{3(t)_{pk}}} \quad (32)$
$\frac{\partial u_k(t)}{\partial w_{3(t)_{pk}}} = (u_k(t) - 1)(-hp(t-1)) \quad (33)$	$\frac{\partial h_k(t)}{\partial w_{3_{pk}}(t)} = -\frac{1}{u_k(t)^2} (u_k(t) - 1)(hp(t-1)) \quad (34)$
$\frac{\partial h_k(t)}{\partial w_{3_{pk}}(t)} = \frac{\left(\frac{1}{h_k(t)} - 1\right)}{\left(\frac{1}{h_k(t)}\right)^2} \cdot hp(t-1) \quad (35)$	$\frac{\partial h_k(t)}{\partial w_{3_{pk}}(t)} = (1 - h_k(t)) h_k(t) \cdot hp(t-1) \quad (36)$
$\frac{\partial \varepsilon_n}{\partial w_{3_{pk}}(t)} = (1 - h_k(t)) h_k(t) \cdot hp(t-1) \sum_{m=0}^C (o_m(t) - y_{nm}(t))(1 - o_m(t)) \cdot o_m(t) \cdot w_{2_{km}}(t) \quad (37)$	
<p>Logo: $\Delta w_{3_{pk}}(t+1) = \Delta w_{3_{pk}}(t) + \eta(1 - h_k(t)) h_k(t) \cdot hp(t-1) \sum_{m=0}^C (y_{nm}(t) - o_m(t))(1 - o_m(t)) \cdot o_m(t) \cdot w_{2_{km}}(t) \quad (38)$</p>	

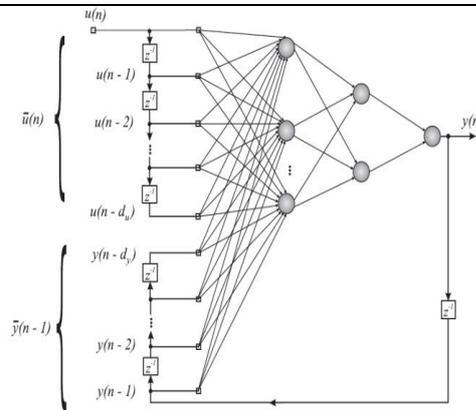


Figura 3 – rede NARX (Fonte [3]) com *du* entradas e *dy* atrasos na saída.

As modificações realizadas no “backpropagation” original foram as seguintes: as saídas na camada intermediária são agora dadas pela equação (40). Observa-se a inclusão das contribuições das recorrências dadas pelo somatório de $m=0$ a C para os termos $om(t-1)$ que correspondem a própria saída realimentada com atrasos e uma entrada com exógena com atrasos, representados pelos termos $Xnj(t)+Xnj(t-1)+\dots+Xnj(t-du)$ na equação (40).

$$hk(t) = \frac{1}{1 + e^{-\sum_{m=0}^C om(t-1) \cdot w3(t)_{mk} + \sum_{j=0}^A [Xnj(t) + Xnj(t-1) + \dots + Xnj(t-du)] w1(t)_{jk}}}, \quad (40)$$

Considerando-se a modificação da função $hk(t)$ para a rede NARX, após deduções matemáticas chegou-se a conclusão que a atualização dos pesos da camada de saída é dada pela equação (4), a atualização dos pesos da realimentação é calculado pela equação (41) e finalmente a atualização dos pesos da camada intermediária é dada pela equação (21).

$$\Delta w1_{jk}(t+1) = \Delta w1_{jk}(t) + \eta [(1 - h_k) h_k \cdot \sum_{m=0}^C (y_{nm} - o_m) (1 - o_m) \cdot o_m \cdot w2_{km} \{x_{nj}(t) + x_{nj}(t-1) + \dots + x_{nj}(t-du)\}] \quad (41)$$

3 – EXPERIMENTOS

Os experimentos realizados, para testes dos modelos apresentados na secção 2, foram baseados na série temporal que representa o índice de preços a consumidor – ICP no período de janeiro de 1988 a dezembro de 2009 (Fonte: Banco Central do Brasil, os dados são mostrados na Tabela 1. O ICP quantifica o custo de produtos em diferentes momentos, em outras palavras são medidas do nível de preços de bens e serviços adquiridos pelas famílias através do tempo, sendo útil para o cálculo da inflação.

	JAN	FEB	MAR	APR	MAI	JUN
1998	0,0126	0,0014	0,0033	0,0023	0,0014	0,0041
1999	0,0064	0,0141	0,0095	0,0052	0,0008	0,0065
2000	0,0101	0,0005	0,0051	0,0025	0,004	-0,0001
2001	0,0064	0,004	0,0056	0,0086	0,0041	0,0052
2002	0,0079	0,0014	0,0042	0,0071	0,0028	0,0055
2003	0,0232	0,0137	0,0106	0,0112	0,0069	-0,0016
2004	0,0108	0,0028	0,0046	0,0031	0,0071	0,0078
2005	0,0085	0,0043	0,007	0,0088	0,0079	-0,0005
2006	0,0065	0,0001	0,0022	0,0034	-0,0019	-0,004
2007	0,0069	0,0034	0,0048	0,0031	0,0025	0,0042
2008	0,0097	0,0056	0,0045	0,0072	0,0087	0,0077
2009	0,0083	0,0021	0,0061	0,0047	0,0039	0,0012
2010	0,0129	0,0068	0,0086	0,0076	0,0021	
	JUL	AUG	SEP	OCT	NOV	DEC
1998	-0,0025	-0,0052	-0,0017	0,002	-0,0019	0,0009
1999	0,012	0,0048	0,0019	0,0092	0,0112	0,006
2000	0,0191	0,0086	0,0004	0,0002	0,004	0,0062
2001	0,0136	0,0054	0,0012	0,0071	0,0085	0,007
2002	0,0103	0,0076	0,0066	0,0114	0,0314	0,0194
2003	0,0034	0,0013	0,0076	0,0021	0,0033	0,0043
2004	0,0059	0,0079	0,0001	0,001	0,0037	0,0063
2005	0,0013	-0,0044	0,0009	0,0042	0,0057	0,0046
2006	0,0006	0,0016	0,0019	0,0014	0,0024	0,0063
2007	0,0028	0,0042	0,0023	0,0013	0,0027	0,007
2008	0,0053	0,0014	-0,0009	0,0047	0,0056	0,0052
2009	0,0034	0,002	0,0018	0,0001	0,0026	0,0024

Tabela 1 – IPC período jan-1998 a dezembro de 2009 valores normalizados.

Inicialmente realizou-se o estudo da capacidade de Generalização das arquiteturas propostas na secção 2 e também da rede MLP. Para isso selecionou-se parte dos dados da Tabela 1 para generalização (identificação) e parte para testes da capacidade extrapolação (extrapolation capacity) das redes (validação). Dessa forma escolheram-se valores do IPC no período



de jan-1998 a dez-2002 para que fosse realizado o treinamento das RNAs e análise da capacidade de generalização das mesmas, o que corresponde a fase de identificação do sistema. Os parâmetros utilizados nos algoritmos de treinamento foram: um (1) neurônio na camada de entrada, quatro (4) na intermediária e um (1) na de saída, taxa de aprendizagem 1.98 e 240000 épocas. O Gráfico 1 mostra esses resultados, percebe-se que o melhor desempenho na generalização foi obtido pela rede NARX e segundo pela rede ARX que é um caso particular da primeira, a MLP consegue generalizar com qualidade inferior as demais. A rede NARX apresentou um bom desempenho e convergência rápida e generalização melhor que as demais, sendo que a rede ARX apresenta desempenho aproximado. Isto ocorre porque o vetor de entrada dos modelos NARX são construídos por meio de uma linha de atrasos com derivação deslizadas sobre o sinal de entrada, junto com uma linha de atraso com derivação formada pelas realimentações do sinal de saída da rede [12].

Para a validação dos modelos testou-se a rede NARX, para diversos parâmetros, inicialmente simulou-se uma rede com o seguinte número de neurônios um (1) na camada de entrada, seis (6) na camada intermediária e um (1) na de saída, taxa de aprendizagem 1.98 e 240000 épocas. O Gráfico 2 ilustra os resultados considerando 3 passos a frente, ou seja na validação considerou-se dados dos anos 2003, 2004 e 2005 que não foram utilizados na fase de identificação (generalização). Considerou-se que esses resultados foram satisfatórios para um erro de 0.0001. Entretanto, mudaram-se alguns parâmetros do treinamento, foram realizados vários experimentos com objetivo de se obter uma melhor validação do modelo NARX. Isolou-se uma das melhores soluções que foi obtida considerando-se os seguintes parâmetros : número de neurônios um (1) na camada de entrada, oito (8) na camada intermediária e um(1) na de saída, taxa de aprendizagem 30.98 e 240000 épocas. O Gráfico 3 ilustra os resultados, percebe-se que à medida que se aumenta o número de neurônios da camada intermediária de 6 para 8 o desempenho dessa rede melhora consideravelmente, ou seja o modelo NARX apresentado na secção 2.3, funciona como um excelente preditor (linha vermelha). O Gráfico 4 mostra os testes de validação para a rede ARX, com os mesmos parâmetros da Rede NARX apresentada anteriormente, ou seja com 8 (oito) neurônios na camada intermediária, percebe-se um bom desempenho da Rede, mas a rede NARX ainda assim é superior. Com objetivo de encontrar um rede mínima capaz se predizer o IPC, utilizou-se o sistema híbrido proposto por [5][8], dessa forma um universo maior de redes foram testadas e o algoritmo evolucionário, direcionado pela função aptidão ("fitness"), busca agora as menores redes capazes de predizer bem o IPC, o Gráfico 5 mostra os resultados obtidos por uma rede com um(1) neurônio na camada de entrada, dois (2) na camada intermediária e um (1) na de saída, taxa de aprendizagem 1.98 e 240000 épocas.

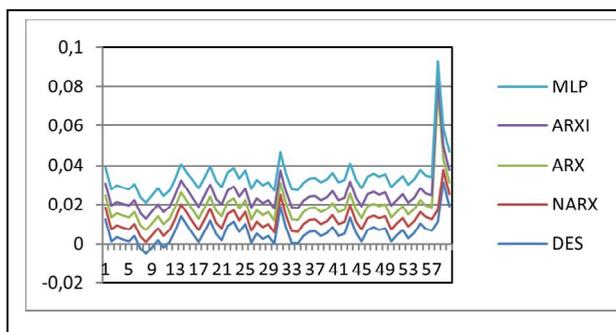


Gráfico 1 - Fase de Identificação – Capacidade de Generalização das Redes

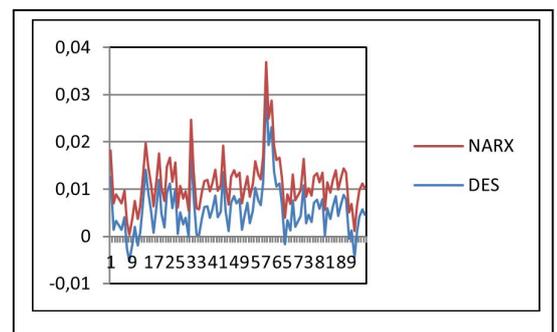


Gráfico 2 - Validação do Modelo Rede NARX N= 3 PASSOS.

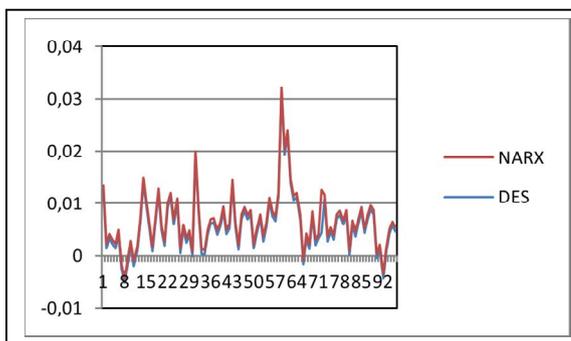


Gráfico 3 - Validação do Modelo Rede NARX N= 3 PASSOS.

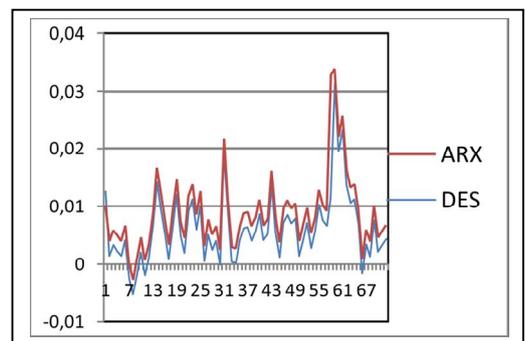


Gráfico 4 - Validação do Modelo Rede ARX N= 1 PASSO.

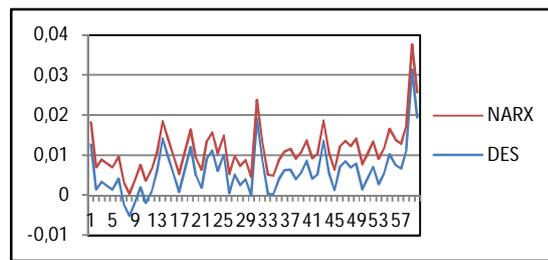


Gráfico 5 - Generalização Modelo Rede NARX/Sistema Híbrido.

4 – CONCLUSÕES

Após os experimentos, chegou-se a conclusão que a Rede NARX, além de apresentar bom desempenho para o problema de predição do IPC, possuiu convergência mais rápida e capacidade de generalização melhor que as outras redes. Isto ocorre porque o vetor de entrada dos modelos NARX são construídos por meio de uma linha de atraso com derivação deslizada sobre o sinal de entrada, junto com uma linha de atraso com derivação formada pelas realimentações do sinal de saída da rede [12]. A rede ARX apresenta um desempenho similar a NARX, para o problema estudado, por outro lado a rede apresentada na secção 2.2, não mostrou uma boa capacidade de generalização nas simulações tendo sido descartada dos estudos de predições a passos futuros, a rede MLP não generalizou bem também para o problema em estudo. O Sistema híbrido [5] é uma excelente opção quando se deseja testar uma população de redes que conseguem generalizar bem o problema, sem a intervenção do projetista, além disso é possível direcionar a busca, como foi o caso da simulação mostrada no Gráfico 5, onde buscou-se uma rede mínima capaz de generalizar bem, o que as vezes é necessário. A Função aptidão utilizada no sistema híbrido proposto por [5] direciona a busca considera um número mínimo de : neurônios na camada intermediária, regras de produção e conexões recorrentes e erro residual o que nem sempre é desejável.

5 - Referencias

- [1]Lam M., Neural network techniques for financial performance prediction : integrating fundamental and technical analysis. Decision Support Systems 37, 2004; 567-581.
- [2]Yao J.T., Towards a better forecasting model for economic indices. In: Proceedings of the 6th Joint Conference on Information Science, 2002; 299-303.
- [3]José M. Menezes Jr. & Guilherme A. Barreto (2006), A New Look at Nonlinear Time Series Prediction with NARX Recurrent Neural Network, IX Brazilian Neural Networks Symposium (SBRN'2006), Ribeirão Preto-SP.
- [4]S. Crone (2005) Stepwise Selection of Artificial Neural Network Models for Time Series Prediction, Journal of Intelligent Systems, Vol. 14, No. 2-3, 2005, pp. 99-122
- [5]De Campos L.M.L, R. M., A Biologically inspired methodology for neural networks design In: IEEE Conference on Cybernetics and Intelligent Systems. IEEE, pp. 619-624, 2004.
- [6]Cant-Paz, E. and C. Kamath, An empirical comparison of combinations of evolutionary algorithms and neural networks for classification problems, IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics, pages 915-927, 2005.
- [7]N. Feng, G. Ning, and X. Zheng, A Framework for simulating axon guidance, In: Proceedings of Neurocomputing, 2005, pp.70-84.
- [8]De Campos, Lídio Mauro Lima ; Roisenberg, Mauro ; Oliveira, Roberto Célio Limão de . (Aceito) Automatic Design of Neural Networks with L-Systems and Genetic Algorithms - A Biologically Inspired Methodology. IJCNN 2011 - IEEE International Joint Conference on Neural Networks – San Jose, California, USA, 2011.
- [9]M.Tomita., Dynamic construction of Finite automata from examples using hill-climbing. In: Proceedings of the Fourth Annual Conference of the Cognitive Science Society. Ann Arbor, MI, pp. 105-108, 1982.
- [10]Aguirre, L.A., Introdução à Identificação de Sistemas, Editora UFMG, terceira edição, 2007.
- [11]PARK, D. C. et. al. Electrical load forecasting using an artificial neural network. Artificial Neural Network: forecasting time series. E.U.A., IEEE Press, p. 43-49, 1994.
- [12]LIN, T. et al. Learning long-term dependencies in NARX recurrent neural networks. IEEE Transactions on Neural Networks, v. 7, n. 6, p. 1424_1438, 1996.