

PROJETO AUTOMÁTICO DE REDES NEURAIIS ARTIFICIAIS PARA O PROBLEMA DE PREVISÃO EM SÉRIES TEMPORAIS

Gustavo Sikora de Melo

Universidade Estadual do Ceará
Av. Paranjana, 1700 - Campus do Itaperi - Fortaleza/CE
gustavo.sikora@gmail.com

Gustavo Augusto Lima de Campos

Universidade Estadual do Ceará
Av. Paranjana, 1700 - Campus do Itaperi - Fortaleza/CE
gustavo@larces.uece.br

Jackson Savio de Vasconcelos Silva

Universidade Estadual do Ceará
Av. Paranjana, 1700 - Campus do Itaperi - Fortaleza/CE
jackson@larces.uece.br

Jefferson Teixeira de Sousa

Universidade Estadual do Ceará
Av. Paranjana, 1700 - Campus do Itaperi - Fortaleza/CE
jeff@larces.uece.br

RESUMO

Redes neurais artificiais MLP *backpropagation* podem ser empregadas, de diversas maneiras, para a previsão baseada em séries temporais. No entanto, a busca pela melhor arquitetura que produza bons resultados ainda é feita muitas vezes de modo intuitivo, por tentativa e erro. Grande parte das abordagens que visam automatizar o projeto da arquitetura da rede neural, não altera a quantidade de neurônios nas camadas e sim a topologia da rede. Este trabalho apresenta uma abordagem para o projeto automático de redes neurais artificiais para previsão utilizando algoritmos genéticos para projetar uma boa arquitetura, ou seja, capaz de resolver o problema de forma satisfatória.

PALAVRAS CHAVE. Redes Neurais Artificiais. Previsão. Séries Temporais. Outras aplicações ou outras metodologias.

ABSTRACT

MLP artificial neural network can be applied in many ways for the time series prediction, yet the search for good architectures that produces good results is done mainly by the trial and error method. Most of the approaches that deals with neural network architectures project automation, deals with the network connection topology instead of changing the numbers of neurons in each layer. This work presents an approach for predicting using genetic algorithms to project good network architectures, which means, that solves the problem in a satisfactory way.

KEYWORDS. Artificial Neural Network. Prediction. Time Series. Other applications or methodologies.

1. Introdução

Um problema muito interessante e relevante para muitas, senão todas as ciências, é o problema de previsão. O estudo de séries temporais desperta a curiosidade sobre o estado futuro de uma situação, seja ela uma série financeira do mercado de ações ou mesmo a previsão meteorológica. Visto desta forma, a análise de séries temporais é relevante para diversas áreas do conhecimento, uma vez que modelos matemáticos de vários problemas são desconhecidos, incompletos ou insatisfatórios.

Muitos problemas de previsão com base em séries temporais são abordados com modelos estatístico-matemáticos lineares que oferecem rapidez, porém aplicações limitadas. Isto ocorre uma vez que vários problemas reais não apresentam informações ou condições ideais para a elaboração de modelos e equações que determinem o comportamento das séries, apresentando fatores complexos a serem analisados, como a não-linearidade dos dados e o comportamento caótico de todo o sistema. Problemas como os mencionados são desta natureza e logo necessitam de outras metodologias, tais como as abordadas pela inteligência computacional, destacando-se entre elas as Redes Neurais Artificiais, ou RNAs.

Várias abordagens baseadas em RNAs têm sido propostas, obtendo resultados promissores. Destacam-se algoritmos de retropropagação através do tempo (*Backpropagation Through Time*), aprendizado recorrente em tempo real (*Real-Time Recurrent Learning*), redes recorrentes Jordan-Elman, dentre outras. O bom resultado obtido por estes modelos, conforme destacado por Forsgren e Kling (2003), se deve pelo fato de tratar problemas não lineares com maior naturalidade devido a própria característica não linear das funções de ativação da rede, bem como do fato do treinamento se abstrair da concepção ideal de equações geradoras das séries e sim aproximações destas.

A proposta deste trabalho baseia-se na abordagem de RNAs para o problema de previsão em séries temporais, aplicando como série de teste o histórico de preços de venda de energia elétrica disponibilizados pela CCEE (Câmara de Comercialização de Energia Elétrica), utilizando como algoritmo principal o *Backpropagation*, em uma rede multicamada, juntamente com um algoritmo genético, responsável pela geração de uma rede neural artificial evolutiva, tornando automatizada a geração da arquitetura da RNA.

2. Previsão Baseada em Séries Temporais e o Projeto Automático de Redes Neurais

2.1 Previsão

A previsão é um problema muito importante e os métodos para efetuar a variam de acordo com o problema e os dados disponíveis. Um tipo de dado que pode ser utilizado para previsão são as séries temporais, que conforme Barros (2004), nada mais são do que um conjunto serialmente dependente de observações de um certo fenômeno ordenadas ao longo do tempo.

A previsão em séries temporais visa eliminar riscos na tomada de decisões de diversos problemas reais. Uma vez que se tenha conhecimento do passado, é possível extrair informações úteis destes dados ao aplicar técnicas de análise de séries temporais e efetuar previsões com uma boa margem de confiança.

Devido a dificuldade em modelar problemas reais e complexos, surgiram técnicas que visam simplificar e auxiliar a previsão em séries temporais que descrevem tais problemas. As principais técnicas de previsão em séries temporais são a modelagem estatística, a modelagem de sistemas inteligentes ou ainda uma modelagem híbrida que utilize ambas as técnicas.

A primeira é considerada a técnica clássica, abordando o problema com a análise da série para extrair padrões e características a partir de conhecimentos prévios por parte de especialistas, para então descrever um modelo estatístico-matemático responsável pela geração da série e assim ser capaz de prever comportamentos futuros.

A segunda abordagem engloba várias propostas de inteligência computacional para previsão em séries temporais tais como: sistemas nebulosos em Petridis e Kehagias (1997), RNAs em Oliveira (2007) e Connor (1994) e RNAs evolutivas em Harrald e Kamstra (1997). Destaca-se aqui a utilização de redes neurais artificiais que possam aprender padrões e o comportamento da

série por conta própria, não sendo necessária a formulação explícita de equações geradoras. A dificuldade de se obter uma boa arquitetura de rede trouxe o projeto evolutivo em RNAs e é este o modelo abordado no presente trabalho.

Vários estudos e aplicações de RNAs para previsão têm sido desenvolvidos. Dentre alguns trabalhos, encontram-se aplicações e novas propostas tais como a “Rede neural de base radial aplicada em previsão de séries temporais” de Coelho e Junior (2000) e a “Aplicação de RNAs na análise de séries temporais econômico-financeiras” de Oliveira (2007).

Para melhor projetar as RNAs, novas idéias surgiram aplicando a computação evolucionária como parte do projeto da rede, definindo aspectos de seu treinamento ou de sua arquitetura, tornando-a assim, uma rede evolutiva como apresenta Mitchell (1997) e Iyoda (2000).

Considerando os avanços nesta linha de pesquisa, Iyoda (2000) destaca que é possível que embora haja uma grande diversidade de abordagens, o potencial de aplicação da computação evolutiva no projeto de redes neurais artificiais ainda não tenha sido totalmente explorado.

2.2. Projeto

Os algoritmos genéticos tem sido utilizados em redes neurais para vários fins: treinar a rede, escolher os melhores atributos ou ainda encontrar uma boa arquitetura. O problema de definição ótima de uma arquitetura de rede neural pode ser visto, conforme Miller (1989), como uma busca por uma arquitetura que apresente melhor desempenho (de acordo com algum critério) em uma tarefa específica.

A arquitetura das redes neurais artificiais é crucial para sua performance e em geral é definida para cada tipo de problema através de uma concepção intuitiva, por tentativa e erro, até que se obtenha uma boa resposta ao problema, sendo assim um dos pontos que mais demoram na concepção de uma RNA. Autores como Caudill (1990), dizem que a escolha do tamanho da camada intermediária deve ter como critério o bom senso: uma camada muito grande memoriza, mas não generaliza muito bem e uma muito pequena apresenta um treinamento muito demorado.

Os esquemas de codificação de RNAs em algoritmos genéticos podem ser classificados em duas categorias principais de acordo com Yao e Liu (1997) e Balakrishnan e Honavar (1995):

- **Codificação direta ou esquema de especificação forte:** todas as conexões e nós de uma arquitetura são especificados no genótipo. Sua maior vantagem é a simplicidade e as desvantagens são que os cromossomos são muito grandes e ocorre a geração de muitas arquiteturas incorretas, necessitando verificação de viabilidade. A codificação direta pode ser exemplificada conforme a figura 1 abaixo.

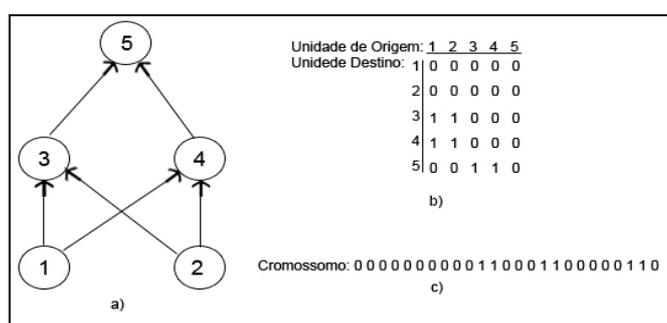


Figura 1. a) RNA com neurônios 1 e 2 na camada de entrada, conectados aos neurônios (3 e 4) na camada intermediária e estes conectados com o neurônio 5 na camada de saída. b) tabela de codificação direta. c) cromossomo que representa a RNA.

- **Codificação indireta ou esquema de especificação fraca (Codificação Gramatical/Sensível ao contexto):** apenas as características mais importantes de uma arquitetura, como o número de camadas escondidas e número de neurônios escondidos são codificadas. Sua vantagem é que tenta-se evitar os problemas da codificação direta e sua desvantagem é que é mais complicada de implementar.

Para o uso da evolução de arquiteturas, Yao (1993) apresenta um esquema simplificado:

- Decodifique cada indivíduo da geração atual em uma arquitetura de rede neural.
- Treine cada rede neural decodificada com um algoritmo de aprendizado fixo e pré-definido.
- Calcule a aptidão de cada indivíduo baseado nos resultados do treinamento.
- Reproduza um número de indivíduos a partir da geração atual de acordo com algum critério de seleção.
- Aplique operadores genéticos e obtenha a nova geração.

3. A Abordagem Proposta

A abordagem proposta utiliza os conceitos de redes neurais artificiais multicamadas treinadas com o algoritmo supervisionado *backpropagation*, sendo efetuada a busca pela melhor arquitetura da rede, variando a quantidade de neurônios nas camadas de entrada, intermediária e saída, com a utilização de um algoritmo genético, responsável pela melhora da arquitetura, de geração em geração, tratando cada indivíduo como uma RNA codificada em forma de um cromossomo binário {0,1}.

A determinação da arquitetura da rede não é fixa, ou seja, a quantidade de neurônios em cada camada pode variar de acordo com a série temporal analisada, com a quantidade de fatos passados e ainda com o tempo computacional que se esteja disposto a usufruir.

Como há modificação no número de neurônios da camada de entrada, ocorre que as várias combinações dos dados de entrada também são exploradas por esta abordagem, permitindo assim uma avaliação de quais atributos irão contribuir de forma mais significativa para a previsão.

Na proposta utilizou-se apenas uma camada intermediária, pois conforme Munakata (2007), além do tempo de processamento aumentar, uma segunda camada intermediária pode ser mais bem aproveitada se houver alguma análise dos dados que influencie a inicialização dos pesos entre a camada de entrada e a intermediária.

A inicialização dos pesos é feita de forma aleatória como mostra a fórmula abaixo:

$$aleatorio = [1 - 2 \times [1 + (RAND_MAX \times rand()) / (RAND_MAX + 1)) / 10000]] / 10$$

O sinal de entrada é passado para a camada intermediária de forma padrão: os sinais de entrada são multiplicados pelos pesos respectivos de suas conexões com os neurônios da camada intermediária e somados. Já a camada intermediária, utiliza a função de ativação sigmóide, como

mostrada na fórmula ao lado: $f(x) = \frac{1}{1 + e^{-x}}$

O resultado do processamento da camada intermediária é passado para a camada de saída onde é anotado o erro de treinamento da rede como a soma do quadrado dos erros. Depois o erro é propagado para trás, realizando a correção de todos os pesos da rede.

Os dados que serão apresentados a rede devem ser agrupados em dois conjuntos, o de treinamento e o de validação. O primeiro conjunto tem por finalidade mostrar como a RNA deve se ajustar, exibindo combinações de dados de entrada com a respectiva saída desejada, treinando a rede para que ao serem apresentados os dados não treinados, se possa obter uma boa previsão de saída. O segundo conjunto tem como função testar se o treinamento da rede foi bem sucedido, ou seja, validar o treinamento e indicar quando o treinamento pode ser interrompido. Assim, a apresentação do conjunto de treinamento, que deve ser normalizado entre 0 e 1 devido a sigmóide utilizada na ativação, é feita apresentando todos os padrões a cada ciclo de treinamento e ao final de cada ciclo é calculado o erro médio de treinamento e o erro médio de validação, o qual é calculado apresentando à rede os dados para validação. Em outro momento, pode-se aplicar *cross-validation* para efetuar o treinamento da RNA.

Foram adotadas as seguintes condições de parada para o treinamento:

- O erro médio de validação atinge um valor aceitável.
- O número máximo de iterações seja atingido.
- A rede decora os dados de treinamento, ou seja, o erro de treinamento diminui e o erro de validação aumenta.

Visando automatizar a geração da arquitetura da RNA, vários trabalhos adotam a codificação direta para mapear a topologia da RNA fixando a quantidade de neurônios da rede, para que se possa manter o tamanho do cromossomo e ser possível efetuar as operações genéticas como em Miller, Hedge e Todd (1989) e em Paz e Kamath (2003) por exemplo.

Em busca de automatizar a quantidade de neurônios da RNA, foi realizada pesquisa para este tipo de abordagem e não foi encontrada uma metodologia de mapeamento da rede para AG, que visasse automatizar a quantidade de neurônios, apenas a mudança da topologia. Desta forma, foi adotada a abordagem abaixo para mapeamento da RNA em um cromossomo binário que codifique apenas a quantidade de neurônios da rede e não sua topologia.

Define-se inicialmente uma quantidade mínima e máxima de neurônios para as camadas de entrada, intermediária e de saída. Uma vez definida as quantidades, cria-se um cromossomo de comprimento igual a soma da quantidade máxima de neurônios em cada camada: $Comprimento_Cromossomo = MAX_ENT + MAX_INTERM + MAX_SAI$.

O próximo passo é gerar aleatoriamente o número de neurônios de cada camada, respeitando cada intervalo mínimo e máximo e uma vez gerada, deve-se preencher o cromossomo com “1s” para indicar que há neurônios na porção referente a camada de entrada, intermediária ou saída e com “0s” para indicar ausência de neurônios.

MIN ENTRADA	3	QENT	3	COMPRIMENTO	18														
MAX ENTRADA	5	QINT	6																
MIN INTERMEDIARIA	3	QSAI	2																
MAX INTERMEDIARIA	10	CROMOSSOMO																	
MIN SAIDA	1	posicoes de entrada		posições da intermediaria						posições saída									
MAX SAIDA	3	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
		1	1	1	0	0	1	1	1	1	1	1	0	0	0	0	1	1	0

Figura 2. Exemplo de codificação de RNA em cromossomo.

Uma vez codificada, a rede é tratada como um individuo dentro de uma população de outras RNAs, sendo possível efetuar a seleção das redes mais aptas para resolver o problema, através dos operadores genéticos de *crossover* e mutação.

A aptidão de cada individuo, será determinada em função do negativo do erro relativo médio (ERM) de validação da RNA, descrita em Schaeffer (1980) e exibida abaixo, onde, N é o número de previsões realizadas, Vc_i é o valor calculado e Vd_i o valor desejado:

$$ERM = \frac{1}{N} \left[\sum_{i=1}^N \left| \frac{Vd_i - Vc_i}{Vd_i} \right| \right]$$

Dessa forma, destaca-se como o mais apto, o individuo que apresentar o maior ERM, significando assim que realizou boa generalização e é uma RNA candidata como melhor solução.

O método de seleção adotado para a escolha dos pares é feito ordenando a aptidão de todos os indivíduos da população inicial e selecionando apenas a primeira metade, ou seja, os mais aptos. O passo seguinte é cruzar seqüencialmente 2 a 2 até que se obtenha uma nova população formada pela melhor metade da geração anterior com os seus respectivos filhos.

Para o cruzamento, foi adotado o sistema de ponto de corte único do cromossomo. Assim, é gerado aleatoriamente um número em um intervalo de corte, IC, para a escolha dos trechos de cromossomos que darão origem aos novos indivíduos.

$$\frac{MAX_ENTRADA}{4} \leq IC$$

$$IC \leq \left[MAX_ENTR + \frac{MAX_INTERM + MAX_SAI}{2} + \frac{MAX_INTERM + MAX_SAI}{4} \right]$$

Após ter sido gerada a nova população, aplica-se a mutação cuja taxa adotada, conforme Paz

e Kamath (2003) é: $\frac{COMPRIMENTO_CROMOSSOMO}{1}$

Assim, de acordo com Haupt (1998), a taxa de mutação determinará a porcentagem de bits dos cromossomos de toda a população que sofrerá mutação. A mutação, com esta codificação, troca o valor 0 para 1 e 1 para 0.



Figura 3. Cromossomo de dois indivíduos da população com o intervalo de corte (IC), o ponto de corte e as operações de *crossover* e de mutação. Verifica-se mudança na arquitetura da rede.

Na presente abordagem nos restringimos em obter primeiramente um resultado direto a respeito da identificação de uma boa arquitetura dentre diversas outras geradas automaticamente, bem como obter estes resultados mais rapidamente em função do tempo computacional. Assim, cada arquitetura representada por um cromossomo tem seu peso instanciado apenas uma vez e a avaliação da melhor rede passa a ser a avaliação do melhor indivíduo.

4. Resultados

A série temporal utilizada foi a de preços do leilão semanal de energia da CCEE entre outubro de 2004 e setembro de 2007, sendo extraídas 156 semanas e os preços equivalentes ao lote Médio NE, da mesma forma que o trabalho “Busca da melhor arquitetura para previsão utilizando Redes Time-Delay”, de Alencar (2007), visando assim estabelecer um comparativo do método automático de concepção da RNA com o método daquele trabalho, onde a arquitetura da rede foi concebida manualmente por tentativa e erro.

O conjunto de treinamento considerou 75% dos valores da série, de forma que o conjunto de treinamento tivesse 117 preços, e os 39 restantes para validação.

Foi definido que a quantidade de neurônios na camada de entrada variasse entre 3 e 36 e a quantidade de neurônios na saída entre 1 a 4 para melhor comparação dos resultados obtidos em Alencar (2007). Adotou-se como número máximo de neurônios na camada intermediária o dobro do número na entrada, ou seja, 72. Com esta configuração, o comprimento do cromossomo que codifica a RNA é de $36 + 72 + 4$, ou seja, 112.

Considerou-se uma população de 24 indivíduos, gerados aleatoriamente conforme o intervalo de quantidade de neurônios em cada camada. Assim, da posição 0 até a 35 são armazenadas as informações da camada de entrada; da posição 36 a 107, as da camada intermediária e da posição 108 a 111, as da camada de saída.

Os parâmetros para treinamento da rede foram definidos como:

- Número máximo de iterações: 10 mil.
- Taxa de aprendizado: 0.05.
- Número de camadas intermediárias: 1.

- Condições de parada da rede: atingir o número máximo de iterações, ou obter um erro médio de validação da ordem de $1.0e-4$ ou ainda se ocorrer por 500 iterações consecutivas uma diminuição do erro de treinamento e aumento do erro de validação, ou seja, a rede decora os valores.
- Condições de parada do AG: obter uma rede cuja aptidão é maior do que a desejada ou obter a vigésima geração de indivíduos.

Os testes realizados em Alencar (2007) foram realizados fixando alguns elementos da arquitetura, tais como número de neurônios na entrada e saída:

- Fixos: 2 neurônios na entrada e 1 na saída, chamado 2x1.
- Fixos: 4 neurônios na entrada e 2 na saída, chamado 4x2.

Para efeito de comparação, como o presente trabalho visa expandir a busca da melhor arquitetura variando a quantidade de neurônios em todas as camadas, fixou-se apenas a quantidade de neurônios na camada de saída:

- 1 neurônio na saída: Teste S1.
- 2 neurônios na saída: Teste S2.

Em ambos ocorre mudanças na quantidade de neurônios na camada intermediária, mas a mudança do número de neurônios no projeto automático desenvolvido é mais ampla, aumentando o espaço de busca da melhor arquitetura e por isso foi aplicado o AG para efetuar a busca da melhor rede automaticamente.

4.1 Resultados dos Testes S1

A tabela 1 abaixo exhibe o resultado da última população gerada em busca de um ERM menor que o melhor ERM dos testes 2x1 mostrada na tabela 2.

Tabela 1. 4ª população gerada em busca de um indivíduo (RNA), cuja aptidão fosse maior do que o negativo do menor ERM da tabela 2. Entrada, Intermediária e Saída representam respectivamente a quantidade de neurônios em cada camada.

4ª População Gerada						
Indivíduo	Entrada	Intermediária	Saída	ERM	ERM*100	Aptidão
1	21	34	1	0,153988575	15,39885748	-0,153988575
2	8	9	1	0,164744575	16,47445748	-0,164744575
3	31	32	1	0,373817719	37,38177189	-0,373817719
4	16	15	1	0,13626857	13,62685699	-0,13626857
5	14	4	1	0,133030897	13,30308975	-0,133030897
6	10	65	1	0,173683379	17,36833789	-0,173683379
7	24	19	1	0,181458954	18,14589539	-0,181458954
8	11	10	1	0,170975096	17,09750963	-0,170975096
9	5	36	1	0,135641518	13,5641518	-0,135641518
10	5	14	1	0,12955198	12,95519796	-0,12955198
11	35	43	1	0,454063095	45,40630952	-0,454063095
12	31	6	1	0,34810601	34,81060104	-0,34810601
13	22	21	1	0,154903644	15,4903644	-0,154903644
14	22	42	1	0,194695071	19,46950713	-0,194695071
15	15	34	1	0,135660707	13,56607074	-0,135660707
16	34	26	1	0,450349621	45,03496208	-0,450349621
17	16	23	1	0,134136038	13,41360378	-0,134136038
18	18	53	1	0,138474847	13,84748473	-0,138474847
19	19	33	1	0,134620306	13,46203055	-0,134620306
20	28	19	1	0,298432033	29,8432033	-0,298432033
21	14	27	1	0,132026115	13,20261153	-0,132026115
22	31	53	1	0,421218798	42,12187982	-0,421218798

23	28	13	1	0,312157419	31,21574194	-0,312157419
24	21	67	1	0,147654865	14,7654865	-0,147654865

Tabela 2. Tabela de resultados dos testes de arquitetura 2x1, apresentando como menor ERM a arquitetura com 2 neurônios na entrada, 1 camada intermediária com 2 neurônios e 1 neurônio na saída. (Adaptada de Alencar, M. 2007)

Arquitetura	ERM
2x1..2x1	0.13982191217708953
2x1..4x1	0.14219270308878912
2x2..2x1	0.14085773183584688
2x2..4x1	0.14443974034578144

Observa-se na tabela 1 que vários indivíduos (em negrito) obtiveram um erro relativo médio abaixo do menor ERM das arquiteturas 2x1, que foi de 0.1398219 para a primeira arquitetura. Encontra-se destacado em cinza na tabela 1, o indivíduo que obteve o menor ERM, conseqüentemente o mais apto.

Observando a tabela 1, podemos perceber que uma variedade de arquiteturas, 8 das 24 geradas, apresentam um ERM abaixo de 0.1398219. Assim, embora existam indivíduos com baixa aptidão dentro da população, o AG foi capaz de selecionar 8 indivíduos com aptidão melhor do que a desejada.

A seguir é exibido na figura 4 o gráfico referente à tabela 1, onde se percebe a relação entre o ERMx100 e a quantidade de neurônios em cada camada, indivíduo por indivíduo.

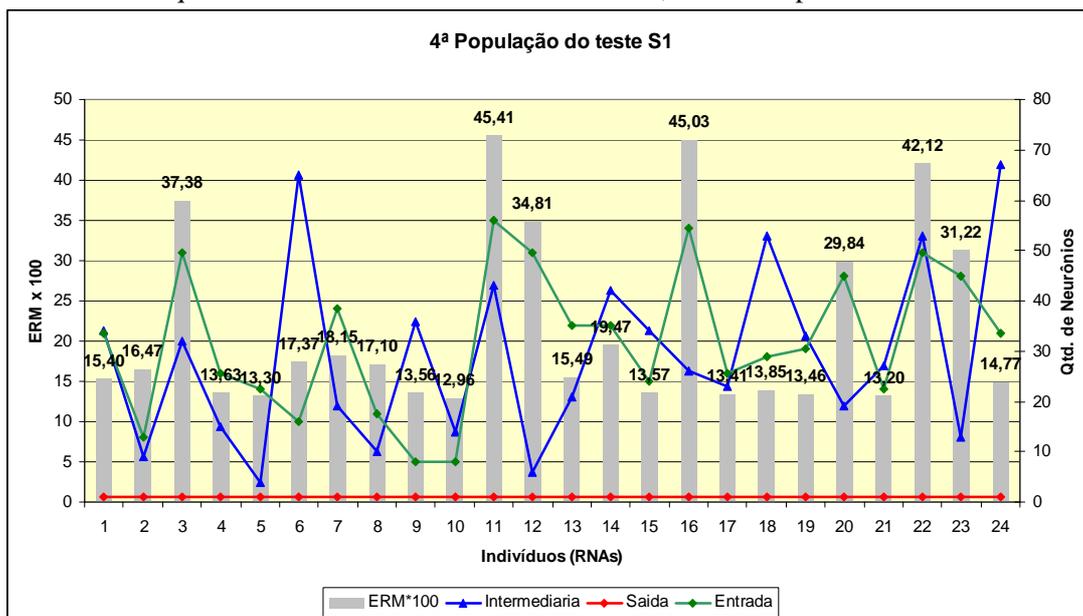


Figura 4. Gráfico da Tabela 1: ERMx100-Indivíduos-Qtd.Neurônios.

Em seguida é exibido o gráfico de validação da rede gerada automaticamente na figura 5.

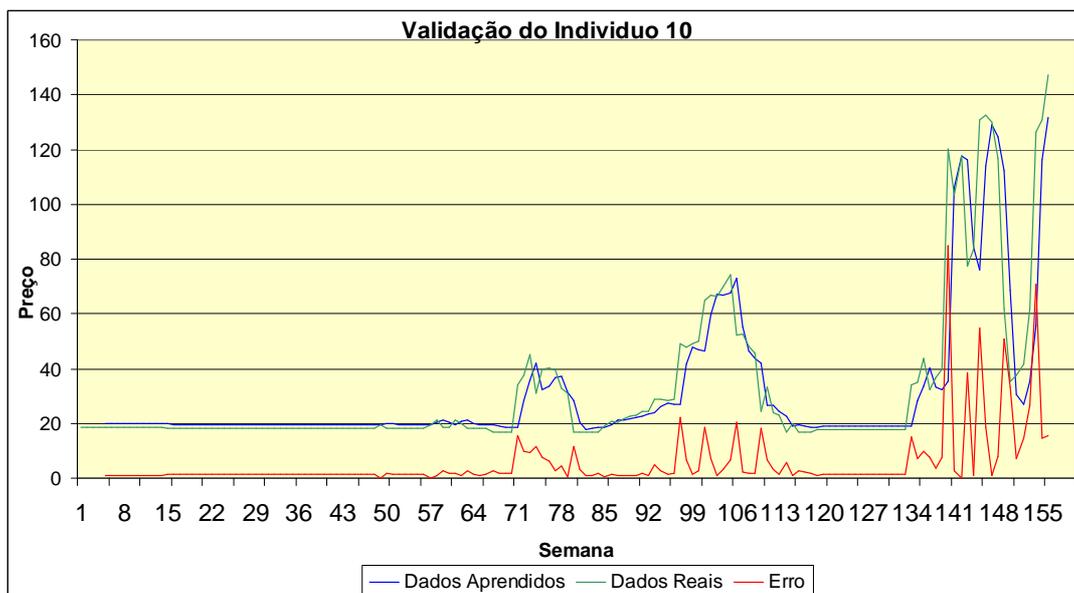


Figura 5. Gráfico de validação dos dados para o indivíduo 10.

Observando o gráfico de validação, podemos notar principalmente na região dos dados não treinados pela rede, os dados de validação, que são as 39 últimas semanas da série indicando que a previsão foi melhor, reduzindo o erro naquela região.

4.2 Resultados dos Testes S2

O teste S2 é comparado ao teste com arquitetura 4x2, uma vez que possuem 2 neurônios na camada de saída.

A tabela 3 abaixo exibe o resultado da última população gerada em busca de um ERM menor que o melhor ERM dos testes 4x2 mostrada na tabela 4

Tabela 3. 6ª população gerada em busca de um indivíduo (RNA), cuja aptidão fosse maior do que o negativo do menor ERM da tabela 4. Entrada, Intermediária e Saída representam respectivamente a quantidade de neurônios em cada camada.

6ª População Gerada						
Indivíduo	Entrada	Intermediária	Saída	ERM	ERM*100	Aptidão
1	10	39	2	0,187654865	18,7654865	-0,187654865
2	27	16	2	0,312157419	31,21574194	-0,312157419
3	17	15	2	0,331218798	33,12187982	-0,331218798
4	13	23	2	0,17955198	17,95519796	-0,17955198
5	18	53	2	0,308432033	30,8432033	-0,308432033
6	22	14	2	0,174620306	17,46203055	-0,174620306
7	29	62	2	0,178474847	17,84748473	-0,178474847
8	6	32	2	0,174136038	17,41360378	-0,174136038
9	21	26	2	0,460349621	46,03496208	-0,460349621
10	13	50	2	0,175660707	17,56607074	-0,175660707
11	5	33	2	0,204695071	20,46950713	-0,204695071
12	5	17	2	0,174903644	17,4903644	-0,174903644
13	15	34	2	0,35810601	35,81060104	-0,35810601
14	22	21	2	0,364063095	36,40630952	-0,364063095
15	14	22	2	0,172026115	17,20261153	-0,172026115
16	16	15	2	0,175641518	17,5641518	-0,175641518
17	21	34	2	0,187975096	18,79750963	-0,187975096
18	19	33	2	0,191458954	19,14589539	-0,191458954

19	21	67	2	0,188683379	18,86833789	-0,188683379
20	14	27	2	0,173030897	17,30308975	-0,173030897
21	18	53	2	0,17626857	17,62685699	-0,17626857
22	16	23	2	0,383817719	38,38177189	-0,383817719
23	5	36	2	0,174744575	17,47445748	-0,174744575
24	9	32	2	0,173988575	17,39885748	-0,173988575

Tabela 4. Tabela de resultados dos testes de arquitetura 4x2, apresentando como menor ERM a arquitetura com 4 neurônios na entrada, 1 camada intermediária com 4 neurônios e 2 neurônios na saída. (Adaptada de Alencar, M. 2007)

Arquitetura	ERM
4x1..4x2	0.18761485251043994
4x1..8x2	0.18797932859073266
4x2..4x2	0.19448361955424864
4x2..8x2	0.24575749788601034

Observa-se na tabela 3 que vários indivíduos (em negrito) obtiveram um erro relativo médio abaixo do menor ERM das arquiteturas 4x2, que foi de 0.1876148 para a primeira arquitetura. Encontra-se destacado em cinza na tabela 3, o indivíduo que obteve o menor ERM, conseqüentemente o mais apto.

Neste teste, 12 RNAs dentre as 24 geradas, apresentam um ERM abaixo de 0.1876148, Assim, o AG foi capaz de selecionar 12 indivíduos com aptidão melhor do que a desejada. A seguir é exibido na figura 6 o gráfico referente à tabela 3.

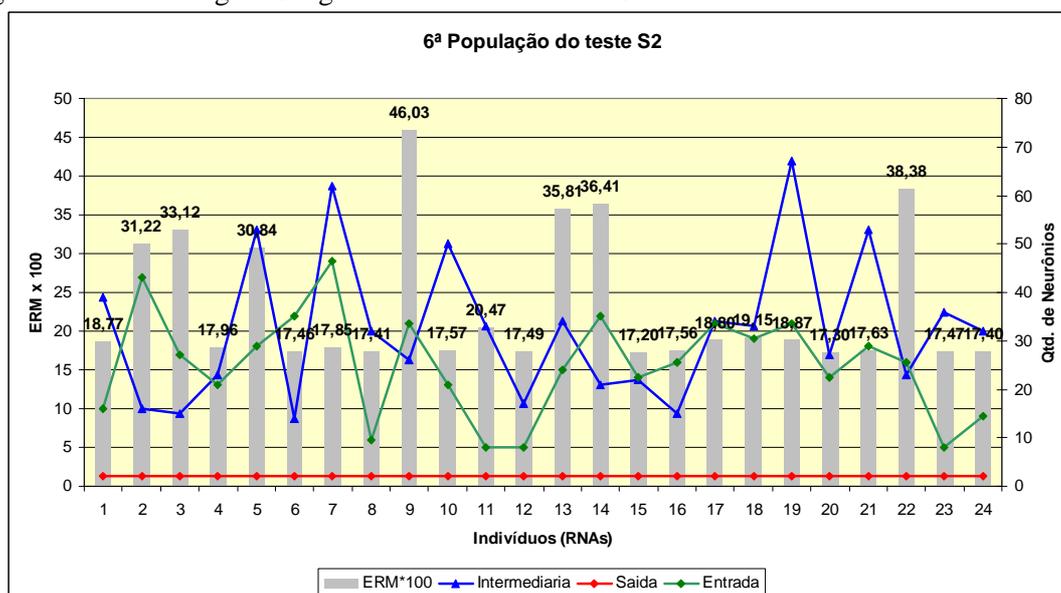


Figura 6. Gráfico da Tabela 3: ERMx100-Individuos-Qtd.Neurônios.

A seguir são exibidos os gráficos de validação da rede gerada automaticamente na Figura 7 e em seguida é mostrado também o mesmo gráfico para a rede 4x1..4x2 na Figura 8.

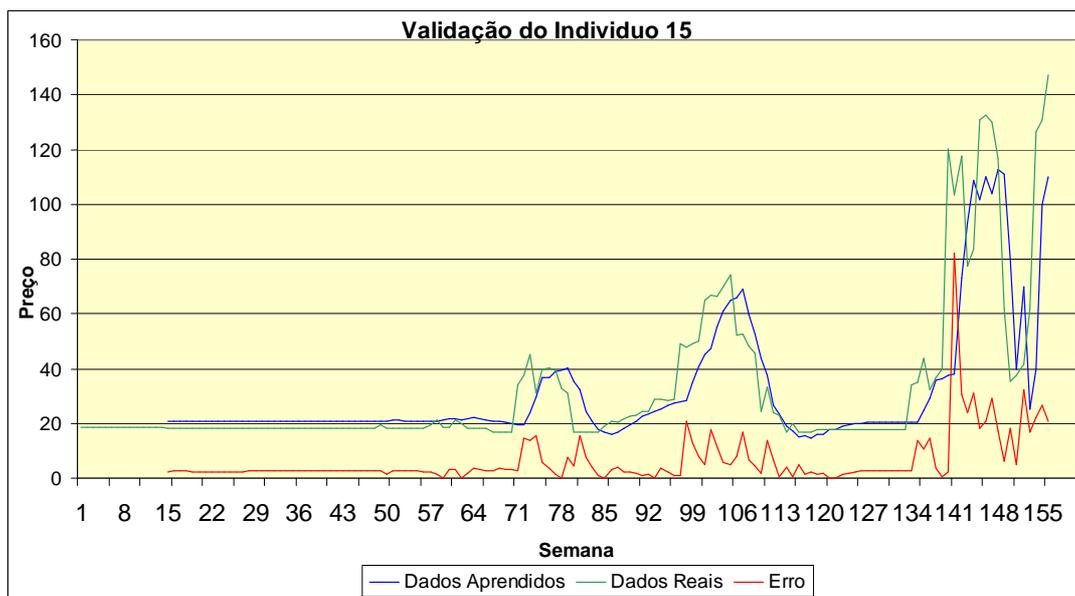


Figura 7. Gráfico de validação dos dados para o indivíduo 15.

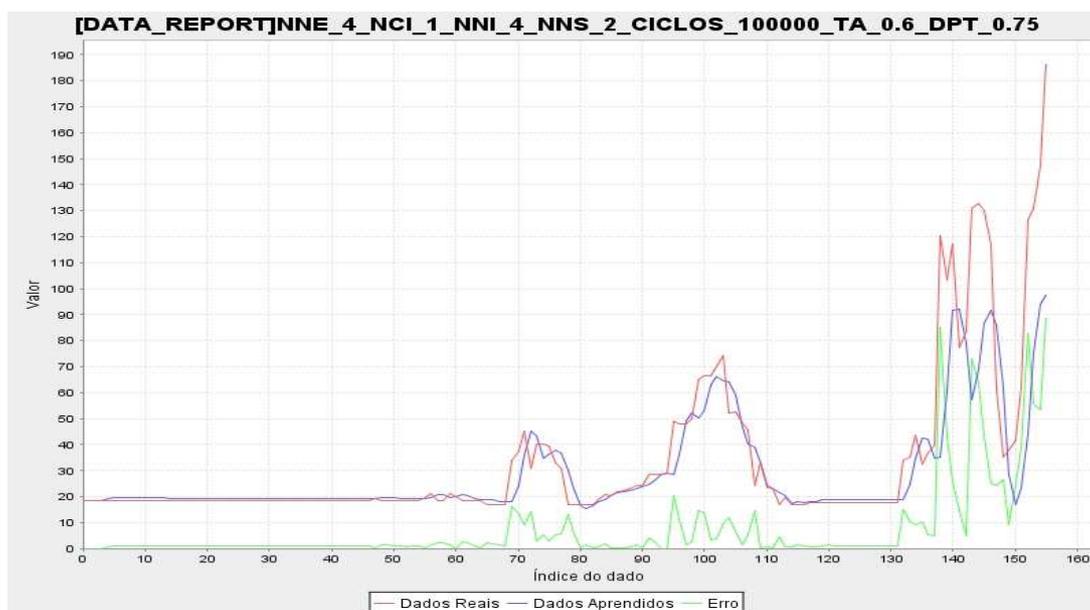


Figura 8. Gráfico de validação dos dados para a rede 4x1..4x2 (Adaptada de Alencar, M. 2007)

Observando os gráficos de validação, podemos notar poucas diferenças. Assim, embora obtendo um ERM menor do que a 4x1..4x2, o desempenho geral foi bastante semelhante. Neste caso, seria interessante permitir ao AG continuar a busca.

4.3. Conclusão dos Testes

Foi possível observar a capacidade da metodologia abordada para melhorar a busca por boas arquiteturas de RNAs, uma vez que em todos os testes apresentados, um ou mais indivíduos se destacaram com boas arquiteturas e um baixo ERM.

No teste S1, com apenas um neurônio na camada de saída, o comportamento dos indivíduos foi melhor, apresentando erros relativos médios mais baixos dos que nos outros testes. Nota-se com este fato, a dificuldade de se realizar previsões a longo prazo, uma vez que vários neurônios na saída, justamente buscam extrapolar ao máximo o horizonte de previsão.

Observou-se também que a série temporal utilizada prejudica o treinamento, pois o intervalo considerado para testes apresenta 2 anos de comportamento praticamente constante (2004 e 2005). Este tipo de característica mascara alguns pontos do treinamento, tornando difícil para algumas arquiteturas, prever os valores futuros já que se cria um aspecto de tendência constante.

5. Considerações Finais

Ao longo do desenvolvimento do trabalho, foi possível perceber o grande potencial das redes neurais evolutivas para o problema da previsão em séries temporais, principalmente no que diz respeito ao projeto automático de RNAs e a busca por boas arquiteturas.

Os teste realizados mostraram que é possível automatizar a escolha do número de neurônios para cada camada, evitando assim a perda de tempo na concepção de qual será a melhor arquitetura para a RNA.

Podemos citar como evoluções do trabalho, o estudo de novas séries temporais aplicadas diretamente na metodologia proposta, a troca de codificação direta do cromossomo para a codificação indireta, a apresentação de mais de uma variável para a série, de modo que ela possa adequar o número de neurônios de acordo com os dados mais importantes.

Outras evoluções na metodologia proposta incluem mudanças nos operadores genéticos, tanto de mutação e de *crossover*, como o de seleção. Ainda no campo de algoritmos genéticos, pode-se acrescentar a automação do processo de treinamento para atualização dos pesos bem como para a mineração dos dados de entrada da rede. Pode-se ainda, fazer uma evolução híbrida da RNA, alterando tanto a topologia de suas conexões juntamente com a alteração da quantidade de neurônios em suas camadas.

6. Referências

- Alencar, M. M.**, (2007), Busca da melhor arquitetura para previsão utilizando Redes Time-Delay. TCC – Ciência da Computação – Universidade Estadual do Ceará.
- Barros, M.**, Processos Estocásticos. 1ª ed. Papel Virtual, Rio de Janeiro 2004.
- Bernacki, M e Wlodarczyk, P.**, Students' Project: "Neural networks and artificial intelligence in electronics". Principles of training multi-layer neural network using backpropagation algorithm. [Online] 2005. http://galaxy.agh.edu.pl/~vlsi/AI/backp_t_en/backprop.html.
- Campos, L. M. L.**, (2001), Metáforas Biológicas Combinadas Para Projeto de Redes Neurais Artificiais. Tese de Mestrado – UFSC.
- Ehlers, Ricardo S.**, (2007), Análise de Séries Temporais. Laboratório de Estatística e Geoinformação Universidade Federal do Paraná.
- Haupt L., Haupt, E.**, (1998), Practical Genetic Algorithms
- Iyoda, E. M.**, (2000), Inteligência Computacional no Projeto Automático de Redes Neurais Híbridas e Redes Neurofuzzy Heterogêneas. Tese de Mestrado – Unicamp.
- Kasabov, K. N.**, Foundations of Neural Networks, Fuzzy Systems, and Knowledge Engineering. MIT Press. London, England, 1998
- Mitchell, M.**, An Introduction to Genetic Algorithms. MIT Press, Cambridge, USA 1996
- Munakata, T.**, Fundamentals of the New Artificial Intelligence. Springer, Londres, 2008
- Oliveira, M. A.**, (2007), Aplicação de Redes Neurais Artificiais Na Análise de Séries Temporais Econômico-Financeiras. Tese de Doutorado – USP.
- Paz, E., Kamath C.**, Evolving Neural Networks for the Classification of Galaxies. Center for Applied Scientific Computing Lawrence Livermore National Laboratory Livermore, CA, 2003.
- Yao, X.** (1999), Evolving artificial neural networks. Proceedings of the IEEE 87, 1423-1447.

7. Agradecimentos

FUNCAP – Fundação Cearense de Apoio ao Desenvolvimento Científico.