

ALGORITMOS EVOLUCIONÁRIOS PARA OTIMIZAÇÃO DE REDES NEURAIS ARTIFICIAIS

Aida FERREIRA (1); Thiago HENRIQUE(2);

(1) Instituto Federal de Pernambuco - IFPE, Av. Prof Luiz Freire, 500 Cidade Universitária - Recife/PE - Brasil, aidaaf@gmail.com

(2) Instituto Federal de Pernambuco - IFPE, Av. Prof Luiz Freire, 500 Cidade Universitária - Recife/PE Brasil, thiago.oliveira19@gmail.com

RESUMO

A otimização global é uma tarefa que visa escolher a melhor alternativa de um conjunto viável de possibilidades, de modo a aperfeiçoar a função objetivo do problema. Os algoritmos mais utilizados para tratar de problemas de otimização numérica podem ser divididos em duas classes: algoritmos específicos para um determinado problema em particular e algoritmos meta-heurísticos de busca para um vasto número de problemas de otimização. Na primeira classe se enquadram, por exemplo, as técnicas de programação linear, programação quadrática e de mínimos quadrados, enquanto que na segunda, podemos citar os algoritmos evolucionários, onde estão inseridos os algoritmos genéticos, que fazem parte do escopo desse trabalho. O treinamento de redes neurais artificiais (RNA) consiste da determinação de valores adequados para seus pesos e parâmetros. O ajuste de pesos e parâmetros de redes neurais artificiais pode ser reformulado como um problema de otimização numérica. Como os algoritmos evolucionários são um método atrativo para otimização de pesos e parâmetros das RNA, neste trabalho nós apresentamos um estudo de caso da busca de pesos e parâmetros de redes neurais recorrentes com algoritmos genéticos aplicados a uma tarefa de previsão de séries temporais. O tempo da busca da melhor rede com algoritmo genético foi de 0,000546 do tempo gasto por uma busca exaustiva dos mesmos parâmetros. Os resultados obtidos com a metodologia proposta neste trabalho foram muito melhores que os resultados obtidos pela previsão realizada com o método da persistência.

Palavras-chave: algoritmos genéticos, redes neurais artificiais, otimização, séries temporais

1. INTRODUÇÃO

Redes Neurais Artificiais são sistemas paralelos distribuídos compostos por unidades de processamento simples (neurônios artificiais) que calculam determinadas funções matemáticas (normalmente não-lineares) fortemente baseadas em modelos biológicos. Tais unidades são dispostas em uma ou mais camadas e interligadas por um grande número de conexões, geralmente unidirecionais. Na maioria dos modelos essas conexões estão associadas a pesos, os quais armazenam o conhecimento adquirido pelo modelo e servem para ponderar a entrada recebida para cada neurônio da rede.

Reservoir computing (RC) é um novo paradigma de redes neurais recorrentes que oferece uma metodologia intuitiva para uso de processamento temporal sem um esforço grande para treiná-la. Foram inicialmente apresentadas como Echo State Networks por Jaeger (2001) e Liquid State Machines por Maass et al. (2002). A diferença entre a Echo State Networks e as Liquid States Machines é que a segunda teve uma inspiração baseada na realidade biológica enquanto a primeira foi desenvolvida para resolver problemas matemáticos. Contudo, segundo Lukoservicius et al (2009), ambas introduziram um novo paradigma no treinamento de redes neurais artificiais recorrentes. Neste trabalho nós utilizamos apenas as Echo State Networks, Jaeger (2001).

Por sua vez, Algoritmos genéticos (AG) são um ramo dos algoritmos evolucionários (modelos computacionais baseado nos processos naturais da evolução) e como tal podem ser definidos como uma técnica de busca baseada numa metáfora do processo biológico de evolução natural, Linden (2008). Os algoritmos genéticos são técnicas heurísticas (método de aproximação das soluções dos problemas, que não segue um percurso claro, mas se baseia na intuição e nas circunstâncias para gerar conhecimento novo) de otimização global.

Pode-se definir séries temporais como sendo uma coleção de observações no tempo e ordenadas sequencialmente. As principais características de uma série temporal são: existe uma dependência entre os dados vizinhos, interesse na análise e modelagem da dependência entre os dados, é muito importante a ordem de observação dos dados, ou seja, a sequência com que os dados são observados, diferentemente de outros procedimentos estatísticos, nas séries temporais faz-se necessário o uso de procedimentos específicos, devido à dependência entre os dados.

Esse trabalho buscou otimizando uma Echo State Networks através de Algoritmos Genéticos fazer previsões de séries temporais. O que mostrou um bom resultado, comparando-se com o método da persistência.

1 1. FUNDAMENTAÇÃO TEÓRICA

2.1 Redes Neurais Artificiais

Para entender como funcionam as redes neurais artificiais é preciso entender como funcionam os neurônios biológicos. O neurônio é a unidade fundamental constituinte do sistema nervoso. Ele é constituído de um corpo denominado **corpo somático**, no qual são produzidos os impulsos nervosos, e de prolongamentos finos e delgados através dos quais estes impulsos são transmitidos e recebidos que são os **dendritos** (mais curtos e ramificados, através dos quais são recebidos os impulsos nervosos provenientes dos órgãos receptores e que se destinam ao corpo central) e os **axônios** (através do qual a célula nervosa transmite os impulsos nela originados). Os neurônios não trabalham de forma autônoma, mas em estreita colaboração recíproca. Os neurônios são associados reciprocamente através de **sinapses** que podem ser químicas ou elétricas.

O modelo de neurônio artificial que foi inicialmente proposto por McCulloch e Pitts (1943) que se usa atualmente é muito semelhante ao neurônio biológico. As sinapses são substituídas por pesos e o corpo somático é, agora, um somador seguido de uma função limitadora. A saída única representa o axônio e pode ser usada por outros neurônios, como no caso natural. A operação desse neurônio é muito simples: as entradas são apresentadas ao neurônio e multiplicadas cada uma por um peso. No resultado desta operação geralmente é aplicada uma função não-linear, produzindo o resultado de saída do neurônio (ver Figura 1).

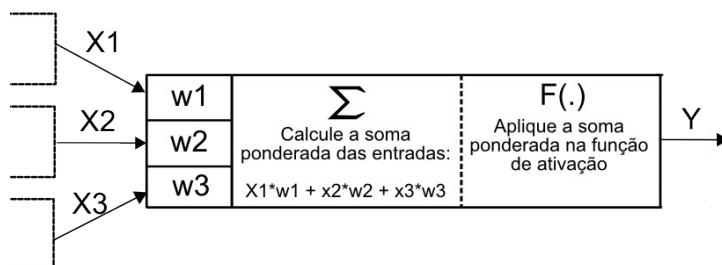


Figura 1 - Estrutura de um neurônio artificial

2.1.2 Tipos de Redes Neurais Artificiais

Enquanto nas redes neurais artificiais (RNA) de uma única camada (ver Figura 2 a), como a rede **Perceptron** ou a rede **Adaline**, pode-se corrigir os pesos da rede comparando-se o sinal de saída com um sinal esperado (no caso do aprendizado supervisionado com professor). Na RNA do tipo **Perceptron de Múltiplas Camadas (MLP)** tem-se a dificuldade de treinar as camadas intermediárias já que se possui apenas o resultado esperado da camada de saída o que dificulta no momento de treinar a camada intermediária (ver Figura 2 b). Para resolver esse problema, desenvolveu-se o algoritmo chamado **Backpropagation** inicialmente proposto por Rumelhart et al (1986) que consiste em dois passos: no primeiro passo propagam-se os sinais no sentido progressivo (da camada de entrada para a camada de saída), calculando-se o sinal de saída e o erro. Os pesos permanecem fixos; no segundo passo, os erros são propagados recursivamente (da camada de saída para a camada de entrada) através da rede e os pesos são ajustados através da regra de ajuste dos pesos (regra delta generalizada).

Também existem as redes neurais recorrentes que se distinguem das outras por ter pelo menos um laço de realimentação (ver Figura 2 c). A presença de laços de realimentação tem um impacto profundo na capacidade de aprendizagem da rede e no seu desempenho. Esta é a arquitetura do novo paradigma de redes neurais recorrentes conhecido por Reservoir Computing.

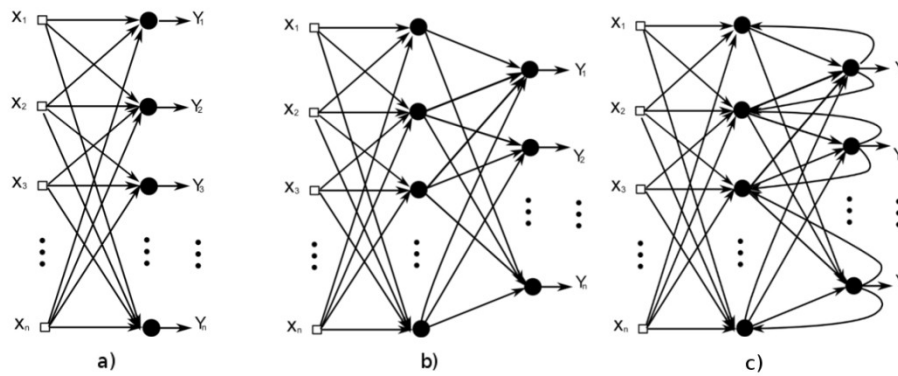


Figura 2 - Arquiteturas de Redes Neurais Artificiais.

2.2 Reservoir Computing

Reservoir Computing é um novo paradigma de redes neurais recorrentes que oferece uma metodologia intuitiva para uso de processamento temporal sem um esforço grande para treiná-la. Foram inicialmente apresentadas como Echo State Networks por Jaeger (2001) e Liquid State Machines por Maass et al. (2002). A principal diferença entre esses dois tipos de redes neurais é que as Liquid State Machines utilizam modelos sofisticados e biologicamente realísticos de neurônios e modelos de conexões de sinapses dinâmicas no reservatório enquanto as Echo State Networks focam na resolução de problemas matemáticos, como a previsão de séries temporais.

O conceito básico é construir uma RNR aleatória e deixar os pesos da mesma fixos. Uma função de regressão linear é treinada separadamente com as respostas do reservatório para os sinais de entrada. O fato de existir recorrência nessas redes tem um profundo impacto, pois:

- Uma Rede Neural Recorrente (RNR) pode desenvolver uma ativação dinâmica temporal autosustentada em seus caminhos interconectados, mesmo na ausência de sinais de entrada. Matematicamente, RNRs com essa característica são consideradas sistemas dinâmicos.
- Se sinais de entrada forem dirigidos a uma RNR, ela preservará no seu estado interno uma transformação não-linear do histórico das entradas - em outras palavras, haverá uma memória dinâmica, e então a rede estará apta a processar informação de contexto temporal.

Neste trabalho nós utilizamos apenas as Echo State Networks (ESN), Jaeger (2001). Na figura 3a pode-se observar um esquema de uma ESN, onde verifica-se a entrada (k unidades de entrada), o reservatório (as N unidades internas) e o readout (as L unidades de saída). Na figura 3b apresentamos a topologia padrão de uma ESN

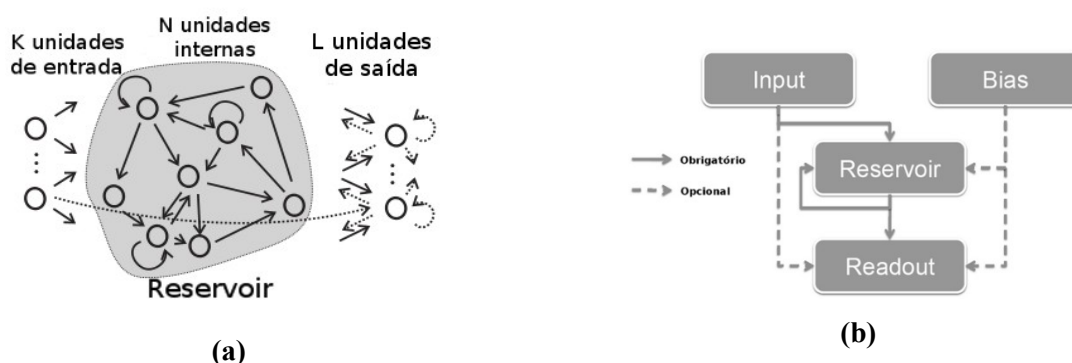


Figura 3 – Echo State Network (a) e Topologia Padrão de ESN (b)

2.2.1 Fluxo padrão de execução de ESNs

Inicialmente uma rede de nós recorrentes (o reservóir) é criada com pesos aleatórios segundo uma distribuição normal com média zero e variância 1. Logo depois, uma classificação separada ou função de regressão (a função readout) é treinada pela resposta do reservóir para os sinais de entrada, Jaeger (2001).

A sequência de passos para criação de uma ESN no nosso trabalho segue o seguinte fluxo:

1. Gerar conjunto de dados
2. Gerar topologia (ver figura 5)
3. Simular reservóir
4. Treinar a camada de readout e avaliar os resultados com o conjunto de teste.

2.3 Algoritmos Genéticos

Algoritmo genético (AG) é um método popular de algoritmos de otimização ou algoritmos iterativos gerais baseados na teoria da evolução natural, Back (1986). As características dos algoritmos genéticos são:

- São algoritmos de aproximação, não garantem encontrar uma solução ótima.
- São cegos, não sabem quando encontraram uma solução ótima. Também não sabem quando interromper a busca a esta solução.
- Possuem a capacidade de ocasionalmente aceitarem movimentos que se afastem de uma solução ótima.
- São gerais, podem facilmente implementar uma diversidade de problemas, bastando para esta tarefa, adaptar a representação da solução, função de custo e o mecanismo para gerar novas soluções sobre o espaço de busca.
- Em certas condições, podem assintoticamente convergir para uma solução ótima.
- O processo de busca do algoritmo genético envolve uma sequência de iterações, onde um conjunto de soluções passa pelos processos de seleção e reprodução.

2.3.1 Operadores genéticos

Os operadores genéticos consistem em aproximações computacionais de fenômenos vistos na natureza, como a reprodução sexuada, a mutação genética e quaisquer outros que se consiga reproduzir.

Operador de crossover

O operador de crossover é operador mais importante dos algoritmos genéticos. O operador de crossover, somado ao módulo de seleção proporcional à avaliação de um indivíduo, é responsável pelo fato de um algoritmo genético não poder ser comparado a uma busca aleatória. No caso do crossover de um ponto (1) inicialmente são escolhidos dois pais através da seleção depois é (2) sorteado um ponto dentre todos os pontos de corte do cromossomo para ser efetuado o crossover. (3) Ao se escolher o ponto, são trocados entre os dois pais a porção de genes referentes a cada um dos dois pais e assim são formados dois filhos podendo (4) esses filhos ser posteriormente alvo do operador de mutação (ver Figura 4).

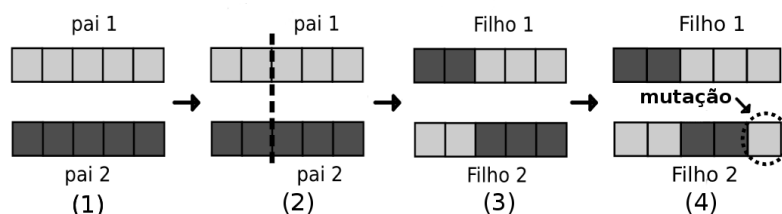


Figura 4 – Operador de crossover e mutação em ação

Operador de mutação

O valor da probabilidade que decide se o operador de mutação será ou não aplicado é um dos parâmetros do Algoritmo Genético que apenas a experiência pode determinar. O conceito fundamental quanto ao valor da probabilidade é que ele deve ser baixo. Se ele for muito alto, o algoritmo genético se parecerá com uma

técnica chamada caminhada aleatória (simplesmente sorteando-se elementos sem usar informações correntes ou passadas).

3. DESCRIÇÃO DA PROPOSTA

A proposta desse trabalho é utilizar a técnica de AG para buscar pesos e topologia de Echo State Networks para previsão da velocidade média horária dos ventos na região nordeste do Brasil, especificamente da cidade de São João do Cariri do estado do Paraná.

4. BASE DE DADOS, METODOLOGIA e ESTUDO DE CASO

4.1 Base de dados

Os dados utilizados no nosso trabalho são dados disponibilizados pelo projeto SONDA (Sistema de Organização Nacional de Dados Ambientais - <http://sonda.cptec.inpe.br/>). O projeto SONDA é um projeto do Instituto Nacional de Pesquisas Espaciais (INPE) para implementação de uma infraestrutura física e de recursos humanos destinado ao levantamento e melhoramento do banco de dados dos recursos de energia solar e eólica no Brasil.

Escolhemos a série de velocidade dos ventos de São João do Cariri (SCR), que é uma cidade do estado da Paraíba e está a 458m de altitude. Os dados utilizados nos nossos experimentos correspondem às velocidades médias horárias desde 01/01/2006 até 31/12/2007 tendo um total de 17.520 registros.

4.2 Método da persistência

Este método foi desenvolvido por meteorólogos como uma ferramenta de comparação para complementar a previsão numérica do clima. Como a exatidão de termos muito pequenos de previsão não foi historicamente considerado importante, a persistência foi suficiente.

O método da persistência assume que as condições do tempo da previsão não mudarão. Por exemplo, se estiver fazendo 30° Celsius hoje em determinada hora, o método da persistência predirá que amanhã nessa mesma hora estará fazendo 30°. Este método trabalha bem com padrões climáticos que mudam muito pouco ao longo do tempo.

Os resultados obtidos com a metodologia proposta neste trabalho serão comparados com os resultados obtidos pela previsão realizada com o método da persistência.

Nas figuras 5 e 6 são exibidas as previsões com modelo de Persistência e com RC, respectivamente, onde a linha vermelha é a previsão da velocidade do vento e a preta é a série temporal da velocidade do vento real de São João do Cariri.

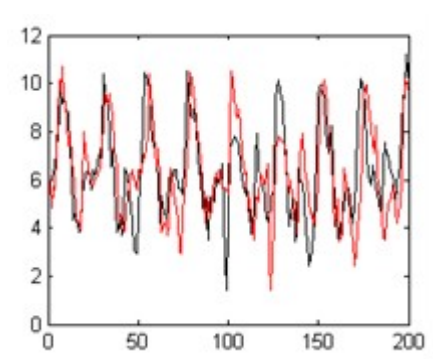


Figura 5 – Previsão com modelo de Persistência.

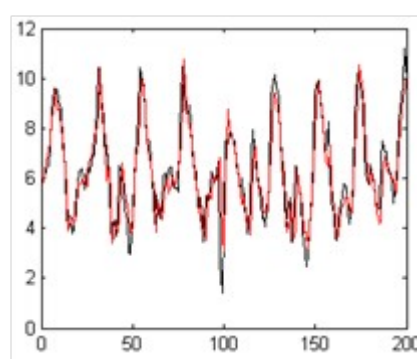


Figura 6 – Previsão com ESN.

4.3 Metodologia

Para resolver um problema de aprendizado de pesos de uma rede neural artificial usando AGs, é necessário transformar a estrutura do problema em alguma forma compreensível para o AG, criando uma codificação adequada. Tem-se, pois, que desenvolver uma estrutura para o AG tal que se possa restringir em blocos a influência da alteração de sinapses, de modo que os operadores de crossover e de mutação possam operar e gerar resultados consistentes.

Em geral, o treinamento das redes neurais artificiais é realizado através de repetidas tentativas com diferentes topologias de rede, até serem obtidos resultados satisfatórios para o problema. Além de consumir tempo e exigir experiência no treinamento de redes neurais artificiais, este processo pode obter redes com conexões e unidades de processamento desnecessárias. Sem considerar o fato de que quanto maior a topologia da rede, mais complexo é o ajuste do valor destas conexões. Desta forma, a otimização de topologia e pesos de redes neurais artificiais através de algoritmos evolucionários é uma abordagem interessante para a geração de redes eficientes com topologias pequenas.

O problema de encontrar um bom ESN utilizando-se de AG pode ser aproximado de três diferentes formas: primeiro, como um processo que opera diretamente na topologia da rede, segundo, como um operador evolucionário que trabalha com parâmetros estocásticos que gera um ESN e, terceiro, com ambos as formas ao mesmo tempo. Se o método evolucionário opera diretamente nas conexões (primeiro e terceiro caso) o espaço de busca pode ser muito grande. Para uma rede com N neurônios, o cromossomo teria elementos e é necessário combinar os K indivíduos das unidades de entrada com as N unidades internas e com as L unidades de saída. A segunda aproximação resulta em um espaço de busca menor para o processo evolucionário. Um problema dessa segunda abordagem seria uma investigação imprecisa para os parâmetros globais, especialmente porque a rede interconectada e pesos são geradas aleatoriamente, mas o tamanho do cromossomo seria diminuído. Todas as aproximações foram descritas por Ishii et al. (2004). Nós utilizamos a terceira abordagem, que é a união das duas primeiras abordagens, no nosso trabalho.

Na metodologia proposta neste trabalho, cada solução é codificada por um vetor que é formado na seguinte ordem por:

1. Número de nodos no reservóir;
2. Se há ou não conexão entre a entrada e a saída da rede;
3. Se há ou não conexão entre o bias e a saída;
4. Se há ou não conexão recorrente entre a saída com ela própria;
5. Se há ou não conexão entre o bias e o reservóir;
6. Se há ou não conexão entre saída e o reservóir;
7. Função do nodo (0-tangente hiperbólica ou 1-sigmoidal);
8. Tipo do treinamento (0-Pseudo inversa ou 1-Ridge-regress);
9. Leak rate, que varia entre [0.1;1]
10. Valor do parâmetro de regularização, que varia entre $[10^{-8} ; 10^{-1}]$;
11. É um vetor que tem os valores dos pesos entre as ligações entre reservóir com ela mesma, bias com reservóir e saída com o reservóir. O peso dos valores varia entre [-1; 1].

A população (solução) inicial é gerada aleatoriamente com 120 indivíduos. A nova solução é gerada a partir de uma simulação da reprodução sexuada através dos operadores genéticos dos pais que foram escolhidos através do método de seleção chamado de Método da Amostragem Estocástica Uniforme de acordo com sua avaliação. A avaliação é dada pela equação 1 sendo tão melhor quanto menor for o erro. Este erro é obtido comparando-se a saída obtida com a saída desejada através do MSE (Mean Squared Error – Equação 2)

$$fitness = MSE_{treinamento} + |MSE_{treinamento} - MSE_{validacao}| \quad [Eq. 1]$$

$$MSE = 100 * \left(\frac{L_{MAX} - L_{MIN}}{N * P} \right) \sum_{p=1}^P \sum_{i=1}^N (L_{pi} - T_{pi})^2 \quad [Eq. 2]$$

onde L_{max} e L_{min} são os valores máximo e mínimo das velocidades dos ventos, respectivamente; N é o número de unidades de saída da RNR. P é o número total de padrões na base de dados; L_{pi} e T_{pi} são a saída atual e desejada do i -ésimo neurônio da camada de saída, respectivamente.

O elitismo foi usado com o valor de 2. o que significa que 2 indivíduos dos 120 da população com as melhores avaliações serão escolhidos para sobreviver para a próxima geração. Com o elitismo pode haver perda de diversidade na população muito embora garanta-se que as possíveis melhores características das soluções não sejam perdidas.

4.3 Estudo de Caso

A metodologia descrita acima foi aplicada à busca da melhor ESN para previsão da velocidade média horária dos ventos da cidade de São João do Cariri (SCR). Após 7 gerações, a configuração da melhor ESN escolhida pelo método proposto apresentou a topologia mostrada na Figura 7. Como podemos ver na figura 7, a melhor ESN possui conexões entre bias (B) e reservóir (R), entrada (E) e reservóir, entrada e saída (S), conexão recorrente no reservóir, saída e reservóir e conexão recorrente na camada de saída. Os parâmetros selecionados pelo método são apresentados também na figura 7.

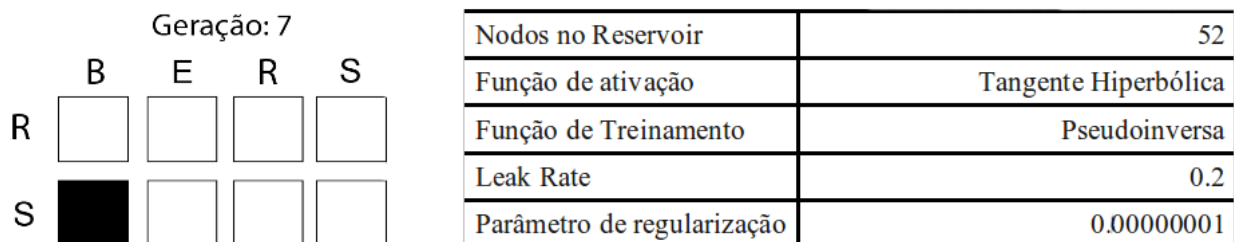


Figura 7 – Configuração da topologia da melhor ESN após 7 gerações na forma de matriz. Quadrado branco significa que há conexão entre as partes correspondentes e quadrado preto não. Parâmetros selecionados pelo método de busca com AG.

A figura 8 apresenta a evolução do melhor fitness ao longo das 7 gerações. Percebe-se que após a geração 4 o algoritmo genético se estabiliza no melhor fitness encontrado 0.0041306. O fitness é baseado no MSE do conjunto de treinamento. Então, quanto menor o fitness, menor o erro desta rede para previsão da série em questão. Ver Eq. 01 e Eq. 02.

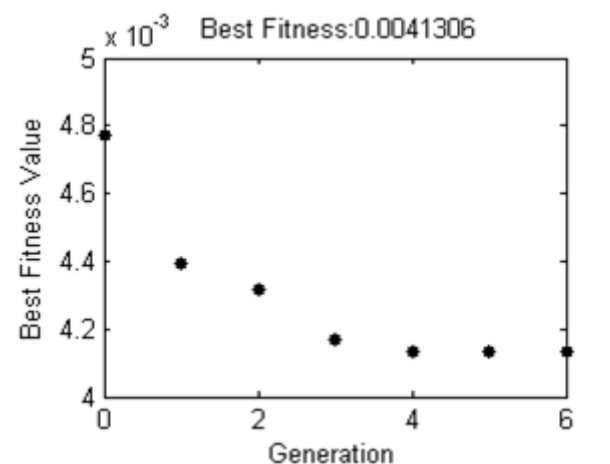


Figura 8 – Evolução do melhor fitness ao longo das gerações.

5. DISCUSSÃO E CONSIDERAÇÕES FINAIS

Para avaliar os resultados foi usado o Cross-validation, que é uma classe de métodos para estimar a taxa de erro verdadeira. Nesses métodos as amostras são divididas em k partições mutuamente exclusivas. A cada iteração uma partição diferente é utilizada para testar o sistema e todas as outras

k-1 partições são utilizadas para treinar o sistema. A taxa de erro é a média das taxas de erro calculadas para as k interações, Ferreira (2004).

Tabela 1 – Erros nas diferentes fases do cross-validation utilizando-se ESN e o erro encontrado com o método da persistência.

	MSE
Treinamento	0,4032
Validação	0,4032
Teste	0,3607
Persistência	1,3545

Analisando-se os dados presentes na tabela 1, observa-se que o método da persistência teve um erro 3,75 vezes maior do que o conjunto de testes utilizando-se ESN otimizado com algoritmos genéticos. Contudo, o método da persistência leva 0.4944 segundos para realizar a previsão enquanto o sistema inteligente híbrido leva cerca de 4.5 segundos. Além do mais, o sistema híbrido levou cerca de 41 minutos para encontrar a melhor rede no treinamento. Conclui-se, pois, que é melhor utilizar um sistema híbrido inteligente para resolver o problema da previsão da velocidade média horária dos ventos do que usar o método da persistência, pois a precisão da previsão foi melhor com a ESN. Outro ponto importante é que se fossemos fazer a busca exaustiva dos parâmetros e topologia da ESN iríamos levar cerca de 53 dias para fazer a busca completa e utilizando algoritmos genéticos encontramos uma solução muito boa em apenas 41 minutos.

6. AGRADECIMENTOS

Agradeço ao CNPq por financiar esse trabalho.

7. REFERÊNCIAS

- BACK, T. **Evolutionary Algorithms in Theory and Practice**, Oxford University Press, 1996.
- FERREIRA, A. **Comparação de arquiteturas de redes neurais para sistemas de reconhecimento de padrões em narizes artificiais**. Dissertação (Mestrado) — (UFPE) Universidade Federal de Pernambuco, 2004.
- ISHII, K.; ZANT, T.; BECANOVIC, V. **Identification of motion with echo state network** 2004 MTS/IEEE. TECHNO- OCEAN 2004 vol. 3, pp. 1205-1210, 2004.
- JAEGER, H. **The echo state approach to analyzing and training recurrent neural networks**, Technical Report GMD 148, German National Resource Center for Information Technology, 2001
- LUKOSERVICIUS, M. e JAEGER, H. **Reservoir computing approaches to recurrent neural network training**. Computer Science Review 3, 2009.
- LINDEN, R. Algoritmos Genéticos - Uma importante ferramenta da Inteligência Computacional. 2. ed. [S.l.: s.n.], 2008.
- MAASS, W., NATSCHLAGER, T. e MAKRAM, H. **Real-time computing without stable states: A new framework for neural computation based on perturbations**, Neural Computation, v. 14, 2002.
- PITTS, W.; MCCULLOCH, W. S.; **A logical calculus of the ideas immanent in nervous activity**. Bulletin of Mathematical Biophysics., v. 5, p. 115–133, 1943.
- RUMELHART E.; HINTON G.; WILLIAMS R.; **Learning representations by back-propagating errors**. Nature, v. 323, p. 533–536, 1986.
- SCHRAUWEN, B., DEFOUR, J., VERSTRAETEN, D. e CAMPENHOUT, J. V. **The introduction of time-scales in reservoir computing, applied to isolated digits recognition**”, 2007, LNCS, 4668, Part I, pp. 471-479.