|  |  |
| --- | --- |
|  | UNIVERSIDADE FEDERAL DE PERNAMBUCO CENTRO DE INFORMÁTICA GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO |

SISTEMA HÍBRIDO BASEADO EM EXTREME LEARNING MACHINE E ALGORITMO GENÉTICO

Romero Teixeira Gonçalves

RECIFE, 19 DE ABRIL de 2013

|  |  |
| --- | --- |
|  | UNIVERSIDADE FEDERAL DE PERNAMBUCO CENTRO DE INFORMÁTICA GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO |

Romero Teixeira Gonçalves

SISTEMA HÍBRIDO BASEADO EM EXTREME LEARNING MACHINE E ALGORITMO GENÉTICO

Trabalho apresentado ao Programa de Graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco como requisito parcial para obtenção do grau de Bacharel em Ciência da Computação.

Orientador: Flávia de Barros Almeida

AGRADECIMENTOS

Gostaria de agradecer inicialmente aos meus pais por terem me criado e educado, solidificando a base para o meu desenvolvimento. Agradeço a eles também, por terem me ensinado o valor do sucesso obtido através de trabalho duro, competência e ética.

Agradeço à professora Flávia de Almeida Barros pela paciência e suporte na realização deste trabalho.

Agradeço também aos amigos pelo apoio, amizade e bons momentos de descontração durante esses quatro anos e meio de graduação.

Por fim, agradeço a todos os professores do curso de Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco, pela boa formação e orientação que recebi.

"Existem muitas hipóteses em ciência que estão erradas. Isso é perfeitamente aceitável, elas são a abertura para achar as que estão certas." (Carl Sagan)

RESUMO

Neste trabalho, foi utilizada uma técnica de busca associada a um algoritmo de treinamento de redes neurais artificiais. O objetivo é desenvolver um sistema híbrido que possa ser utilizado por pessoas de outras áreas de conhecimento. Além de gerar a matriz de pesos, o sistema faz a seleção de variáveis de entrada e monta a arquitetura da rede, selecionando o número de neurônios na camada escondida. O algoritmo de treinamento guia a busca escolhendo os indivíduos que apresentam menor: erro de previsão/classificação, variáveis de entrada e neurônios na camada escondida. No geral, o sistema híbrido obteve desempenho satisfatório, possuindo qualidade de resposta superior ou equivalente aos métodos utilizados por [1] e [2].

**Palavras-chave:** Redes Neurais Artificiais, Seleção de Variáveis, Extreme Learning Machine, Algoritmo Genético, Previsão de Séries Temporais.ABSTRACT

In this work I used a heuristic search algorithm associated to an artificial neural network’s training algorithm. The aim is to develop a hybrid system which can be used by people from other knowledge areas. Besides generating a weight matrix, the system selects the input variables and builds the network's architecture, selecting the number of neurons in the hidden layer. The training algorithm leads the search choosing the individuals which show the minor: forecast/classification error, input variables and neurons in the hidden layer. In general, the hybrid system reached a satisfactory performance, obtaining a better quality of response or equivalent to the methods used by [1] and [2].

**Keywords:** Artificial Neural Network, Variables Selection, Extreme Learning Machine,

Temporal Forecasting Series. SUMÁRIO

[CAPÍTULO 1 - INTRODUÇÃO 1](#_Toc354686188)

[1.1 Motivação e Caracterização do Problema 1](#_Toc354686189)

[1.2 Suposição e Objetivos 2](#_Toc354686190)

[1.2.1 Objetivos Gerais 2](#_Toc354686191)

[1.2.2 Objetivos Específicos 2](#_Toc354686192)

[1.3 Estrutura da Monografia 2](#_Toc354686193)

[CAPÍTULO 2 – REDE NEURAL ARTIFICIAL E EXTREME LEARNING MACHINE 3](#_Toc354686194)

[2.1 Rede Neural Artificial 3](#_Toc354686195)

[2.1.1 Multilayer Perceptron 3](#_Toc354686196)

[2.2 Algoritmo de Treinamento Extreme Learning Machine (ELM) 4](#_Toc354686197)

[2.2.1 Modelagem Matemática da SLFN 4](#_Toc354686198)

[2.2.2 ELM 5](#_Toc354686199)

[2.2.3 Considerações 5](#_Toc354686200)

[CAPÍTULO 3 – ALGORITMO GENÉTICO 6](#_Toc354686201)

[3.1 Definição 6](#_Toc354686202)

[3.2 Estrutura 6](#_Toc354686203)

[3.3 Funcionamento 6](#_Toc354686204)

[3.3.1 Inicialização 6](#_Toc354686205)

[3.3.2 Seleção 6](#_Toc354686206)

[3.3.3 Operadores Genéticos 7](#_Toc354686207)

[3.3.4 Critérios de Parada 7](#_Toc354686208)

[CAPÍTULO 4 – SISTEMA HÍBRIDO PROPOSTO 8](#_Toc354686209)

[4.1 Modelagem do Sistema 8](#_Toc354686210)

[4.2 Formatação dos Operadores 8](#_Toc354686211)

[4.2.1 Funções de Avaliação 9](#_Toc354686212)

[4.3 Funcionamento do Sistema 9](#_Toc354686213)

[CAPÍTULO 5 – EXPERIMENTOS E RESULTADOS 11](#_Toc354686214)

[5.1 Experimentos 11](#_Toc354686215)

[5.1.1 Parte 1 – Validando o Sistema Proposto 11](#_Toc354686216)

[5.2.2 Parte 2 – Teste de Previsão de Séries Temporais 12](#_Toc354686217)

[5.2 Análise dos Resultados 14](#_Toc354686218)

[5.2.1 Análise Parte 1 14](#_Toc354686219)

[5.2.2 Análise Parte 2 18](#_Toc354686220)

[CAPÍTULO 6 – CONCLUSÕES E TRABALHOS FUTUROS 22](#_Toc354686221)

[REFERÊNCIAS BIBLIOGRÁFICAS 23](#_Toc354686222)

LISTA DE FIGURAS

[Figura 1: Arquitetura de uma rede MLP [3] 13](#_Toc354511944)

[Figura 2: Exemplo de representação do vetor de busca 19](#_Toc354511945)

[Figura 3: Gráfico da série temporal de Salto Caxias (quantidade de vazão x tempo) 23](#_Toc354511946)

[Figura 4: Gráfico da série temporal de Furnas (quantidade de vazão x tempo) 23](#_Toc354511947)

[Figura 5: Gráfico boxplot do RMSE para a base Abalone 25](#_Toc354511948)

[Figura 6: Gráfico boxplot do RMSE para a base Delta Ailerons 26](#_Toc354511949)

[Figura 7: Gráfico boxplot do RMSE para a base Triazines 27](#_Toc354511950)

[Figura 8: Gráfico boxplot do RMSE para a base Machine CPU 28](#_Toc354511951)

[Figura 9: Gráfico boxplot do EPMA para a base Salto Caxias 29](#_Toc354511952)

[Figura 10: Gráfico dos resultados da previsão para a base Salto Caxias 29](#_Toc354511953)

[Figura 11: Gráfico boxplot do EPMA para a base Furnas 30](#_Toc354511954)

[Figura 12: Gráfico dos resultados da previsão para a base Furnas 31](#_Toc354511955)

LISTA DE TABELAS

[Tabela 1: Características das bases de dados da Parte 1 21](#_Toc354511963)

[Tabela 2: Configuração das bases de dados da Parte 1 para o algoritmo $ELMpuro$ 22](#_Toc354511964)

[Tabela 3: Configuração das bases de dados da Parte 1 para o algoritmo $ELMGA1$ e $ELMGA2$ 22](#_Toc354511965)

[Tabela 4: Características estatísticas da série temporal Salto Caxias 23](#_Toc354511966)

[Tabela 5: Características estatísticas da série temporal Furnas 23](#_Toc354511967)

[Tabela 6: Configuração das bases de dados da Parte 2 para o algoritmo $ELMGA1$ e $ELMGA2$ 24](#_Toc354511968)

[Tabela 7: Comparação dos resultados para a base Abalone 25](#_Toc354511969)

[Tabela 8: Comparação dos resultados para a base Delta Ailerons 25](#_Toc354511970)

[Tabela 9: Comparação dos resultados para a base Triazines 26](#_Toc354511971)

[Tabela 10: Comparação dos resultados para a base Machine CPU 27](#_Toc354511972)

[Tabela 11: Comparação dos resultados para a base Salto Caxias 29](#_Toc354511973)

[Tabela 12: Comparação dos resultados para a base Furnas 30](#_Toc354511974)

LISTA DE ALGORITMOS

[Algoritmo 1: Pseudocódigo do Extreme Learning Machine 15](#_Toc354512186)

[Algoritmo 2: Pseudocódigo do Sistema Proposto 20](#_Toc354512187)

GLOSSÁRIO

ELM – Extreme Learning Machine

EMQ – Erro Médio Quadrático

EPMA – Erro Percentual Médio Absoluto

GA – Genetic Algorithm

MLP – Multilayer Perceptron

ONS – Operador Nacional do Setor Elétrico

PSE – Predicted Squared Error

RMSE – Root Mean Square Error

RNA – Rede Neural Artificial

SLFNs – Single-hidden-layer Feedforward Networks

SVM – Support Vector Machine

TVB – Tamanho do Vetor de busca

VB – Vetor de busca

VME – Valor máximo de entradas

VMNCE – Valor máximo de neurônios da camada escondida

GA – Geração atual

VMG – Valor máximo de gerações

TP – Tamanho da população

# CAPÍTULO 1 - INTRODUÇÃO

Neste trabalho, foi utilizado o algoritmo genético (técnica de busca e otimização) conjunto com o algoritmo ELM (Extreme Learning Machine) para a seleção de variáveis de entrada e treinamento de uma rede neural artificial. O sistema híbrido proposto monta a arquitetura da rede, selecionando o número de neurônios na camada escondida, a matriz de peso da rede e escolhe as variáveis de entrada.

O capítulo introdutório da monografia está dividido em três seções: a Seção 1.1 mostra a motivação da realização deste trabalho, além de apresentar o problema abordado pelo mesmo. Em seguida, na Seção 1.2 é apresentada a suposição da possível solução do problema e os objetivos deste trabalho. Por fim, na Seção 1.3 a estrutura do documento é descrita.

## Motivação e Caracterização do Problema

Redes neurais MLP (Multilayer Perceptron) têm sido exaustivamente utilizadas em diversos domínios devido à suas habilidades de: (1) aproximar mapeamentos complexos não lineares a partir de amostras de entrada; e (2) prover modelos para uma grande classe de fenômenos naturais e artificiais que são difíceis de lidar utilizando técnicas de parametrização clássica [1].

Em geral o tempo de treinamento de redes neurais MLP baseadas no gradiente descendente é lento além de se ter grande chance de se ficar preso em mínimos locais [1]. Além disto, a maneira como é feita a escolha dos parâmetros da rede acontece de forma empírica, necessitando assim, da intervenção do usuário que, na maioria dos casos, não tem domínio sobre a problemática.

Algoritmos baseados no gradiente descendente são os mais utilizados para o treinamento de redes neurais MLP, sendo determinantes no tempo de treinamento. Nestes casos, todos os parâmetros da rede são ajustados iterativamente, tornando o processo de aquisição de conhecimento lento, podendo durar várias horas ou até mesmo dias [1].

O algoritmo mais utilizado para o treinamento de redes MLP é o backpropagation, e o seu sucesso está associado à escolha dos parâmetros da rede, os quais são:

* As variáveis de entrada, que representam as características do domínio em questão. Deve-se optar por aquelas que possuem maior relevância ao problema [4];
* O número de neurônios na camada escondida, seu valor é arbitrado. Os valores menores são os mais apropriados, pois tornam a arquitetura da rede mais compacta levando a um menor tempo de treinamento;
* O parâmetro taxa de aprendizado, que busca aproximar a solução atual ao mínimo global. Seu valor tem influência no passo do ajuste dos pesos da rede;
* O parâmetro momento, que tem por objetivo aumentar o passo de ajuste dos pesos sem a necessidade de aumentar a taxa de aprendizado, bem como reduzir o risco de se ficar mais facilmente preso a mínimos locais, aumentando a velocidade de convergência.

Diante deste contexto, se faz necessária a busca por soluções alternativas, que proporcionem um menor tempo de treinamento das redes MLP e que diminuam a necessidade de interferência do usuário nos ajustes dos parâmetros de configuração da rede.

## Suposição e Objetivos

A hibridização de algoritmos de treinamento que consomem menor tempo e métodos de busca e otimizaçao pode ser uma boa solução para resolver os problemas de tempo de treinamento e seleção dos parâmetros de configuração das redes neurais MLP.

### Objetivos Gerais

Desenvolver um algoritmo híbrido de treinamento de uma rede neural MLP. O algoritmo deve ter um tempo de treinamento menor que os baseados em gradiente descendente, além de ser capaz de selecionar as melhores combinações das variáveis de entrada - podendo representar uma porção menor da quantidade de variáveis de entrada originais - e os parâmetros de configuração da rede neural, sem diferenças significativas na qualidade da aprendizagem.

Para a construção do algoritmo proposto, será utilizado o algoritmo de treinamento, ELM [1] em conjunto com a técnica de busca e otimização Algoritmo Genético [5].

### Objetivos Específicos

* Implementar e validar o algoritmo de treinamento ELM;
* Implementar e validar o algoritmo genético;
* Integrar o algoritmo genético com o algoritmo de treinamento ELM;
* Submeter o algoritmo criado a conjuntos de benchmarks;
* Comparar os resultados obtidos com os resultados encontrados em outros trabalhos.

## 1.3 Estrutura da Monografia

Está organizada em seis capítulos. No Capítulo 2, serão abordados aspectos teóricos referentes às Redes Neurais Artificiais (RNA) e ao algoritmo de treinamento ELM. Em seguida, no Capítulo 3, serão abordados os aspectos teóricos referentes ao algoritmo genético. O Capítulo 4 descreve o Sistema Híbrido proposto, incluindo o seu funcionamento. No Capítulo 5, será feita uma discussão acerca dos experimentos, bem como a análise dos resultados. Por fim, no Capítulo 6, serão expostas as conclusões e as proposições dos trabalhos futuros.

# CAPÍTULO 2 – REDE NEURAL ARTIFICIAL E EXTREME LEARNING MACHINE

Neste capítulo serão abordados os aspectos teóricos referentes às Redes Neurais Artificiais e ao algoritmo de treinamento Extreme Learning Machine. Na seção 2.1 a rede neural artificial será definida e caracterizada. Em seguida, na seção 2.2 será exposto o algoritmo de treinamento Extreme Learning Machine.

## 2.1 Rede Neural Artificial

É um modelo matemático baseado na estrutura neural dos seres humanos e que adquire conhecimento através da experiência. Em outras palavras, é um aproximador universal de funções, que mapeia conjuntos de dados de entrada para uma saída apropriada através de um aprendizado por experiência, onde seus pesos são ajustados à medida que são fornecidos novos conjuntos de entrada.

### 2.1.1 Multilayer Perceptron

Uma MLP é uma RNA do tipo feedforward (conexões entre unidades não forma um ciclo dirigido) que possui várias camadas de nós em um grafo direcionado, com cada elemento de uma camada ligado à próxima (Figura 1). Cada nó representa um neurônio (ou unidade de processamento) com uma função de ativação não-linear.

Uma das características mais importantes das redes MLP está na sua capacidade de poder aproximar qualquer função contínua arbitrária utilizando uma única camada escondida e qualquer função diferenciável como função de ativação [6].



Figura 1: Arquitetura de uma rede MLP [3]

## 2.2 Algoritmo de Treinamento Extreme Learning Machine (ELM)

Algoritmo de treinamento para redes SLFNs (single hidden layer feedforward networks) - redes MLP com uma camada escondida - proposto por [1]. Considerada a rede como um sistema linear, onde os pesos da camada de saída (ligam a camada escondida à camada de saída) podem ser analiticamente (i.e., não há ajuste iterativo dos parâmetros da rede) determinados através do cálculo da inversa generalizada da matriz de saída da camada escondida.

### 2.2.1 Modelagem Matemática da SLFN

Para $N$ amostras arbitrárias distintas $(x\_{i}, t\_{i})$, onde $x\_{i}=[x\_{i1}, x\_{i2}, …, x\_{in}]^{T} \in R^{n}$ é o vetor de entradas e $t\_{i}=[t\_{i1}, t\_{i2}, …, t\_{in}]^{T} \in R^{n}$ é o vetor de saída, SLFNs com $\tilde{N}$ sendo a quantidade de nós na camada escondida e função de ativação $g(x)$ são matematicamente modeladas como:

$$\sum\_{i=1}^{\tilde{N}}β\_{i}g\_{i}(x\_{j})= \sum\_{i=1}^{\tilde{N}}β\_{i}g\_{i}(w\_{i}.x\_{j}+b\_{i})= o\_{j}, j=1, …, N (2.1)$$

Onde $w\_{i}=[w\_{i1}, w\_{i2}, …, w\_{in}]^{T}$ é o vetor de pesos que conecta o i-ésimo nó da camada escondida à camada de entrada, $β\_{i}=[β\_{i1}, β\_{i2}, …, β\_{in}]^{T}$ é o vetor de pesos que conecta o i-ésimo nó da camada escondida à camada de saída e $b\_{i}$ é o bias do i-ésimo nó da camada escondida. O termo $w\_{i}.x\_{j}$ denota o produto interno entre $w\_{i}$ e $x\_{j}$.

Esta SLFN com $\tilde{N}$ nós na camada escondida e função de ativação $g(x)$ pode aproximar estas $N$ amostras com erro zero. Isto significa que $\sum\_{i=1}^{\tilde{N}}| o\_{j}- t\_{j }|=0$, isto é, existe $β\_{i}$, $w\_{i}$ e $b\_{i}$ tal que:

$$\sum\_{i=1}^{\tilde{N}}β\_{i}g\_{i}(w\_{i}.x\_{j}+b\_{i})= t\_{j}, j=1, …, N (2.2)$$

As equações acima podem ser escritas de forma compacta como:

$$Hβ=T (2.3)$$

Onde:

$$H\left(w\_{1},…,w\_{n},b\_{1},…,b\_{\tilde{N}},x\_{1},…,x\_{N}\right)= \left[\begin{matrix}g(w\_{1}.x\_{1}+b\_{1})&\cdots &g(w\_{\tilde{N}}.x\_{1}+b\_{\tilde{N}})\\\vdots &…&\vdots \\g(w\_{1}.x\_{N}+b\_{1})&\cdots &g(w\_{\tilde{N}}.x\_{N}+b\_{\tilde{N}})\end{matrix}\right] (2.4)$$

$β=\left[\begin{array}{c}β\_{1}^{T}\\\vdots \\β\_{\tilde{N}}^{T}\end{array}\right]\_{\tilde{N}×m}$ e $T=\left[\begin{array}{c}t\_{1}^{T}\\\vdots \\t\_{N}^{T}\end{array}\right]\_{N×m} (2.5)$

$H$ é chamada de matriz de saída da camada escondida; a i-ésima coluna de $H$ é o i-ésimo nó de saída da camada escondida com relação as entradas $x\_{1,},x\_{2},…,x\_{n}$.

### 2.2.2 ELM

O treinamento consiste em encontrar uma solução $\hat{β}$ do sistema linear da Equação 2.3 utilizando o método dos mínimos quadráticos:

$$\left|H\left(w\_{1},…,w\_{\tilde{N}},b\_{1},…,b\_{\tilde{N}}\right)\hat{β}-T\right|=min\_{β}\left|H\left(w\_{1},…,w\_{\tilde{N}},b\_{1},…,b\_{\tilde{N}}\right)β-T\right| (2.6)$$

Se a quantidade de nós escondidos $\tilde{N}$ é igual à quantidade de amostras de treinamento $N, \tilde{N}=N$, a matriz $H$ é quadrada e inversível. Quando os pesos de entrada $w\_{i}$ e bias $b\_{i}$ são escolhidos aleatoriamente, SLFNs podem aproximar as amostras de treinamento com erro zero.

Na maioria dos casos o número de nós escondidos é muito menor do que o de amostras de treinamento, $\tilde{N} \ll N$, logo $H$ não é uma matriz quadrada não existindo $w\_{i}$, $b\_{i}$, $β\_{i}$ $(i=1,…,\tilde{N})$ tal que $Hβ=T$. De acordo com [1] a menor solução de mínimos quadráticos para o sistema é:

$$\hat{β}=H^{†}T (2.7)$$

Onde $H^{†}$ é a inversa generalizada de Moore-Penrose da matriz $H$. Dessa forma, o ELM funciona de acordo com o pseudocódigo apresentado no Algoritmo 1.



Algoritmo 1: Pseudocódigo do Extreme Learning Machine

### 2.2.3 Considerações

Em [1] o desempenho do ELM foi comparado com outros algoritmos de treinamento de SLFNs, como o convencional backpropagation e SVM (Support Vector Machine). Em suma, o algoritmo ELM mostrou-se dezenas - em alguns casos centenas - de vezes mais rápido do que os outros algoritmos citados, e obteve valores semelhantes em relação as métricas de avaliação de desempenho de redes neurais, como Erro Médio Quadrático (EMQ) e seu Desvio-padrão.

# CAPÍTULO 3 – ALGORITMO GENÉTICO

Neste capítulo serão abordados os aspectos teóricos referentes ao algoritmo genético. Na seção 3.1 é feita uma definição do algoritmo. Em seguida, na seção 3.2 é exposta sua estrutura. Na seção 3.3 é feito um detalhamento de seu funcionamento.

## 3.1 Definição

 É um algoritmo de busca e otimização que utiliza técnicas baseadas na evolução natural, como herança, seleção, mutação e crossover.

## 3.2 Estrutura

 No algoritmo genético, uma população indivíduos (soluções para um determinado problema de otimização) é evoluída para a obtenção de soluções melhores. Cada indivíduo possui um conjunto de propriedades (genes) que pode sofrer mutações e alterações. Esses genes são normalmente representados por sequências binárias, mas dependendo do problema podem ser representados de outras formas, como números inteiros e símbolos repetidos (para problemas de cálculo e agrupamento, respectivamente).

 O algoritmo genético requer basicamente uma representação genética do domínio da solução, uma função $fitness$ para avaliar as soluções e operadores genéticos para gerar novas soluções a partir das soluções atuais.

## 3.3 Funcionamento

 O funcionamento do algoritmo genético consiste em 4 passos: inicialização, seleção, operadores genéticos e critérios de parada.

### 3.3.1 Inicialização

 Inicialmente os indivíduos são gerados aleatoriamente para gerar uma população inicial com um determinado tamanho $TP$ (tamanho da população). Esse tamanho depende da natureza do problema, e pode chegar a centenas ou milhares de indivíduos. Geralmente a população inicial é gerada de forma aleatória, para permitir que se tenha uma grande variedade de entradas, para aumentar o espaço de busca (fazendo com que as áreas de soluções ótimas sejam atingíveis).

### 3.3.2 Seleção

 A cada geração de um novo conjunto de indivíduos, uma porção da geração atual ($GA$) é selecionada para gerar o novo conjunto. Esses indivíduos são selecionados através de um processo baseado no $fitness$, isto é, baseado em quão aptos esses indivíduos são para a resolução do problema. Os indivíduos mais aptos devem possuir maior probabilidade de serem escolhidos para gerar a próxima geração. Alguns métodos selecionam aqueles que são mais aptos, de forma a chegar mais facilmente nas soluções ótimas, já outros escolhem aleatoriamente, pelo fato de tal avaliação para escolher os mais aptos ser uma tarefa que pode consumir bastante tempo.

 O método de seleção utilizado neste projeto foi o método da roleta. Neste método cada indivíduo é representado pela fatia de um todo, e quanto mais apto o indivíduo, mais o pedaço da fatia. E então é selecionado aleatoriamente um pedaço deste todo, de forma que quanto mais apto o indivíduo, maior a chance de ele ser escolhido para gerar a próxima geração. O método da roleta, embora tenha a tendência de escolher os indivíduos mais aptos, ainda deixa a chance de que indivíduos menos aptos sejam escolhidos para gerar a próxima geração; e isso é uma vantagem, pois embora a solução seja fraca, ela pode conter algum componente que possa ser útil no processo de recombinação dos indivíduos.

### 3.3.3 Operadores Genéticos

 O próximo passo é o de gerar a próxima geração de indivíduos escolhidos através do método de seleção. Essa geração é feita através dos operadores genéticos crossover (recombinação) e/ou mutação. Para cada nova solução a ser produzido, um par (pais) de indivíduos da população atual é selecionado. Esse “filho” produzido compartilha das características dos pais, mas é diferente deles. Novos pais são selecionados para cada novo filho, e o processo continua até que uma população do tamanho desejado seja gerada.

 O crossover é um método que, dados dois pais, suas soluções são combinadas de forma a criar uma única solução, podendo-se, por exemplo, definir um ponto de corte aleatório na solução (sequência de bits) e para um lado desse ponto usar as características de um pai e para o outro lado usar as características do outro pai.

A mutação é um método que troca aleatoriamente alguma característica do pai para gerar um novo indivíduo. Para um indivíduo que é representado por uma sequência de bits, por exemplo, pode-se inverter aleatoriamente um desses bits para gerar o novo indivíduo.

### 3.3.4 Critérios de Parada

 O processo de gerações de novos indivíduos é repetido até que um critério de parada seja atingido. Critérios de parada comuns são:

* Valor máximo de gerações ($VMG$);
* Novas iterações não produzem resultados melhores que as anteriores;
* Um determinado período de tempo;
* Uma solução é encontrada que satisfaz um critério mínimo.

# CAPÍTULO 4 – SISTEMA HÍBRIDO PROPOSTO

O sistema proposto visa simplificar a utilização de redes neurais por usuários de outras áreas de conhecimento, dada que, a ausência de sensibilidade por parte desses usuários durante a escolha dos parâmetros de configuração da rede neural torna o uso da técnica, em aplicações corriqueiras, extremamente difícil, senão inviável.

Na seção 4.1 é discutida a modelagem do sistema. Em seguida, na seção 4.2 é exposta a formatação dos operadores. Por fim, na seção 4.3 é feito um detalhamento de seu funcionamento.

## 4.1 Modelagem do Sistema

Este sistema utiliza o algoritmo genético e ELM em conjunto. O primeiro faz a seleção das variáveis de entrada e escolhe o número de neurônios na camada escondida. O segundo, treina a rede neural com os parâmetros gerados pelo algoritmo genético. Dessa forma, as métricas de avaliação do treinamento da rede guiam a avaliação da busca, onde os melhores indivíduos são aqueles que possuem um menor número de variáveis de entrada e de neurônios na camada escondida sem apresentar diferenças significativas na qualidade da aprendizagem.

## 4.2 Formatação dos Operadores

Os operadores utilizados pelo algoritmo genético e ELM foram ajustados e sua descrição pode ser vista a seguir:

* $VMNCE$ - Valor máximo de neurônios da camada escondida. Representa o número máximo de neurônios que a camada escondida da RNA poderá assumir.
* $VME$ - Valor máximo de entradas. Representa o número máximo de entradas da RNA, sendo igual ao número de variáveis de entrada do conjunto de dados a serem selecionadas.
* $TVB$ - Tamanho do Vetor de busca, corresponde ao tamanho do vetor de busca, seu valor é igual à soma do $VMNCE$ com o $VME$, $TVB=VMNCE+VME$.
* $VB$ - Vetor de busca. Representa o indivíduo. Foi dividido em duas partes, onde a primeira indica as variáveis de entrada selecionadas (indicadas pelo índice dos elementos que tiverem valor um), e a segunda, o número de neurônios na camada escondida que serão utilizados no treinamento da rede neural (sendo igual à quantidade de elementos de valor um).

Um exemplo de um VB pode ser visto na Figura 2, onde o VME = 4; VMNCE = 6; as variáveis de entrada selecionadas para o treinamento da rede foram as de índice um, dois e três; e o número de neurônios da camada escondida igual a três.



Figura 2: Exemplo de representação do vetor de busca

### 4.2.1 Funções de Avaliação

Neste trabalho foram utilizadas duas funções de avaliação, a primeira é uma variação do critério PSE (Predicted Squared Error) [7] e está descrita na Equação 4.1.

$$f\_{1}\left(x\right)= EMQ\_{val}+2σ\_{ε}^{2}\frac{p}{N}+\frac{η}{VMNCE} (4.1)$$

Onde $EMQ\_{val}$ é o erro médio quadrático do conjunto de validação; $σ\_{ε}^{2}$ é a variância do$EMQ\_{val}$; p é o número de variáveis de entrada selecionadas para o treinamento; $N$ o número de exemplos do conjunto de validação; e $η$ é o número de neurônios da camada escondida selecionados para o treinamento.

A segunda foi criada segundo uma análise empírica das variáveis do sistema levando-se em conta as variáveis que são o foco da otimização. Sua estrutura pode ser vista na Equação 4.2.

$$f\_{2}\left(x\right)= EMQ\_{val}+\frac{p}{VME}+\frac{η}{VMNCE} (4.2)$$

## 4.3 Funcionamento do Sistema

O sistema proposto treina uma rede neural utilizando o vetor de busca para configurar os parâmetros da RNA, onde o vetor $ζ$ que representa o melhor conjunto de configurações da rede neural, e as métricas de avaliação aplicadas a rede guiam a construção dos novos indivíduos.

Seu funcionamento é dividido nos seguintes passos:

1. Inicializar os parâmetros de configuração do sistema:
* Carregar as bases de dados de treinamento, validação e teste;
* Atribuir o valor da variável $VME$ a partir da base de dados;
* Definir o valor da variável $VMNCE$;
* Definir o valor máximo de iterações do sistema;
* Escolher o tamanho da população ($TA$);
* Escolher os valores das taxas de crossover e mutação;
* Definir a função de avaliação do sistema;
* Definir a função de ativação da rede neural;
1. Gerar vetores para o preenchimento da $GA$. Na geração da segunda parte do vetor distribuir os valores de forma a abranger todas as faixas de valor da variável $VMNCE$. Por exemplo, supondo que o $VMNCE=100$ e $TP=5$, as faixas de valores são as seguintes 1-20, 20-40, 40-60, 60-80, 80-100.
2. Gerar uma nova geração de vetores $X^{'}$ para substituir $GA$:
* O melhor indivíduo da $GA$ é mantido na nova geração;
* Os novos indivíduos são obtidos através de crossover e mutação dos indivíduos da $GA$ com probabilidade de 80% e 5% respectivamente;
1. A partir da configuração que cada $X^{'}$ representa treinar uma RNA utilizando o algoritmo ELM e a base de dados de treinamento. Com os pesos do treinamento, gerar o $EMQ\_{val}$ (utilizando a base de dados de validação).

Os passos 3 e 4 são repetidos até que o critério de parada seja alcançado (máximo de iterações ou valor médio do fitness dos vetores da $GA$ for igual ou superior a 90% do valor do fitness do melhor vetor). O pseudocódigo do Sistema proposto está representado no Algoritmo 3.



Algoritmo 2: Pseudocódigo do Sistema Proposto

CAPÍTULO 5 – EXPERIMENTOS E RESULTADOS

Neste capítulo serão abordados os experimentos e a análise dos resultados alcançados. Na seção 5.1 será exposta a metodologia utilizada na condução dos experimentos. Em seguida, na seção 5.2 é feita uma análise dos resultados obtidos, comparando-os com outros trabalhos.

## 5.1 Experimentos

Está seção está dividida em duas partes: Na primeira foram utilizadas algumas das bases de dados descritas por [1], a fim de validar e comparar o desempenho do sistema proposto com a versão do artigo; na segunda foram utilizadas bases de dados do histórico da vazão diária de barragens brasileiras, visando comparar seus resultados com os fornecidos pelo Operador Nacional do Sistema Elétrico (ONS) [2] [8].

### 5.1.1 Parte 1 – Validando o Sistema Proposto

Tendo como objetivo validar e comparar o desempenho do sistema proposto, foram coletadas quatro bases utilizadas por [1] em seu trabalho. As bases aqui utilizadas foram extraídas de [9] e suas características podem ser vistas na Tabela 1. A seguir é feita uma breve descrição de cada uma dessas bases.

**Abalone** - Este conjunto de dados pode ser usado para obter um modelo para prever a idade de abalone a partir de medições físicas. A idade é determinada pelo corte da concha, para análise de sua estrutura interna, onde é feita a contagem do número de anéis internos através de um microscópio. Outras medidas mais fáceis de obter são utilizadas para prever o número de anéis que determina a idade.

**Delta Ailerons** - Este conjunto de dados aborda um problema de controle, mais especificamente pilotar uma aeronave F16. Os atributos descrevem o estado do avião, enquanto o objetivo é prever a ação de controle sobre os ailerons da aeronave.

**Triazines** - Este conjunto de dados é utilizado para aprender uma equação de regressão, regra ou árvore para prever a atividade dos atributos descritivos estruturais. Mais detalhes podem ser vistos em [10].

**Machine CPU** - Este conjunto de dados diz respeito à informações de desempenho de uma CPU de computador.

Tabela 1: Características das bases de dados da Parte 1

|  |  |  |
| --- | --- | --- |
| # Base de dados | # Observações | # Atributos |
|  | Total | Contínuo | Nominal |
| Abalone | 4177 | 7 | 1 |
| Delta Ailerons | 7129 | 6 | 0 |
| Triazines | 186 | 60 | 0 |
| Machine CPU | 209 | 6 | 0 |

Todas as bases tiveram suas variáveis de entrada normalizadas entre $[0, 1]$ e variáveis de saída (alvo) normalizadas entre $[-1, 1]$ e foram executadas no algoritmo ELM desenvolvido em MATLAB por [1] disponível em [11]. Neste trabalho este algoritmo será chamado de $ELM\_{puro}$. Estas bases de dados também foram utilizadas para execução do sistema proposto para as duas funções de avaliação citadas no Capítulo 4. O algoritmo proposto será chamado de $ELM\_{AG1}$ quando estiver utilizando a função de avaliação da Equação 4.1 e será chamado de $ELM\_{AG2}$ quando utilizar a função de avaliação da Equação 4.2. Nos três algoritmos a função de ativação da rede neural escolhida foi a Sigmoide Logística. Mais informações sobre a função podem ser vistos em [6].

Para o $ELM\_{puro}$ foram utilizados os mesmos parâmetros de configuração citados em [1], tais parâmetros podem ser vistos na Tabela 2 e estão descritos a seguir.

**Observações** - Divisão da base de dados em quantidade de Treinamento e Validação. **Neurônios** - Quantidade de neurônios na camada escondida.

Tabela 2: Configuração das bases de dados da Parte 1 para o algoritmo $ELM\_{puro}$

|  |  |  |
| --- | --- | --- |
| # Base de dados | # Observações | # Neurônios |
|  | Treinamento | Validação |  |
| Abalone | 2000 | 2177 | 25 |
| Delta Ailerons | 3000 | 4129 | 45 |
| Triazines | 100 | 86 | 10 |
| Machine CPU | 100 | 109 | 10 |

Para o $ELM\_{GA1}$ e $ELM\_{GA2}$ foram utilizados os parâmetros que se encontram na Tabela

3. Com o objetivo de avaliar o sistema proposto (após as bases selecionadas já terem sofrido algum tipo de pré-processamento), nesta etapa dos experimentos não será feita a seleção de variáveis de entrada, ou seja, ficará a cargo do sistema proposto treinar a rede selecionando apenas o número de neurônios da camada escondida. A fim de gerar amostras para testes estatísticos, cada base foi executada cinquenta e uma vezes em cada algoritmo.

Os parâmetros de configuração dos algoritmos $ELM\_{GA1}$ e $ELM\_{GA2}$ são os seguintes:

**Máximo neurônios** - Quantidade máxima de neurônios na camada escondida que a rede neural pode assumir.

**Máximo iterações** - O número máximo de iterações do algoritmo.

Tabela 3: Configuração das bases de dados da Parte 1 para o algoritmo $ELM\_{GA1}$ e $ELM\_{GA2}$

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # Observações | # Máximo neurônios | # Máximo iterações | # TP | # VMG |
| #Treinamento | # Validação |  |  |  |  |
| 75% | 25% | 200 | 200 | 100 | 10 |

### 5.2.2 Parte 2 – Teste de Previsão de Séries Temporais

A fim de medir o desempenho do sistema proposto em cenários com dados reais foram selecionadas bases de dados contendo a vazão média diária em $m^{3} / s$ das barragens de Salto Caxias e Furnas. O objetivo de se utilizar estas séries é que se dispõe dos erros atuais cometidos pelos modelos estatísticos, em uso, disponibilizados pelo ONS.

A previsão de vazões consiste em um dos processos mais complexos por se tratar de um fenômeno não linear.

**Salto Caxias** - Construída no trecho final do Rio Iguaçu, entre os municípios de Capitão Leônidas Marques e Nova Prata do Iguaçu, distantes cerca de 400 quilômetros da capital Curitiba, no estado do Paraná. Esta série representa a vazão diária no período de 31 de dezembro de 1967 a 31 de dezembro de 2007. Na Figura 3 pode-se ver o gráfico da série e na Tabela 4 suas características estatísticas.



Figura 3: Gráfico da série temporal de Salto Caxias (quantidade de vazão x tempo)

Tabela 4: Características estatísticas da série temporal Salto Caxias

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| # Mínimo | # Máximo | # Média | # Variância | # Desvio-padrão | #Observações |
| 123 | 24711 | 1501.295 | 2073945 | 1440.169 | 14611 |

**Furnas** - Está localizada no curso médio do Rio Grande, entre os municípios de São José da Barra e São João Batista do Glória, no estado de Minas Gerais Esta série representa a vazão diária no período de 31 de dezembro de 1967 a 31 de dezembro de 2007. Pode-se ver o gráfico da série na Figura 4 e suas características estatísticas na Tabela 5.



Figura 4: Gráfico da série temporal de Furnas (quantidade de vazão x tempo)

Tabela 5: Características estatísticas da série temporal Furnas

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| # Mínimo | # Máximo | # Média | # Variância | # Desvio-padrão | #Observações |
| 111 | 7497 | 914.2143 | 480893.100 | 693.4645 | 14611 |

Este experimento consiste na previsão da vazão em $m^{3}/s$ de sete dias à frente utilizando como variáveis de entrada para seleção, trinta dias passados. Neste caso as duas bases tiveram suas variáveis de entrada e saída (alvo) normalizadas entre $[0.15, 0.85]$. O sistema treinará a rede e fará a seleção de variáveis de entrada e o número de neurônios da camada escondida.

Com o intuito de gerar amostras para testes estatísticos, cada base foi executada trinta vezes nos algoritmos $ELM\_{GA1}$ e $ELM\_{GA2}$ com os parâmetros da Tabela 6 para ambos os algoritmos a função de ativação da rede neural utilizada foi a Sigmoide Logística.

Tabela 6: Configuração das bases de dados da Parte 2 para o algoritmo $ELM\_{GA1}$ e $ELM\_{GA2}$

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # Observações | # Máximo neurônios | # Máximo iterações | # TP | # VMG |
| #Treinamento | # Validação |  |  |  |  |
| 75% | 25% | 200 | 200 | 100 | 10 |

## 5.2 Análise dos Resultados

Está seção está dividida em duas partes. Na primeira serão analisados os resultados referentes a subseção 5.2.1 comparando-os com os resultados encontrados em [1]. Na segunda, os da subseção 5.2.2 serão comparados com os do ONS.

### 5.2.1 Análise Parte 1

Para estes casos a métrica de avaliação utilizada será o RMSE (Root Mean Square Error), a qual é frequentemente utilizada para medir a diferença entre valores previstos por um modelo e valores efetivamente observados. Sua fórmula está descrita na Equação 5.1.

Dados $O=\left[\begin{array}{c}o\_{1}\\\vdots \\o\_{n}\end{array}\right]$ e $P=\left[\begin{array}{c}p\_{1}\\\vdots \\p\_{n}\end{array}\right]$ onde o conjunto $O$ representa os valores observados e $P$ os valores previstos temos que:

$$RMSE= \sqrt{\frac{\sum\_{i=1}^{n}(o\_{i}- p\_{i})^{2}}{n}} (5.1)$$

Onde $n$ é o número de ocorrências.

Para cada base foi calculada a média do RMSE e seu desvio-padrão dos três algoritmos. De posse destes resultados, é possível calcular o teste t de Student com um nível de confiança de 95%. A distribuição t de Student é uma distribuição de probabilidade teórica, simétrica, campaniforme (em forma de sino), e semelhante à curva normal padrão, porém com caudas mais largas. Uma simulação de t de Student pode gerar valores mais extremos que uma simulação normal [4].

**Abalone** - Para esta base o sistema proposto obteve menor RMSE e desvio-padrão para as duas funções de avaliação, tendo um melhor resultado para $ELM\_{GA2}$. O teste t de Student mostrou com 95% de chance que há diferença estatística entre os modelos $EML\_{puro}$ com $ELM\_{GA1}$ e $ELM\_{puro}$ com $ELM\_{GA2}$. Os resultados podem ser vistos na Tabela 7. Um gráfico boxplot do RMSE dos algoritmos pode ser visto na Figura 5.

Tabela 7: Comparação dos resultados para a base Abalone

|  |  |  |  |
| --- | --- | --- | --- |
|  | $$ELM\_{puro}$$ | $$ELM\_{GA1}$$ | $$ELM\_{GA2}$$ |
| RMSE | 0.151 | 0.145 | **0.144** |
| Desvio-padrão | 0.003 | **0.001** | **0.001** |
| t | - | 13.906 | 15.333 |
| Neurônios escondida | **25** | 67 | 67 |



Figura 5: Gráfico boxplot do RMSE para a base Abalone

**Delta Ailerons** - Para esta base o sistema proposto obteve menor RMSE para as duas funções de avaliação, tendo um melhor resultado para $ELM\_{GA2}$. Porem, o $ELM\_{puro}$ obteve menor desvio-padrão. O teste t de Student mostrou com 95% de chance que há diferença estatística entre os modelos $ELM\_{puro}$ com $ELM\_{GA1}$ e $ELM\_{puro}$ com $ELM\_{GA2}$. Os resultados podem ser vistos na Tabela 8. Um gráfico boxplot do RMSE dos algoritmos pode ser visto na Figura 6.

Tabela 8: Comparação dos resultados para a base Delta Ailerons

|  |  |  |  |
| --- | --- | --- | --- |
|  | $$ELM\_{puro}$$ | $$ELM\_{GA1}$$ | $$ELM\_{GA2}$$ |
| RMSE | 0.080 | 0.078 | **0.077** |
| Desvio-padrão | **0.000** | 0.004 | 0.001 |
| t | - | 2.689 | 19.313 |
| Neurônios escondida | **45** | 74 | 64 |



Figura 6: Gráfico boxplot do RMSE para a base Delta Ailerons

**Triazines** - Para esta base o sistema proposto obteve menor RMSE para as duas funções de avaliação, tendo um melhor resultado para $ELM\_{GA1}$. Porém, o $ELM\_{puro}$ obteve menor desvio-padrão. O teste t de Student mostrou com 95% de chance que há diferença estatística entre os modelos $ELM\_{puro}$ com $ELM\_{GA1}$ e $ELM\_{puro}$ com $ELM\_{GA2}$. Os resultados podem ser vistos na Tabela 9, um gráfico boxplot do RMSE dos algoritmos pode ser visto na Figura 7.

Tabela 9: Comparação dos resultados para a base Triazines

|  |  |  |  |
| --- | --- | --- | --- |
|  | $$ELM\_{puro}$$ | $$ELM\_{GA1}$$ | $$ELM\_{GA2}$$ |
| RMSE | 0.401 | **0.371** | 0.385 |
| Desvio-padrão | **0.021** | 0.026 | 0.028 |
| t | - | 6.44 | 3.33 |
| Neurônios escondida | **10** | 21 | 23 |



Figura 7: Gráfico boxplot do RMSE para a base Triazines

**Machine CPU** - Para esta base o sistema proposto obteve o mesmo valor do RMSE para a função de avaliação $ELM\_{GA1}$. Porém, o $ELM\_{puro}$ obteve menor desvio-padrão. O teste t de Student mostrou com 95% de chance que há diferença estatística entre os modelos $ELM\_{puro}$ com $ELM\_{GA1}$ e $ELM\_{puro}$ com $ELM\_{GA2}$. Os resultados podem ser vistos na Tabela 10, um gráfico boxplot do RMSE dos algoritmos pode ser visto na Figura 8.

Tabela 10: Comparação dos resultados para a base Machine CPU

|  |  |  |  |
| --- | --- | --- | --- |
|  | $$ELM\_{puro}$$ | $$ELM\_{GA1}$$ | $$ELM\_{GA2}$$ |
| RMSE | **0.196** | **0.196** | 0.210 |
| Desvio-padrão | **0.054** | 0.062 | 0.066 |
| t | - | 0.012 | 1.150 |
| Neurônios escondida | **10** | 39 | 34 |



Figura 8: Gráfico boxplot do RMSE para a base Machine CPU

### 5.2.2 Análise Parte 2

Para estes casos a métrica de avaliação utilizada será o $EPMA$ (Erro Percentual Médio Absoluto). Sua fórmula está descrita na Equação 5.2.

Dados $O=\left[\begin{array}{c}o\_{1}\\\vdots \\o\_{n}\end{array}\right]$ e $P=\left[\begin{array}{c}p\_{1}\\\vdots \\p\_{n}\end{array}\right]$ onde o conjunto $O$ representa os valores observados e $P$ os valores previstos temos que:

$$EPMA= \frac{1}{n}\sum\_{i=1}^{n}\frac{(o\_{i}- p\_{i})}{o\_{i}}100 (5.2)$$

Onde $n$ é o número total de ocorrências.

Para cada base foi calculada a média do $EPMA$ e seu desvio-padrão para $ELM\_{GA1}$ e $ELM\_{GA2}$. De posse destes resultados, é possível calcular o teste t de Student com um nível de confiança de 95%. Além disto, também foi calculada a média da quantidade de neurônios na camada escondida e a média de variáveis de entrada selecionadas pelo sistema.

**Salto Caxias** - Para esta base o sistema proposto obteve menor $EPMA$ para as duas funções de avaliação, tendo um melhor resultado para $ELM\_{GA1}$. O menor desvio-padrão foi alcançado por $ELM\_{GA1}$. O t de Student mostrou com 95% de chance que há diferença estatística entre os modelos $ELM\_{GA1}$ com $ELM\_{GA2}$. Os resultados podem ser vistos na Tabela 11. Um gráfico boxplot do EPMA dos algoritmos pode ser visto na Figura 9 e um gráfico dos resultados da previsão para o período de 01/01/2007 a 21/11/2007 pode ser visto na Figura 10.

Tabela 11: Comparação dos resultados para a base Salto Caxias

|  |  |  |  |
| --- | --- | --- | --- |
|  | $$ONS$$ | $$ELM\_{GA1}$$ | $$ELM\_{GA2}$$ |
| EPMA | 33.600% | **9.181%** | 11.396% |
| Desvio-padrão | - | **0.243%** | 3.594% |
| t | - | - | 3.426 |
| Neurônios escondida | - | 138 | **13** |
| Entradas | - | **21** | **21** |



Figura 9: Gráfico boxplot do EPMA para a base Salto Caxias



Figura 10: Gráfico dos resultados da previsão para a base Salto Caxias

**Furnas** - Para esta base o sistema proposto obteve menor $EPMA$ para as duas funções de avaliação, tendo um melhor resultado para $ELM\_{GA1}$. O menor desvio-padrão foi alcançado por $ELM\_{GA1}$. O t de Student mostrou com 95% de chance que há diferença estatística entre os modelos $ELM\_{GA1}$ com $ELM\_{GA2}$. Os resultados podem ser vistos na Tabela 12. Um gráfico boxplot do EPMA dos algoritmos pode ser visto na Figura 11 e um gráfico dos resultados da previsão para o período de 01/01/2007 a

25/11/2007 pode ser visto na Figura 12.

Tabela 12: Comparação dos resultados para a base Furnas

|  |  |  |  |
| --- | --- | --- | --- |
|  | $$ONS$$ | $$ELM\_{GA1}$$ | $$ELM\_{GA2}$$ |
| EPMA | 27.600% | **10.562%** | 14.906% |
| Desvio-padrão | - | **0.241%** | 4.231% |
| t | - | - | 5.712 |
| Neurônios escondida | - | 150 | **12** |
| Entradas | - | **21** | 23 |

 

Figura 11: Gráfico boxplot do EPMA para a base Furnas

 

Figura 12: Gráfico dos resultados da previsão para a base Furnas

# CAPÍTULO 6 – CONCLUSÕES E TRABALHOS FUTUROS

Neste trabalho foi proposto um sistema híbrido para o treinamento e seleção das variáveis de entrada de uma rede neural artificial MLP com uma camada escondida. Experimentos mostraram que há diferenças estatísticas entre o sistema proposto e [1] e os dados fornecidos pelo ONS. A partir deste trabalho será possível desenvolver uma ferramenta capaz de auxiliar usuários de outras áreas de conhecimento na utilização de redes neurais artificiais, através da qual é possível obter menos parâmetros de configuração.

No geral, a função de avaliação $ELM\_{GA1}$ mostrou-se mais eficiente em relação a $ELM\_{GA2}$, por gerar modelos com menor erro e desvio. No entanto, gerou arquiteturas de redes neurais maiores, por escolher um maior número de neurônios na camada escondida.

Em trabalhos futuros pretende-se criar novas funções de avaliação para o sistema, que melhorem o foco na busca da quantidade de neurônios na camada escondida, a fim de tornar o tempo de treinamento e o tamanho da rede neural gerada mais uniforme.

Além disto, pretende-se adaptar a busca atual para levar em conta o coeficiente de correlação entre as variáveis de entrada na geração de novos indivíduos, a fim de aperfeiçoar a escolha das variáveis de entrada.

Por fim, outras técnicas de otimização podem ser testadas para comparar os resultados obtidos entre elas. Desta forma, a ferramenta terá uma maior robustez, já que cada técnica tem um comportamento diferente, podendo resolver de forma mais eficiente um determinado tipo problema do que outro.

# REFERÊNCIAS BIBLIOGRÁFICAS

[1] HUAN, Q.-Y. Z. G.-B.; SIEW, C.-K. Extreme learning machine: Theory and applications. Science Direct, Neurocomputing, v. 70, p. 489\_501, 2006.

[2] ELÉTRICO, O. N. do S. Site do Operador Nacional do Sistema Elétrico. 2011 (acessado em Abril 19, 2013). Disponível em: <http://www.ons.org.br>.

[3] HAYKIN, S. REDES NEURAIS: Princípios e prática. 2th. ed. [S.l.]: BOOKMAN, 2007.

ISBN 9788573077186.

[4] IVNA, C. Modelos Híbridos Baseados em Redes Neurais, Lógica Fuzzy e Busca para Previsão de Séries Temporais. Dissertação (Mestrado) - Centro de Informática, Universidade Federal de Pernambuco, UFPE, Recife, 2010.

[5] PACHECO, M. A. C. Algoritmos Genéticos: Princípios e Aplicações. (acessado em Abril 19, 2013). Disponível em: <http://www.ica.ele.puc-rio.br/downloads/38/ce-apostila-comp-evol.pdf>

[6] VALENÇA, M. J. S. Aplicando Redes Neurais: Um Guia Completo. Recife, Pernambuco, BR: Editora Livro Rápido - Elógica, 2005.

[7] MOODY, J. Prediction risk and architecture selection for neural networks. In: . [S.l.]:

Springer-Verlag, 1994. p. 147\_165.

[8] ELéTRICO, O. N. do S. I Workshop Previsão de Vazões. 2011 (acessado em Abril 19, 2013). Disponível em: <http://www.ons.org.br/home/previsao\_vazoes.htm>.

[9] UCI Machine Learning Repository (acessado em Abril 02, 2013). Disponível em: <http://archive.ics.uci.edu/ml/ >.

[10] KING, R. D.; HIRST, J. D.; STERNBERG, M. J. E. Comparison of artificial intelligence

methods for modeling pharmaceutical qsars. Applied Artificial Intelligence, v. 9, n. 2, p.

213\_233, 1995.

[11] HUANG, G.-B. MATLAB Codes of ELM Algorithm. 2011 (acessado em Abril 19, 2013). Disponível em: <http://www.ntu.edu.sg/home/egbhuang/elm\_codes.html>.