

## APLICAÇÃO DE ALGORITMOS GENÉTICOS PARA CONFIGURAÇÃO AUTOMÁTICA DE REDES NEURAS ARTIFICIAIS COM MÚLTIPLAS CAMADAS

Inácio Gomes Medeiros<sup>1</sup> e Diego Silveira Costa Nascimento<sup>2</sup>  
E-mail: inacio.medeiros@ifrn.edu.br<sup>1</sup>; diego.nascimento@ifrn.edu.br<sup>2</sup>

### RESUMO

A proposta dessa pesquisa consiste em apresentar uma abordagem híbrida, via Algoritmos Genéticos, para configuração automática dos parâmetros de controle de Redes Neurais Artificiais com múltiplas camadas. Experimentos empíricos foram conduzidos

utilizando oito diferentes problemas de classificação de padrões e, de forma geral, a análise dos resultados aponta benefícios interessantes na escolha automática das configurações do modelo de rede resultante.

**PALAVRAS-CHAVE:** Redes Neurais Artificiais com Múltiplas Camadas; Algoritmos Genéticos; Classificação de Padrões.

## APPLICATION OF GENETIC ALGORITHMS FOR AUTOMATIC CONFIGURATION OF MULTILAYER ARTIFICIAL NEURAL NETWORKS

### ABSTRACT

The proposal of this research consists in presenting a hybrid approach, via Genetic Algorithms, for automatic configuration of control settings of Multilayer Artificial Neural Networks. Empirical

experiments were conducted using eight different pattern classification problems and, in general, results analysis points interesting benefits on automatic choice for neural network settings.

**KEY-WORDS:** Multilayer Artificial Neural Networks, Genetic Algorithms, Pattern Classification.

## 1 INTRODUÇÃO

A cada dia é notável a crescente utilização dos métodos de Aprendizado de Máquina em diferentes domínios de pesquisa. Isso se deve, principalmente, ao fato de novas investigações científicas estarem gerando extensos conjuntos de dados. Esses dados, por sua vez, precisam ser analisados posteriormente, fazendo-se necessária, para tanto, a utilização de técnicas automáticas que auxiliem na interpretação dos resultados finais. Dos algoritmos que recebem maior atenção na literatura, e foco de estudo nessa pesquisa, as Redes Neurais Artificiais são as mais utilizadas.

O estudo de Redes Neurais Artificiais, iniciado por McCulloch e Pitts (1943), surgiu como uma tentativa computacional de tornar possível a construção de sistemas capazes de adquirir conhecimento de forma autônoma através da experiência, ou seja, aprendendo, errando e fazendo descobertas. Essa ideia, no entanto, é inspirada no modelo de estrutura neural de organismos inteligentes. As redes neurais podem ser aplicadas na resolução de diversos problemas práticos, tais como:

- Aproximação de função;
- Previsões de séries temporais; e
- Reconhecimento de padrões.

A proposta deste trabalho está em investigar e combinar Algoritmos Genéticos para construção e configuração automática dos parâmetros de controle de Redes Neurais Artificiais com múltiplas camadas. A principal contribuição, no entanto, é idealizar o conceito de uma ferramenta capaz de, para diferentes tipos de problemas de classificação, recomendar ao usuário uma melhor configuração de uma rede para resolver um novo problema em questão.

Este documento está organizado em seis seções, incluindo esta. A Seção 2 apresenta parte da fundamentação teórica necessária para o entendimento de Redes Neurais artificiais com múltiplas camadas e, de igual, a Seção 3 faz uma apresentação em relação a Algoritmos Genéticos. Na Seção 4, é apresentada a abordagem evolutiva proposta para configuração automática de redes neurais com múltiplas camadas. Na Seção 5, é realizada uma análise em relação aos resultados obtidos. Por fim, na Seção 6, são feitas as considerações finais da pesquisa.

## 2 FUNDAMENTOS TEÓRICOS

### 2.1 Redes neurais

Podemos definir uma rede neural artificial, conforme encontrado em Faceli *et al* (2011), como um sistema computacional distribuído composto por unidades de processamento densamente interconectadas, denominadas neurônios artificiais, responsáveis por computarem funções matemáticas. As unidades são dispostas em uma ou mais camadas e interligadas por um grande número de conexões, as quais simulam as sinapses biológicas, e possuem pesos que ponderam a entrada recebida por cada neurônio da rede.

De forma mais abrangente, podemos dizer que a concepção de uma rede neural depende de três elementos fundamentais: neurônio, arquitetura e aprendizagem. O neurônio é a unidade

computacional básica da rede; a arquitetura é a estrutura topológica de como os neurônios são conectados; e a aprendizagem é um processo que adapta a rede de modo a computar uma função desejada.

## 2.2 Redes neurais com múltiplas camadas

Uma importante classe de redes mais conhecidas é aquela representada por *Perceptrons* de Múltiplas Camadas (*Multilayer Perceptron* - MLP). Sua arquitetura é formada por uma camada de entrada, uma ou mais camadas intermediárias e uma camada de saída, sendo todas elas completamente conectadas, vide Figura 1.

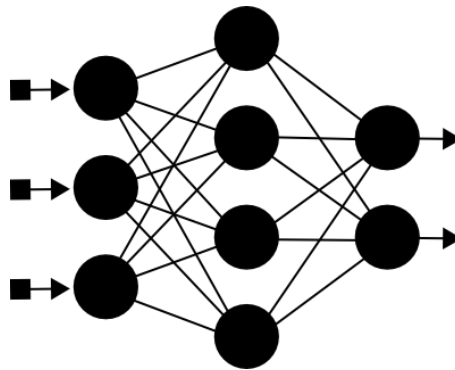


Figura 1: Ilustração de uma rede neural com múltiplas camadas.

O processo de aprendizado de uma rede MLP pode ser descrito em duas fases. Na primeira fase, os valores dos atributos de uma amostra do conjunto de treinamento são inseridos em separados nas entradas da rede e propagados em um único sentido (*feed-forward*), camada a camada, até a produção das respectivas saídas. Os valores dos pesos sinápticos e limiares de seus neurônios permanecem inalterados durante cada execução desta fase. As respostas produzidas pelas saídas da rede são então comparadas aos valores das respostas desejadas. Uma vez cometido erro pela rede, dá-se início à segunda fase, também chamada de propagação reversa (*back-propagation*). Em outras palavras, é realizada uma retropropagação dos erros para realizar os ajustes dos pesos. Esse ajuste, ao contrário da primeira fase, prossegue da camada de saída até a primeira camada intermediária.

Em suma, as aplicações da primeira e segunda fase fazem com que os pesos sinápticos e limiares dos neurônios se ajustem automaticamente a cada iteração, ocasionando uma gradativa diminuição da soma dos erros produzidos pelas respostas da rede frente àquelas desejadas. A segunda fase do treinamento de redes MLP será explanada com maiores detalhes na Seção 3.1.

## 2.3 Algoritmos genéticos

Os Algoritmos evolutivos, dentre os quais os Algoritmos Genéticos (EIBEN; SMITH, 2003), representam uma classe de métodos meta-heurísticos de busca e otimização inspirados nos mecanismos evolutivos naturais, buscando seguir os princípios do Neodarwinismo, que advoga: “quanto melhor um indivíduo se adaptar ao seu meio ambiente, maior será à sua chance de sobreviver e gerar descendentes”.

Os Algoritmos Genéticos trabalham com uma população de indivíduos, denominados cromossomos, os quais geralmente são gerados aleatoriamente, cada qual representando uma possível solução para problema. A cada indivíduo se associa um grau de aptidão (*fitness*) que reflete quão boa é a solução que ele representa para o problema. Os melhores indivíduos têm maior probabilidade de serem selecionados para se reproduzir (*crossover*) e gerar descendentes, enquanto os menos aptos tendem a serem eliminados ao longo das gerações. Durante a reprodução, modificações do material genético dos indivíduos podem acontecer através de mutações nos valores (alelos) dos genes. Todo o processo se repete até que se alcance um número limite de gerações.

### 3 PROBLEMÁTICA DA PESQUISA

As redes neurais artificiais, assim como as árvores de decisões, classificadores baseados em regras e modelos de regressão linear, fazem parte de um grupo de algoritmos de aprendizado de máquina dito de natureza instável (KENNEDY; EBERHART, 1995). São assim classificados devido à alta sensibilidade a pequenas mudanças nos dados de treinamento. Desta forma, o processo de treinamento de uma rede neural acaba sendo tratado como um problema NP-Difícil (BLUM; RIVEST, 1988). O uso da abordagem evolutiva surge, nesse sentido, como uma forma de automatizar esta busca pela melhor configuração possível.

#### 3.1 Treinamento de uma rede neural

Conforme já comentado anteriormente, na segunda etapa de treinamento de uma rede MLP é realizada uma retropropagação dos erros para realizar os ajustes dos pesos das conexões entre os neurônios, iniciando-se na camada de saída e prosseguindo até a primeira camada intermediária. Os ajustes citados previamente utilizam-se da Equação (1).

$$w_{jl}(t + 1) = w_{jl}(t) + \eta x^j \delta_l \quad (1)$$

Em que  $w_{jl}$  representa o peso entre o neurônio  $l$  e o  $j$ -ésimo atributo de entrada (ou a saída do  $j$ -ésimo neurônio da camada anterior) no instante de tempo  $t$ ;  $\delta_l$ , o erro associado ao  $l$ -ésimo neurônio;  $\eta$ , a taxa de aprendizado<sup>1</sup> da rede MLP; e  $x^j$ , a entrada recebida pelo neurônio (ou a saída do  $j$ -ésimo neurônio da camada anterior).

É importante frisar que o  $\delta_l$  é conhecido apenas nos neurônios da camada de saída. Para aqueles que estão nas camadas intermediárias, tal valor é estimado como sendo a soma dos erros dos neurônios da camada seguinte à sua, cujos terminais de entrada (das posteriores) são ponderados pelo peso associado às conexões. Nesse sentido, a fórmula utilizada para calcular tal indicador torna-se fortemente relacionada à camada em que o neurônio está, sendo definida pela Equação (2).

---

<sup>1</sup> Parâmetro que define a magnitude do ajuste realizado no valor de cada peso.



$$\delta_l = \begin{cases} f'_a \epsilon_l, & \text{se } n_l \in c_{sai} \\ f'_a \sum w_{lk} \delta_l, & \text{se } n_l \in c_{int} \end{cases} \quad (2)$$

Em que  $n_l$  é o  $l$ -ésimo neurônio,  $c_{sai}$  e  $c_{int}$ , as camadas de saída e intermediária, respectivamente;  $f'_a$ , a derivada parcial da função de ativação de neurônio, e  $\epsilon_l$ , o erro quadrático cometido pelo neurônio de saída quando sua resposta é comparada à desejada, sendo expresso pela Equação (3).

$$\epsilon_l = \frac{1}{2} \sum_{q=1}^k (y_q - \hat{f}_q)^2 \quad (3)$$

Sendo  $y_q$  a saída desejada, e  $\hat{f}_q$ , a produzida pelo neurônio. Segundo Faceli *et al* (2011),  $f'_a$  é responsável pelo ajuste dos pesos em si, utilizando o gradiente descendente da função de ativação. Essa derivada mede a contribuição de cada peso no erro da rede para a classificação do objeto de entrada.

Ainda de acordo com os autores previamente citados, a taxa de aprendizado  $\eta$  apresenta forte influência na velocidade de convergência da rede. Se for muito pequena, a quantidade de computação necessária para a indução de um bom modelo tornar-se-á grande. O uso de um valor alto para  $\eta$  poderá ocasionar oscilações que dificultam a convergência. Para amenizar tal situação, é utilizado o termo momentum  $\alpha$ , o qual quantifica o grau de importância da variação de peso da etapa de computações anterior à atual. Tal indicador é utilizado no algoritmo back-propagation com termo momentum, sendo os ajustes de pesos definido pela Equação (4).

$$w_{jl}(t+1) = w_{jl}(t) + \eta x^j \delta_l + \alpha(w_{jl}(t) - w_{jl}(t-1)) \quad (4)$$

Em que  $x^j$  representa o valor do  $j$ -ésimo atributo do objeto de entrada (ou a saída do  $j$ -ésimo neurônio da camada anterior).

A partir dos tópicos comentados até então, observa-se que a quantidade de camadas MLP, o número de neurônios em cada camada, a taxa de aprendizado  $\eta$  e o termo momentum  $\alpha$  aparecem como parâmetros fundamentais sobre os quais é necessária reflexão quanto da modelagem de uma rede MLP, haja vista que boa parte destes parâmetros atua diretamente nas equações responsáveis pelo treinamento da rede. O uso de Algoritmos Genéticos neste projeto visa buscar uma configuração mais adequada nos valores de tais parâmetros, de modo que o desempenho alcançado pela rede possa apresentar vantagem quando comparado ao uso da configuração padrão para tais modelos matemáticos.

### 3.2 Utilização de algoritmos genéticos para a configuração de redes neurais

Observamos que o processo de construção de uma rede neural artificial promissora para certo conjunto de dados não é uma tarefa das mais simples, Faceli *et al* (2011) vêm sugerindo a utilização de Algoritmos Genéticos.

Seguindo os princípios comentados na Seção 3, estes geram um conjunto de redes neurais com configurações variadas, passando, posteriormente, a combiná-las entre si, de modo a obter

uma arquitetura que apresente as características mais adequadas para adquirir o melhor desempenho possível quando da indução de um dado problema.

O algoritmo genético utilizado nesta pesquisa para a abordagem híbrida proposta foi o NSGA-II (DEB, 2002), por possuir complexidade computacional relativamente baixa  $O(M \times N^2)$ , em que  $M$  representa o número de objetivos, enquanto  $N$ , o tamanho da população de indivíduos. O NSGA-II, em linhas gerais, é um algoritmo genético com abordagem multiobjetivo, ou seja, é capaz de otimizar em uma única execução mais de uma função objetivo. Em nossos experimentos, foi otimizado apenas o erro médio produzido por uma rede neural MLP a partir da escolha de diferentes parâmetros de controle já mencionados. A Figura 2 ilustra a distribuição dos parâmetros de controle em um cromossomo.

Cada indivíduo é representado por um vetor binário de 46 posições. A primeira posição do cromossomo corresponde à normalização nos valores dos atributos, no qual 1 indica que a normalização ocorrerá, enquanto que 0 denota o contrário. As posições de 2 a 10 correspondem ao número de épocas, o qual possui representação binária, mas que posteriormente é convertido para a representação decimal. Tal parâmetro refere-se aos instantes de tempo  $t$ , os quais estão associados aos pesos nas Equações (1) e (4).

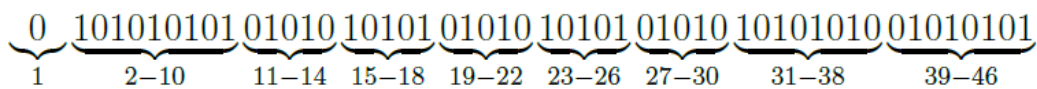


Figura 2: Representação de um cromossomo.

Os conjuntos de posições, de 11 a 14, de 15 a 18, de 19 a 22, de 23 a 26, e de 27 a 30, correspondem às representações em binário do número de neurônios que cada camada intermediária conterá<sup>2</sup>, com a respectiva ordem (o primeiro conjunto de posições corresponde ao número de neurônios da primeira camada, e assim por diante). Os conjuntos de posições de 31 a 38 e de 39 a 46, por fim, correspondem a representações binárias de números reais, em que as duas primeiras posições de cada conjunto fazem referência ao expoente, e o restante, à mantissa. O primeiro conjunto refere-se à taxa de aprendizado da rede neural, e o segundo, à taxa de momentum.

Para finalizar esta seção, explanaremos acerca dos três operadores utilizados pelo algoritmo genético: seleção, cruzamento e mutação. A seleção, conforme o próprio nome sugere, é o processo responsável pela escolha dos indivíduos na população para cruzamento. Uma das técnicas mais utilizada é seleção por roleta, nela, os cromossomos são colocados em uma roleta, sendo a porção ocupada por cada cromossomo proporcional ao fitness relativo à população. A roleta, então, é girada  $n$  vezes, selecionando a cada turno o indivíduo correspondente à porção apontada pela agulha.

<sup>2</sup> Para a pesquisa, estabelecemos que as redes MLP utilizadas teriam cinco camadas intermediárias, cada camada contendo no máximo 15 neurônios.

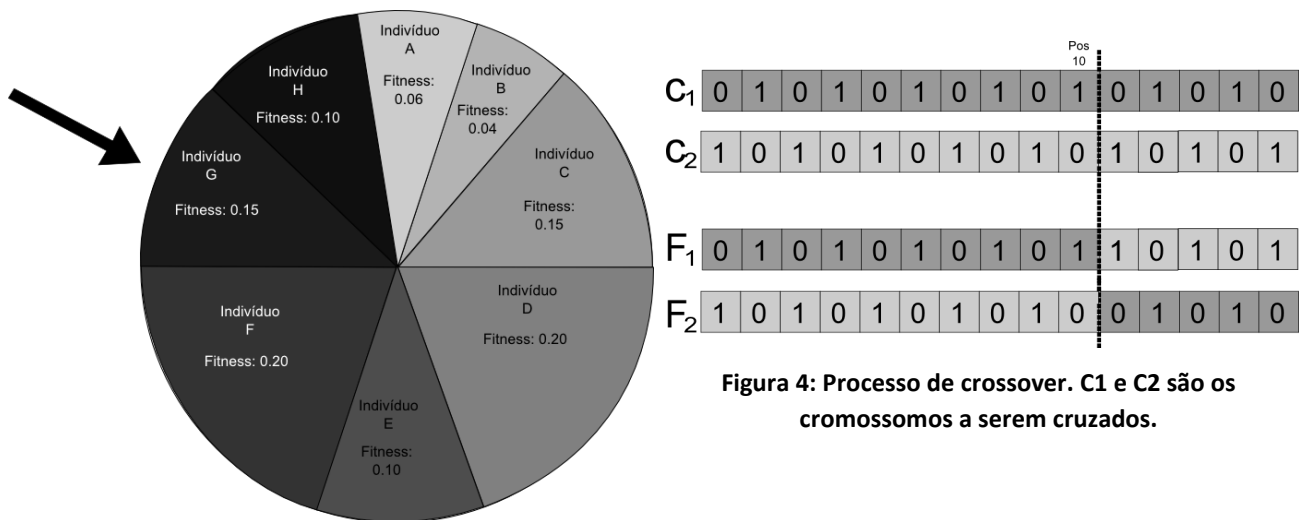


Figura 3: Seleção por roleta.

Figura 4: Processo de crossover. C<sub>1</sub> e C<sub>2</sub> são os cromossomos a serem cruzados.

O processo de cruzamento consiste em produzir novos indivíduos a partir daqueles selecionados anteriormente. Uma posição fixa *i* é escolhida de modo aleatório. Cada indivíduo é então dividido em duas partes, sendo o “ponto de corte” justamente a posição *i* definida anteriormente. O cruzamento ocorre de fato nesta etapa. A porção anterior (à posição *i*) do primeiro indivíduo C<sub>1</sub> se junta à posterior do segundo indivíduo C<sub>2</sub>, formando o primeiro descendente F<sub>1</sub>, e a porção anterior (à posição *i*) do segundo indivíduo C<sub>2</sub> se junta à posterior do primeiro indivíduo C<sub>1</sub>, formando o segundo descendente F<sub>2</sub>. A Figura 4 ilustra esse processo.

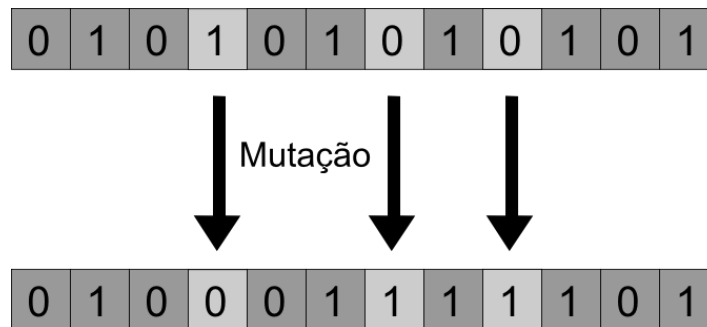


Figura 5: Processo de mutação.

E por fim, o operador de mutação é responsável por modificar aleatoriamente o valor original de pelo menos um ou mais bits do cromossomo, conforme é possível observar na Figura 5.

Apesar da simplicidade, a mutação desempenha um importante papel durante a execução do GA, por ser o mecanismo responsável pela manutenção da diversidade, visto que, permiti o aparecimento de novos indivíduos na população.

#### 4 METODOLOGIA E RESULTADOS

Os experimentos foram conduzidos sobre oito problemas de classificação de padrões retirados do repositório UCI (ASUNCIÓN; NEWMAN, 2007), tais problemas são descritos na Tabela 1. Grande parte deles, se não todos, já serviu de investigação em trabalhos correlatos. De forma a

garantir resultados estatisticamente significantes, para cada conjunto de problemas foi aplicado um particionamento estratificado na ordem de 90% para treinamento e 10% para teste, ambos gerados aleatoriamente.

**Tabela 1: Descrição dos problemas de classificação.**

| Bases         | Descrição   | Atributos | Instâncias | Classes |
|---------------|---|-----------|------------|---------|
| Breast-cancer | Busca identificar a presença de tumores em mamas, que pode ser benignos ou malignos.  | 286       | 9          | 2       |
| Colic         | Avalia se cólicas em cavalos são provenientes de lesões cirúrgicas ou não.  | 28        | 368        | 2       |
| Credit-a      | Aprovar ou não o crédito a um dado cliente mediante a análise de seu perfil de crédito.   | 15        | 690        | 2       |
| Heart-c       | Diagnosticar a presença ou ausência de risco de doença coronária (enfarto) a partir dos resultados de vários testes clínicos.   | 13        | 303        | 5       |
| Sick          | Classifica a presença ou ausência de tireoidiana.   | 29        | 3772       | 2       |
| Sonar         | Distinguir entre materiais metálicos ou rochosos de acordo com os níveis de intensidade de sinais de sonar enviados sob diferentes condições (ângulos, frequências, etc.) | 60        | 208        | 2       |
| Vote          | Avalia os votos da Câmara de Representantes nos EUA se foram realizados por um republicano ou um democrata.   | 16        | 435        | 2       |
| Waveform      | Classificação de tipos de onda.   | 21        | 5000       | 3       |

Um protótipo da abordagem evolutiva para configuração automática de redes neurais com múltiplas camadas foi implementado na linguagem Java, lançando-se mão de insumos providos pelos ambientes WEKA (WITTEN; FRANK, 2005) e jMetal (DURILLO; NEBRO, 2011). É válido mencionar que esses frameworks vêm sendo recentemente bastante adotados como base de desenvolvimento e validação de novas abordagens de aprendizado de máquina. O primeiro citado provê implementações de diversos algoritmos de aprendizado de máquina, dentre os quais as redes MLP, utilizadas nesta pesquisa. Já o segundo, por sua vez, apresenta uma suíte de apoio através da qual foram codificados o problema na forma de cromossomo e o cálculo do fitness, além de possuir uma implementação do NSGA-II.

Foram produzidos resultados de erro médio de teste para redes neurais MLP via configuração padrão, vide Tabela 2, e para a abordagem de redes neurais configuradas automaticamente via Algoritmo Genético, conforme resultados de erro médio de teste apresentados na Tabela 3.



**Tabela 2: Descrição dos problemas de classificação.**

| Parâmetro   | Valor                 |
|---|-----------------------|
| Normalização de atributos.  | Verdadeiro            |
| Número de épocas.   | 500                   |
| Quantidade de neurônios nas camadas intermediárias <sup>3</sup> . | $\frac{n_a + n_c}{2}$ |
| Taxa de aprendizado.  | 0,3                   |
| Termo <i>momentum</i> .   | 0,2                   |

**Tabela 3: Resultados referentes aos valores de erro médio de teste para rede.**

| BASES         | MLP PADRÃO | MLP EVOLUTIVA |
|---------------|------------|---------------|
| Breast-cancer | 0,0334     | 0,0404        |
| Colic         | 0,2068     | 0,1873        |
| Credit-a      | 0,1507     | 0,1652        |
| Heart-c       | 0,1717     | 0,1945        |
| Sick          | 0,0302     | 0,0342        |
| Sonar         | 0,1683     | 0,1588        |
| Vote          | 0,0393     | 0,0553        |
| Waveform      | 0,1542     | 0,1607        |

A partir dos resultados analisados, observou-se que a rede neural configurada através do NSGA-II apresentou melhor desempenho frente rede que utiliza configuração padrão em 80% dos experimentos realizados. Desta forma, pode-se sugerir que a proposta de combinação de sistemas inteligentes sugerida nesta pesquisa apresenta considerável relevância quanto ao seu uso.

O ambiente construído a partir dos frameworks citados anteriormente também foi relevante para a pesquisa, visto que permite uma melhor comparação dos resultados aqui produzidos como outros trabalhos correlatos, bem como, servirá de base para avanços em outras pesquisas no tema aqui apresentado.

## 5 CONSIDERAÇÕES FINAIS

Neste trabalho foi apresentada uma abordagem evolutiva para configuração automática dos parâmetros de controle de redes neurais com multiplicas camadas a serem aplicadas em problemas de classificação de padrões. Os resultados indicam uma diminuição do erro médio de teste produzido pela nossa abordagem frente aos resultados obtidos por redes do mesmo tipo que utilizam configuração padrão (encontrados na maioria das ferramentas de aprendizado de máquina). Outro ponto a receber destaque foi a utilização dos frameworks WEKA e jMetal, haja vista as contribuições no que se refere a reuso de código, diminuindo, assim, o tempo empreendido na codificação dos experimentos, bem como, a possibilidade de comparação dos resultados com outros trabalhos correlatos que utilizam tais ferramentas.

<sup>3</sup> A variável  $n_a$  representa o número de atributos e  $n_c$  o número de classes.

Como trabalho futuro, pretende-se utilizar outras abordagens bioinspiradas, tais como Otimização por Enxame de Partículas (KENNEDY; EBERHART, 1995) e Sistemas Imunológicos Artificiais (CASTRO; ZUBEN, 2002), para configuração automática de redes neurais com múltiplas camadas e de diferentes topologias.

## 6 REFERÊNCIAS BIBLIOGRÁFICAS

ASUNCIÓN, A.; NEWMAN, D. J. UCI Machine Learning Repository. University of California at Irvine. Disponível em: <<http://ics.uci.edu/~mlearn/MLRepository.html>>. Acesso em 01 nov. 2012.

BLUM, Avrim; RIVEST, Ronald. Training a 3-node neural network is NP complete. **Workshop on Computational Learning Theory**, Cambridge, p. 9-18. jan. 1988.

CASTRO, L. N. de; VON ZUBEN, F. J.. Learning and optimization using the clonal selection principle. *IEEE Transactions on Evolutionary Computation*, Portland, p. 239-251. 07 ago. 2002.

DEB, Kalyanmoy et al. A fast and elitist multiobjective genetic algorithm: Nsga-ii. **Ieee Transactions On Evolutionary Computation**, Portland, p. 182-197. 07 ago. 2002.

DURILLO, Juan J.; NEBRO, Antonio J.. JMetal: A java framework for multi-objective optimization. **Advances In Engineering Software**, Amsterdam, p. 760-771. 10 out. 2011.

EIBEN, A. E.; SMITH, J.E.. *Introduction to Evolutionary Computing*. Berlin: Springer, 2003.

FACELI, Katti et al. **Inteligência Artificial: Uma Abordagem de Aprendizado de Máquina**. São Paulo: LTC, 2011.

HASTIE, Trevor; TIBSHIRANI, Robert; FRIEDMAN, Jerome. **The Elements of Statistical Learning: Data Mining**. Berlin: Springer, 2001.

KENNEDY, James; EBERHART, Russell C.. Particle swarm optimization. **Proceedings Of The Ieee International Conference On Neural Networks**, Perth, p. 1942-1948. 27 nov. 1995.

MCCULLOCH, Warren S.; PITTS, Walter. A logical calculus of the ideas immanent in nervous activity. **Bulletin Of Mathematical Biology**, Berlin, p. 115-133. 01 dez. 1943.

WITTEN, I. H.; FRANK, E. **Data Mining: Pratical Machine Learning Tools and Techniques**. 2. ed. Berlin: Springer, 2005.