



# Aplicação de Redes Neurais Profundas para Detecção de Notas Bancárias Falsas

**Jherson Haryson Almeida Pereira**  
**harysonjherson@gmail.com**  
UFPA

**Danilo Duarte**  
**danilo8@gmail.com**  
UFPA

**Lidio Mauro Lima de Campos**  
**limadecampos@gmail.com**  
UFPA

**Resumo:** Todos os dias milhões de pessoas usam notas para fazer transações. A segurança dessas notas bancárias para governos e bancos é, portanto, um fator essencial para combater fraudes. Este artigo mostra a utilização de um Sistema Neural Híbrido para a detecção de notas bancárias falsas. Para isso, foram realizados aperfeiçoamentos no sistema original, dotando o mesmo de técnicas de aprendizado profundo (deep learning), aprimorando assim, a capacidade das RNAs em classificar, reconhecer, detectar e compreender universos complexos. O modelo citado (ADEANN-Deep) atingiu as expectativas, encontrando soluções satisfatórias para o problema proposto.

**Palavras Chave:** Redes neurais - Sistemas Híbridos - Algoritmos Genéticos - -

## 1. Introdução

O projeto de uma Rede Neural Artificial (RNA) pode ser considerado um processo complexo de tomada de decisão que frequentemente depende da experiência do usuário. Em geral, essa tarefa de projeto de RNA é um processo de tentativa e erro, em que um número de diferentes funções de transferência e quantidade de neurônios devem ser ajustados para resolver um problema específico. Como o projeto de RNA, ainda é feito, manualmente, a automação deste processo de projeto beneficia o processo de tomada de decisão feito por especialistas humanos (Bukhtoyarov e Semenkin, 2013).

Algoritmos bio-inspirados mostraram-se eficientes em diferentes problemas de otimização não-linear (Kromer, P. Plato et al 2014), (Zhang, J et al., 2007), (Ojha, V. K. et al., 2015), (Kazaryan, D. E. et al., 2017). Devido à sua eficiência e adaptabilidade, o interesse em pesquisa no campo da neuroevolução (abordagem evolucionária para projetar RNAs) tem aumentado ultimamente. A questão central na neuroevolução é construir um esquema de codificação indireta eficiente (Stanley, K. O. Miikkulainen et al., 2012), (De Campos et al., 2011), (Gauci, J. et al., 2010), (Lee, D. W. et al., 2005), (Risi, S. et al 2012). O que significa o algoritmo genético evolui uma descrição comprimida da RNA em vez da própria RNA (Gauci, J. et al., 2010).

O problema de projeto automático de rede neural artificial (RNA) pode ser formulado como um problema de otimização, em que se deseja encontrar, automaticamente, uma arquitetura de RNA, que seja capaz de minimizar uma função de avaliação que, a princípio, consiste no erro quadrático médio entre valores desejados nas saídas dos neurônios da RNA, em um conjunto de observações e valores calculados nestas saídas (FRANÇA, Thyanne et al., 2018). A topologia de uma RNA pode ser descrita como um conjunto finito de neurônios, representados por 'nós' em um grafo, e um conjunto finito de conexões entre neurônios, ou seja, 'arcos' direcionados no grafo.

O objetivo principal dessa pesquisa é aperfeiçoar o sistema híbrido ADEANN proposto em (De Campos et al., 2016). Um dos principais focos dessa nova melhoria é a inclusão de conceitos e técnicas de *deep learning* ao algoritmo citado, a fim de torná-lo capaz de solucionar problemas complexos como classificação de imagens, reconhecimento de fala, detecção de objetos e descrição de conteúdo.

O presente artigo é organizado da seguinte forma. A Seção 2 apresenta a revisão de literatura a respeito do tema. A Seção 3 apresenta a arquitetura do modelo base ADEANN. A Seção 4 apresenta aplicações da nova versão do programa resolvendo problemas de otimização. A Seção 5 apresenta algumas considerações finais.

## 2. Revisão de Literatura

A maioria dos algoritmos neuroevolutivos (ANEs) utiliza sistema de codificação direta (SCD) (Miller, G. F et al., 1989), que especifica cada conexão e nó do genótipo que aparecerá no fenótipo (RNA). No NEAT (Stanley, K. O. Miikkulainen et al., 2012), o SCD incorpora algumas entidades biologicamente plausíveis e altera tanto os parâmetros de ponderação como as estruturas das redes. (Sanchez, D et al., 2014) desenvolveram uma nova otimização multi-objetivo para um algoritmo genético hierárquico (MOHGA) baseado na abordagem micro-GA. No entanto, esse método não foi testado em outros aplicativos como a previsão de séries temporais e não geram RNAs profundas.

A crescente complexidade da computação evolutiva exige métodos mais sofisticados do que o mapeamento direto de genótipo para fenótipo. No outro extremo, estão os sistemas de codificação indireta (SCIs) (De Campos et al., 2011), (Boozarjomehry, R. B et al., 2001), (Kitano, H et al., 1990), (Gauci, J. et al., 2010), (Lee, D. W. et al., 2005), (Risi, S. et al 2012),

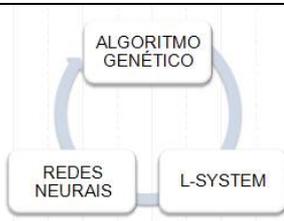
(Kazaryan, D. E. et al., 2017). Em um SCI, a geração de rede é indiretamente especificada pelo genótipo. A descrição da solução é compactada, permitindo gerar topologias complexas de RNAs.

Os seguintes métodos (De Campos et al., 2011), (Lee et al., 2005), (Kazaryan, D. E. et al., 2017), pertencem à classe de GDSs (sistemas generativos e de desenvolvimento) usando a evolução gramatical (GE). As metodologias deram um passo em direção a abordagens inspiradas biologicamente. (Lee et al., 2005) buscaram inspiração no DNA, em sua pesquisa, a informação é codificada usando os símbolos A, G, T e C. Uma sequência de três desses símbolos é conhecida como um códon. A sequência de códons entre esses delimitadores é traduzida em uma regra de produção para o desenvolvimento de um controlador neural. A regra de produção do *L-System* usada por (Lee et al., 2005) é livre de contexto e não permite gerar redes recorrentes. Uma ideia similar (De Campos et al., 2011) usando cadeias binárias também existe em que o processo de leitura e tradução de bits pode ser repetido a partir de diferentes bits na cadeia para produzir diferentes regras de produção. O ANE proposto por (De Campos et al., 2011) tem uma desvantagem, que é a dificuldade em definir os parâmetros da função de adequação do Algoritmo Genético para direcionar a busca por arquiteturas mínimas.

### 3. Arquitetura do modelo base ADEANN

O sistema citado híbrido é composto por basicamente com três tecnologias, sendo elas:

Algoritmo Genético	Rede Neural	L-System
Geração populações de indivíduos e aperfeiçoá-los com base em cruzamentos entre indivíduos.	Capacidade de aprendizado de máquina com o objetivo de solucionar problemas complexos.	Descrever características (Genótipos), que representam arquiteturas de redes neurais (RNAs), fenótipos.



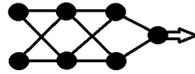
**Figura 1.** Arquitetura do ADEANN. **Fonte:** os autores.

Os três subsistemas atuam de forma integrada como descrito a seguir: o *L-system* cria e modela a arquitetura da Rede Neural com base em regras de produção. A RNA aprende com base no treinamento, sendo gerado uma aptidão (fitness) relacionada a capacidade de solucionar um problema estático ou dinâmico. Por sua vez Algoritmo Genético gera vários indivíduos (conjunto de regras de produção em um L-system) e executa técnicas biologicamente inspiradas para o cruzamento e obtenção de melhores indivíduos, ao final é gerada outra geração com os melhores indivíduos da geração anterior.

### 4. Aprimoramentos para o ADEANN-DEEP

As fases de aprimoramento foram planejadas em 4 etapas, tomando por base, a arquitetura base do ADEANN (Figura 1), sendo elas:





**Figura 4.** RNA equivalente aos genótipos gerados na Figura 3. **Fonte:** Os autores

## 4.2 Aprimoramento da Rede Neural Artificial

Após a conclusão da primeira fase, começou-se o aprimoramento do modelo de redes neurais utilizado pelo ADEANN, onde anteriormente o sistema gerava RNAs de três camadas e redes recorrente recorrentes. Adicionalmente, ampliou-se o modelo de rede neural para o modelo de *deep learning*, baseado no método do gradiente descendente.

O interesse em redes de *feedforward* profundas (*deep feedforward networks*) foi revivido por volta de 2006 (Hinton, G. E. et al, 2006), (Bengio, Y et al., 2006) por um grupo de pesquisadores reunidos pelo Instituto Canadense de Pesquisa Avançada (CIFAR). Nessa pesquisa buscou-se implementar esse tipo de arquitetura de RNA, visando acoplá-la ao antigo ADEANN, possibilitando obter a nova versão denominada ADEANN-Deep.

As modificações citadas acerca do aprimoramento da rede neural do ADEANN (figura 1) foram desenvolvidas a fim de torná-lo compatível ao modelo de *deep learning* com “n” camadas ocultas.

## 4.3 Aprimoramento do Algoritmo Genético

O AG irá avaliar o treinamento de redes neurais de três e quatro camadas, dessa forma, a equação abaixo representa cálculo para uma RNA de três camadas, penalizando, RNAs, que possuem um número elevado de neurônios utilizados na camada oculta (NINT), bem como uma RNA, que possua, uma generalização ruim, ou seja redes que possuem um alto quadrático médio (emq).

$$fitness = 1000 * (\exp(-emq) * \exp(-NINT)) + \left(\frac{1}{emq * NINT}\right) \quad (1)$$

Para o cálculo do fitness para uma rede de quatro camadas levou-se em consideração, os números de neurônios das camadas intermediárias, que são representados por NINT1 e NINT2, além disso o erro médio quadrático. A equação abaixo, privilegia redes com topologias econômicas e que conseguem generalizar bem.

$$fitness = 1000 * (\exp(-emq) * \exp(-NINT1) * \exp(-NINT2)) + \left(\frac{1}{emq * NINT1 * NINT2}\right) \quad (2)$$

Onde: NINT1, NINT2 e emq são respectivamente o número de neurônios nas CAMADA\_OCULTA\_1, CAMADA\_OCULTA\_2 e erro médio quadrático.

## 4.4 Integração dos modelos aprimorados descritos nas seções 4.1, 4.2, 4.3

Ao finalizar os aprimoramentos dos subsistemas, descritos nas seções 4.1, 4.2 e 4.3, os mesmos foram integrados conforme mostra a (Figura 1) formando um único sistema híbrido, denominado ADEANN-Deep. Agora, esse sistema é capaz de solucionar problemas complexos que demandam maior poder de processamento utilizando técnicas de *deep learning*, dessa forma se tornando um sistema preditivo que se difunde e se adapta melhor do

que seu modelo anterior (ADEANN), melhorando continuamente à medida que novos dados são adicionados, sendo mais dinâmicos do que sistemas preditivos baseados em regras de negócios.

## 5. Funcionamento e configurações do ADEANN-Deep

O sistema pode ser configurado para gerar diversas arquiteturas de RNAs. Dessa forma, essas configurações controlam os modelos de RNAs evoluídas pelo Algoritmo Genético, podendo ser gerados modelos de RNA's com três camadas, ou quatro, ou ambos e redes recorrentes.

O ADEANN-DEEP é inicializado com N gerações e M indivíduos por geração. Assim, ao iniciar o sistema com configurações para executar 100 gerações com 30 indivíduos, com redes neurais de três e quatro camadas, ao final, o sistema irá gerar no máximo 6000 redes neurais, sendo elas, no máximo 3000 redes para o modelo de três camadas e 3000 para o modelo de quatro camadas. O ADEANN-Deep gera modelos de redes neurais com métodos construtivos e de poda (Reed et al., 1993), podendo aumentar de tamanho ou diminuir dependendo da avaliação do fitness.

Ao término da execução do algoritmo, o sistema gera os dois melhores indivíduos das 100 gerações, sendo um deles, o melhor indivíduo do modelo de três camadas e o outro, sendo o melhor indivíduo do modelo de quatro camadas.

## 6. Experimentos realizados com o ADEANN-DEEP

Utilizou-se nas simulações o “dataset” *Banknote Authentication* obtido do repositório <https://archive.ics.uci.edu/ml/datasets/banknote+authentication>.

Características dos dados:	Características dos atributos:	Número de instancias:	Número de atributos:	Tarefas associadas
Multivariada	Real	1372	5	Classificação

Figura 5 - Fonte: <https://archive.ics.uci.edu/ml/datasets/banknote+authentication>

A extração dos dados foi realizada a partir de imagens tiradas de espécimes genuínos e forjadas semelhante a notas de banco. As imagens finais têm 400 x 400 pixels. As imagens estão em escala de cinza com uma resolução de cerca de 660 dpi.

Foram extraídos quatro atributos das imagens do *Banknote Authentication*, sendo elas:

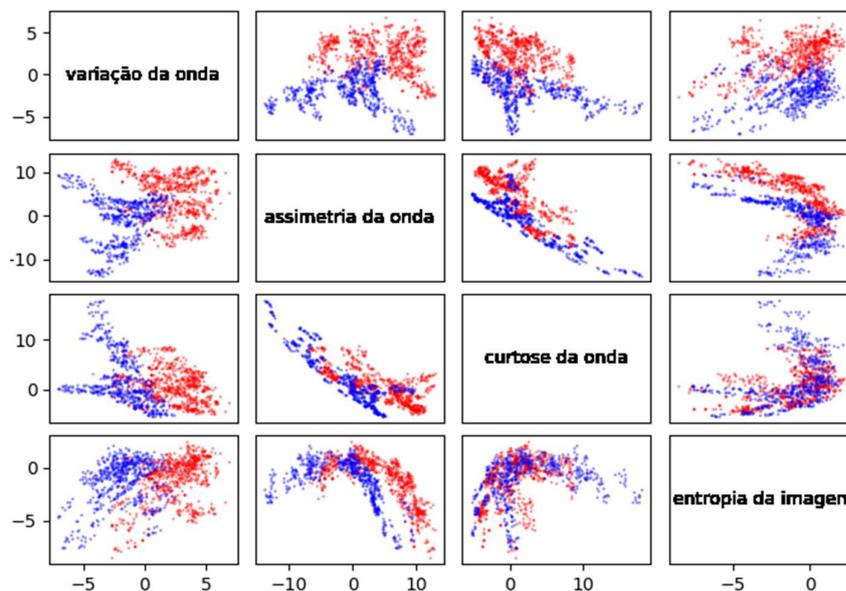
- Variância da onda transformada, utilizada como entrada.
- Assimetria da onda transformada, utilizada como entrada.
- Curtose de onda transformada, utilizada como entrada.
- Entropia de imagem, utilizada como entrada.
- Classe, utilizada como saída (1 – Nota Verdadeira e 2 – Nota Falsa)

A matriz de dispersão abaixo avalia as relações entre todos os pares de variáveis possíveis das entradas, onde os pontos vermelhos representam notas verdadeiras, e o azul, notas falsas.

Para a simulação do dataset citado foram utilizadas três arquiteturas de RNAs do ADEANN-Deep, sendo elas, RNAs de três camadas, quatro e três camadas recorrente. A fim de aferir a acurácia e precisão do sistema proposto, as simulações foram executadas com variações na estratégia de treinamento, onde o dataset foi particionado e dividido em uma porcentagem N para treino e M para teste. Para a padronização dos testes foi fixado o total de 10 gerações com 30 indivíduos cada, totalizando a geração de 300 redes neurais por modelo. A normalização da saída foi efetivada utilizando-se o ponto médio entre os valores de saídas 0 e 1, dessa forma, os valores menores a 0.5 serão considerados como 0, e os maiores a 0.5 foram considerados como 1.

## 7. Simulações realizadas

- **Sem Normalização e sem Pré-Processamento e Limpeza dos dados:**
  - **Simulação 1** – 2/3 dos registros para treino e 1/3 para teste
- **Com Normalização de Dados e com Pré-Processamento e Limpeza dos dados:**
  - **Simulação 2** – 4/5 dos registros para treino e 1/5 para teste.
  - **Simulação 3** – 2/3 dos registros para treino e 1/3 para teste.
  - **Simulação 4** – 1/2 dos registros para treino e 1/2 para teste.
  - **Simulação 5** – 1/3 dos registros para treino e 2/3 para testes.



**Figura 6** - Matriz de dispersão das entradas do Banknote Authentication. **Fonte:** Os autores.

### 7.1. Resumo das Simulações realizadas

Durante a simulação 1, o sistema foi testado utilizando-se 1/3 das amostras do *dataset* para teste e 2/3 para treino, como estabelece a técnica *holdout*. Em todas as simulações o ADEANN-Deep gerou três tipos de arquiteturas de RNAs: redes diretas com três camadas (3C), redes diretas com quatro camadas (4C) e redes recorrentes com três camadas. A Tabela 2 resume os resultados das simulações realizadas.

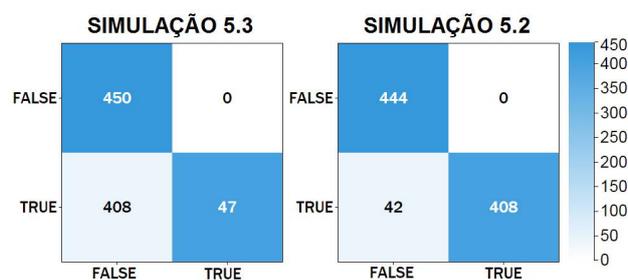
O resultado das simulações para a estratégia proposta de 1/3 dos dados para treino e 2/3 para teste, com pré-processamento de dados, mostrou que o melhor modelo de RNA, teve

quatro camadas e obteve 94.67% de acurácia, acertando 852 das 900 amostras, modelos de rede direta e recorrente com três camadas (3C) acertaram respectivamente 442 e 592 do total de 900 amostras. Quando se diminui a quantidade de instâncias para treino foi reduzida (Linhas 13 e 15 da Tabela 2), resultou no esperado, as acurácias dos modelos diminuíram, dessa forma, observou-se que os modelo de rede direta e recorrente com três camadas (3C), foram os que obtiveram os piores resultados, mostrando que são menos robustos, quando se utilizou um número pequeno de amostras (1/3) para treino, do que comparado a rede profunda com quatro camadas (4C). O modelo recorrente com três camadas (3C), ver Linha 15 da Tabela 2, obteve o pior desempenho dos três modelos.

**Tabela 2. Resumo das simulações realizadas**

Sim	RNA	Treino	Teste	EMQ	Geração/Ind.	NINT1	NINT2	Acurácia	TAPI/ AP2
1.1	3C	2/3	1/3	0.100700000	7/17	2	-	83.89%	0.65
1.2	4C	2/3	1/3	0.009100000	6/16	2	3	85.35%	0.3/0.85
1.3	R 3C	2/3	1/3	0.016900000	6/9	2	-	85.02%	0.5
2.1	3C	4/5	1/5	0.000326960	7/17	5	-	100%	0.01
2.2	4C	4/5	1/5	0.006054941	8/4	7	2	100%	0.01/0.01
2.3	R 3C	4/5	1/5	0.005663465	9/2	5	-	100 %	0.01
3.1	3C	2/3	1/3	0.000442760	8/28	5	-	100%	0.01
3.2	4C	2/3	1/3	0.000838865	4/8	7	7	99.364%	0.01/0.01
3.3	R 3C	2/3	1/3	0.005129151	4/29	5	-	99.575%	0.01
4.1	3C	½	½	0.001039442	8/28	5	-	99.855%	0.01
4.2	4C	½	½	0.006655863	7/15	5	7	100%	0.01/0.01
4.3	R 3C	½	½	0.008028630	4/29	2	-	98.11%	0.01
5.1	3C	1/3	2/3	0.042506489	8/28	7	-	61.36%	0.01
5.2	4C	1/3	2/3	0.006655863	6/13	7	2	94.67%	0.01/0.01
5.3	R 3C	1/3	2/3	0.090946561	7/29	2	-	54.67%	0.1

A matriz de confusão abaixo é referente as simulações 5.3 e 5.2

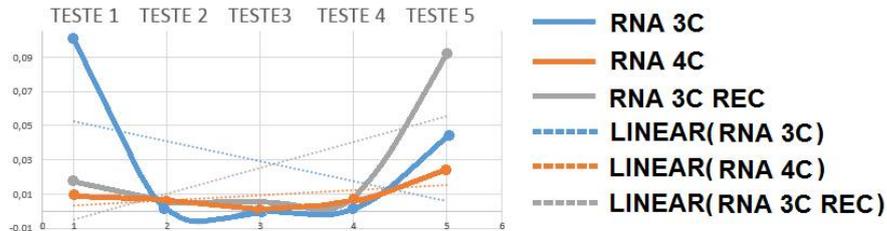


**Figura 7.** Matriz de confusão da simulação 5.3 e 5.2. **Fonte:** Os autores.

Ao analisar a simulação 5.3 da Figura 7 (referente a linha 13 da Tabela 2), pode ser visto que este modelo classificou corretamente todas as 450 notas falsas, porém ao classificar notas verdadeiras, o modelo conseguiu classificar corretamente como “verdadeiro” somente 42 notas, das 450 notas verdadeiras, dessa forma, gerando 408 falsos negativos. Nesta simulação, os modelos de rede direta e recorrente com três camadas (3C), obtiveram uma margem de acerto muito baixa (Linhas 13 e 15 da Tabela 2). O modelo de rede profunda com quatro camadas (4C) conseguiu generalizar e classificar bem os dados de teste, mesmo levando em consideração

a quantidade de 1/3 do total de dados para ser executado o treino (Linha 14 da Tabela 2) como pode ser visto na matriz de confusão abaixo (Figura 7, simulação 5.2).

A comparação entre esses três modelos mostra que o modelo de rede profunda com quatro camadas (4C) é mais robusto quanto a sensibilidade dos dados, onde em um universo onde a obtenção de dados para treino é mais escassa ele seria o sistema mais recomendado para ser colocado em produção, dado o universo proposto.



**Figura 9.** Tendência do erro médio quadrático das simulações. **Fonte:** Os autores.

## 8. Conclusão

Após o desenvolvimento dessa pesquisa, obteve-se um sistema de generativo artificial, baseado em L-System, que gera automaticamente fenótipos para redes neurais dos tipos 1-1, N-1, 1-N e M-N, com N camadas ocultas, distinguindo e representando com sucesso a cada um dos modelos, aqui citados, de acordo com a gramática estabelecida.

Ao todo foram desenvolvidas 15 simulações utilizando o dataset *Banknote Authentication* com variações na estratégia de seleção de *dataset*, a fim de verificar o desempenho do sistema proposto.

O ADEANN-Deep obteve resultados satisfatórios para o problema processamento de imagem, a nova funcionalidade de deep learning consegue identificar e classificar notas verdadeiras e falsas com maior precisão, inclusive com conjuntos de dados pequenos para treino (1/3 dos dados).

Acerca da eficiência do novo modelo de deep-learning o gráfico mostrado, na figura 9, mostra a tendência ao longo dos testes do erro médio quadrático em relação a cada simulação.

É possível notar um erro médio quadrático elevado para a RNA de três camadas, utilizando a primeira estratégia de treinamento de divisão de dataset com 4/5 dos dados para treino e 1/5 para teste sem normalização. Este modelo foi extremamente prejudicado pela *duplicidade e redundância de dados*. Após a normalização (simulação 2) pode ser visto que a RNA direta com três camadas (3C) melhorou consideravelmente.

Ao analisar o desempenho do algoritmo de *deep learning* para o processamento de imagens, também é possível notar através das linhas de tendências geradas através de uma função linear, que este é o sistema com menor variância, onde o coeficiente angular é a média são mais próximo de zero, ver Figura 9.

Dessa forma ao analisar cenário 5 (simulação 5), onde o conjunto de dados para treino (universo conhecido) é extremamente pequeno é necessário que seja classificado um conjunto de dados maior, o ADEANN-Deep com rede profunda de quatro camadas (4C) consegue classificar imagens com um erro médio quadrático menor que os modelos de rede direta e recorrente com (3C). Esses modelos obtiveram um percentual de acerto para a simulação 5 de 61,36% e 54,67% (Linhas 13 e 15 da Tabela 2) respectivamente e o novo modelo de rede profunda de 4 Camadas (Linha 14 da Tabela 2) conseguiu classificar corretamente cerca de 94,67% dos dados.

Apesar da grande maioria dos repositórios conterem um volume alto de registros, em alguns casos o que ocorre é o inverso. Neste caso, o sistema proposto conseguiu classificar perfeitamente os registros com uma margem aceitável e obteve resultados satisfatórios em casos onde o número de registros (universo conhecido) é limitado (simulação 5). Dessa forma o sistema mostrasse mais robusto, apresentando novas funcionalidades aplicadas em situações onde o volume de registros é escasso.

## 9. Agradecimentos

Os autores agradecem a Pró-Reitoria de Pesquisa e Pós-Graduação da UFPA (Universidade Federal do Pará) por meio do Programa PRODOUTOR, pela bolsa de estudos e auxílio financeiro que possibilitou a o desenvolvimento dessa pesquisa.

## 10. Referencias

- Bukhtoyarov, V.—Semenkin, E.: Evolutionary Three-Stage Approach for Designing of Neural Networks Ensembles for Classification Problems. In: Tan, Y., Shi, Y., Mo, H. (Eds.): *Advances in Swarm Intelligence (ICSI 2013)*, Lecture Notes in Computer Science, 2013, pp. 467–477.
- Kromer, P. Plato, J.—Sna, V.: Nature-Inspired Meta-Heuristics on Modern GPUs: State of the Art and Brief Survey of Selected Algorithms. *International Journal of Parallel Programming*, 2014, pp. 681–709, doi: 10.1007/s10766-013-0292-3.
- Stanley, K. O. Miikkulainen, R.: Evolving Neural Networks Through Augmenting Topologies. *Evolutionary Computation*, 2002, pp. 99–127, doi: 10.1162/106365602320169811.
- De Campos, L. Roisenberg, M. de Oliveira, R.: Automatic Design of Neural Networks with L-Systems and Genetic Algorithms – A Biologically Inspired Methodology. *The 2011 International Joint Conference on Neural Networks (IJCNN)*, 2011, pp. 1199–1206, doi: 10.1109/IJCNN.2011.6033360.
- Gauci, J.—Stanley, K. O.: Autonomous Evolution of Topographic Regularities in Artificial Neural Networks. *Neural Computation*, 2010, pp. 1860–1898, doi: 10.1162/neco.2010.06-09-1042.
- Lee, D.-W.—Kong, S. G.—Sim, K.-B.: Evolvable Neural Networks Based on Developmental Models for Mobile Robot Navigation. *Proceedings of 2005 IEEE International Joint Conference on Neural Networks (IJCNN '05)*, 2005, pp. 337–342.
- Risi, S.—Stanley, K. O.: An Enhanced Hypercube-Based Encoding for Evolving the Placement, Density, and Connectivity of Neurons. *Artificial Life*, 2012, pp. 331–363.
- Miller, G. F.—Todd, P. M.—Hegde, S. U.: Designing Neural Networks Using Genetic Algorithms. *Proceedings of the Third International Conference on Genetic Algorithms*, Morgan Kaufmann Publishers Inc., 1989, pp. 379–384.
- Zhang, J.-R.—Zhang, J.—Lok, T.-M.—Lyu, M. R.: A Hybrid Particle Swarm Optimization-Back-Propagation Algorithm for Feedforward Neural Network Training. *Applied Mathematics and Computation*, Vol. 185, 2007, No. 2, pp. 1026–1037, Special Issue on Intelligent Computing Theory and Methodology.
- Kitano, H.: Designing Neural Networks Using Genetic Algorithms with Graph Generation System. *Complex Systems*, Vol. 4, 1990, pp. 461–476.
- Boozarjomehry, R. B.—Svrcek, W. Y.: Automatic Design of Neural Networks Structures. *Computers and Chemical Engineering*, Vol. 25, 2001, No. 7-8, pp. 1075–1088, doi: 10.1016/S0098-1354(01)00680-9.
- Sanchez, D.—Melin, P.: Optimization of Modular Granular Neural Networks Using Hierarchical Genetic Algorithms for Human Recognition Using the Ear Biometric Measure. *Engineering Applications of Artificial Intelligence*, 2014, pp. 41–56, doi: 10.1016/j.engappai.2013.09.014.



Kazaryan, D. E.—Savinkov, A. V.: Grammatical Evolution for Neural Network Optimization in the Control System Synthesis Problem. *Procedia Computer Science*, Vol. 103, 2017, pp. 14–19, doi: 10.1016/j.procs.2017.01.002.

Ojha, V. K.—Dutta, P.—Chaudhuri, A.—Saha, H.: Understating Continuous Ant Colony Optimization for Neural Network Training: A Case Study on Intelligent Sensing of Manhole Gas Components. *International Journal of Hybrid Intelligent Systems*, 2015, No. 4, pp. 185–202.

De Campos, Lídio Mauro Lima, de Oliveira ,Roberto Célio Limão, Roisenberg, Mauro. Optimization of neural networks through grammatical evolution and a genetic algorithm, *Expert Systems with Applications*, 2016, Pages 368-384, ISSN 0957-4174

Hinton, G. E., Osindero, S. & Teh, Y.-W. A fast learning algorithm for deep belief nets. *Neural Comp.* 18, 1527–1554 (2006).

Bengio, Y., Lamblin, P., Popovici, D. & Larochelle, H. Greedy layer-wise training of deep networks. In *Proc. Advances in Neural Information Processing Systems 19* 153–160 (2006).

FRANÇA, Thayanne et al. An Agent Program Capable of Applying Local Search Strategies in the State Space of Well Defined Problems. *Anais do Encontro Nacional de Inteligência Artificial e Computacional (ENIAC), (S.I.)*, p. 377-388, oct. 2018. ISSN 0000-0000.