

# Avaliação de um algoritmo neuro evolutivo em problemas de classificação

Cássio K. da Costa<sup>1</sup>, Marcos P. H. Onishi<sup>1</sup>, Rafael L. de S. e Silva<sup>1</sup>, Eurico L. P. Ruivo<sup>1</sup>

<sup>1</sup>Faculdade de Computação e Informática - Universidade Presbiteriana Mackenzie (UPM) – São Paulo, SP – Brasil

{31929265, 31927416, 31948571}@mackenzista.com.br,  
eurico.ruivo@mackenzie.br

**Abstract.** *An evolutionary neural network mainly consists of the evolution of decision making from the interconnection of data. A classification base is defined in the organization of standardized information among several individuals that result in their classification in different groups. From the selection of a neural network called NEAT (Neuroevolution of Augmenting topologies), normally used for decision making in games, the question of its effectiveness when performing classification bases arises. This work aims to validate if the modification of its configurations, in addition to the comparison with a specific algorithm in the area, are sufficient to verify the use in the classification of individuals.*

**Keywords:** *NEAT. Classification Problem. Neuroevolution Algorithm.*

**Resumo.** *Uma rede neural evolutiva consiste, principalmente, na evolução de tomada de decisões a partir da interconexão de dados. Uma base de classificação se define na organização de informações padronizadas dentre diversos indivíduos que resultem na classificação dos mesmos em diferentes grupos. A partir da seleção de uma rede neural denominada NEAT (Neuroevolution of Augmenting topologies), normalmente utilizada para tomada de decisões em jogos, surge o questionamento de sua eficácia ao executar bases classificativas. Este trabalho tem por objetivo analisar se as modificações de suas configurações, além da comparação com um algoritmo específico da área, são suficientes para verificar sua utilização na classificação de indivíduos.*

**Palavras-chaves:** *NEAT. Problema de Classificação. Algoritmo Neuroevolutivo.*

# 1. Introdução

## 1.1 Contextualização e Relevância do Tema

Segundo Lippmann (1987), sobre redes neurais: “Estes modelos buscam obter um bom desempenho através da densa interconexão de elementos computacionais simples. A este respeito, a estrutura da rede neural artificial é baseada em nosso atual entendimento do sistema nervoso biológico.”(LIPPMANN, R.P. 1987, tradução livre). Alahakoon (2006) oferece duas visões para alcançar o ‘bom desempenho’ citado por Lippmann: “A topologia apropriada e o peso da conexão são duas propriedades muito importantes que uma rede neural deve ter para realizar a classificação de dados com sucesso.” (CHEN, L.; DAMMINDA, A. 2006, “tradução nossa”).

Uma base de dados de classificação se consiste, principalmente, em diversas informações definidas (sejam respostas de um questionário ou características situacionais) que auxiliem na classificação de um indivíduo de acordo com os parâmetros informados. Para treinar uma rede neural, é essencial obter uma base de informação sólida que permita o melhor aprendizado artificial do algoritmo implementado. O estudo realizado neste artigo aborda bases pré-selecionadas do site UCI (DUA; GRAF, 2019), que abrange diversas pesquisas já autorizadas e direcionadas ao uso para o treino de redes neurais.

Dentre os diversos métodos para evoluir redes neurais, o NEAT (Neuroevolution of Augmenting Topologies)(STANLEY; MIKKULAINEN, 2002), foi o algoritmo selecionado para o estudo neste artigo. Diferente de outras estratégias, a maior força deste método se apresenta principalmente por ‘marcações históricas’, estas dão ao NEAT uma nova e poderosa capacidade: saber exatamente quais genes combinam entre si.

Com o estudo e entendimento do algoritmo NEAT, surge a possibilidade de testá-lo em um cenário em que sua participação é pouco observada: Classificar indivíduos. Este trabalho teve por objetivo avaliar sua qualidade em bases de classificação, para por fim compará-lo com outro algoritmo reconhecido no assunto, denominado KNN.

O KNN (k-Nearest Neighbors) é um algoritmo de classificação e predição de dados, porém, não bio-inspirado. Isso se deve, principalmente, à simplicidade de suas configurações e seu custo reduzido de tempo e poder computacional.

## 1.2 Contextualização do Problema de Pesquisa

O algoritmo NEAT é famoso dentre as estruturas de redes neurais por sua categorização de populações em diferentes “espécies”, baseando-se na estrutura e peso de cada variável. Levando tal afirmativa em consideração, Chen Alahakoon constata em seu livro *NeuroEvolution of Augmenting Topologies with Learning for Data Classification*:

“Algoritmos evolutivos podem ser usados para evoluir o peso e a topologia da rede neural simultaneamente, uma abordagem de evolução que tem sido bem sucedida. NeuroEvolution of Augmenting Topologies (NEAT) é um desses métodos.”

Chen em seu livro enaltece o algoritmo por sua grande habilidade de evoluir tanto a rede estrutural quanto os pesos da conexão. Tal afirmação colabora para o propósito deste trabalho, pois apresenta-o como um bom candidato para ser aplicado em bases de classificação. Isso se deve, principalmente, à sua capacidade de evoluir tipos de estruturas a partir da mutação e preservar outras dependendo da forma em que organiza as “espécies” analisadas.

Por muitas vezes, um algoritmo neuro evolutivo é limitado a um conjunto específico de cenários, necessitando adaptações e reestruturações em sua composição. Neste contexto, esta pesquisa busca viabilizar a utilização do algoritmo NEAT em problemas de classificação.

### **1.3 - NEAT**

Um algoritmo de neuroevolução pode ser categorizado como “Machine Learning” (Aprendizado de máquina) e, como o nome sugere, é um algoritmo que visa a evolução. Sendo algo próximo do conceito de uma inteligência artificial, corresponde a ideia de uma máquina que possa evoluir e eventualmente tomar decisões a partir de um determinado tipo de treinamento.

Considerado "bioinspirado", o NEAT recebe tal denominação pela sua forma de evoluir, que se assemelha a conjuntos de seres vivos evoluindo naturalmente em cadeia, como uma colmeia de abelhas. Assim como na organização/topologia de insetos, o algoritmo NEAT também usa recursos como: genes, neurônios, indivíduos, grupos, espécies e gerações (STANLEY; MIIKKULAINEN, 2002).

O NEAT é um método de neuroevolução do tipo "TWEANN" (Topology and Weight Evolving Artificial Neural Network), um tipo de redes neural com evolução por peso, capaz de evoluir redes com qualquer número de entrada ou saída. Tal algoritmo supera significativamente o método de neuroevolução de topologia fixa (sem peso), e é utilizada principalmente na área de jogos digitais. Cada gene(neurônio) de conexão específica três principais estruturas: Nó de entrada, Nó de saída e Peso da conexão (STANLEY; MIIKKULAINEN, 2002).

Abordando as mutações do algoritmo para cada geração, denomina-se como uma mutação “add node” quando uma conexão existente é dividida e o novo nó é colocado onde a conexão antiga costumava ser. A conexão antiga é desabilitada e duas novas conexões (“add connection”) são adicionadas ao genoma. Este método de adição de nós é selecionado para integrar novos nós imediatamente à rede. Tanto o novo gene de conexão criado na primeira mutação quanto os dois novos genes de conexão adicionados durante a nova mutação do nó recebem os números de inovação (valores que nunca serão

alterados que permitem encontrar genes correspondentes durante o cruzamento e criar uma cronologia de cada gene no sistema). No futuro, sempre que esses genomas se cruzarem, os descendentes herdarão os mesmos números de inovação em cada gene, sendo assim, a origem histórica de cada gene no sistema é conhecida ao longo da evolução.

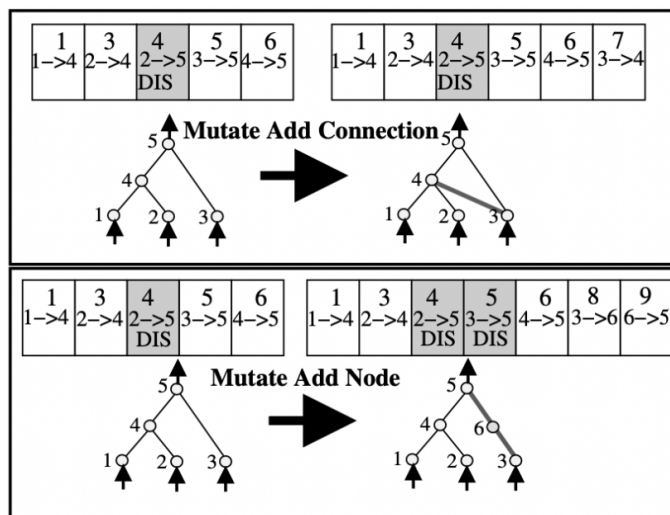


Figura 1: Os dois tipos de estruturas mutantes do NEAT. Retirado de (Stanley; Miikkulainen, 2002)

Certas “marcações históricas” dão ao NEAT a capacidade de saber quais genes se combinam. Os genes que não correspondem são disjuntos ou excessivos dependendo se ocorrem dentro ou fora do intervalo dos números de inovação. Ao cruzar, os genes em ambos os genomas com os mesmos números de inovação são alinhados, tendo-se os que não combinam herdando do pai mais apto. Se forem igualmente aptos, herdarão de ambos os pais aleatoriamente. Dessa forma, as ‘marcações históricas’ permitem que o NEAT realize o cruzamento sem a necessidade de análises topológicas complexas, sendo este a maior característica do NEAT.

## 2. Referencial Teórico

### 2.1 Algoritmos Bioinspirados

Algoritmos bioinspirados seguem a linha de estudo da ciência da computação, com a resolução de problemas utilizando modelos da biologia como, comportamentos sociais, emergenciais e até mesmo formas. Alguns exemplos de utilização são, a máquina de Turing (TURING, 1936), o formigueiro (HOFSTADTER, 1979) e também de redes neurais artificiais (MCCULLOCH; PITTS, 1943).

#### 2.1.1 Algoritmos Neuroevolutivos

A neuroevolução são técnicas de inteligência artificial que aplicam algoritmos evolucionários para geração de redes neurais artificiais (RNAs), parâmetros e regras

(STANLEY, 2017). Tendo como inspiração o processo biológico evolutivo do sistema nervoso na natureza. Esse tipo de algoritmo é comumente utilizado para estudos de jogos digitais, no qual uma estrutura de inteligência artificial é programada para concluir um jogo de forma a obter as formas otimizadas (RISI; TOGELIUS, 2017).

Algoritmos evolutivos operam com com uma população de genótipos ou também chamados de genomas. Os mesmos são mapeados em uma rede neural e são avaliadas com alguns requisitos para conseguir obter um resultado de acurácia através dos testes. Para o caso de estudo, foi utilizado a codificação direta, no qual o genótipo é diretamente mapeado no fenótipo (KASSAHUN et. al, 2007), utilizando o método *NeuroEvolution of Augmenting Topologies* (NEAT) (STANLEY; MIIKKULAINEN, 2002).

### **2.1.2 Algoritmos Genéticos**

Entre 1950 até 1960 cientistas da computação modelaram um conceito de evolução que deu origem ao conceito de algoritmos genéticos. Estes algoritmos geram soluções otimizadas e para problemas de procura, alguns operadores utilizados são: mutação, seleção, recombinação (crossing over), hereditariedade (MITCHELL, 1996).

### **2.1.3 NeuroEvolution of Augmenting Topologies (NEAT)**

O artigo “Efficient Evolution of Neural Network Topologies” é referência sobre o algoritmo NEAT, escrito por seus criadores, eles abordam temas como a evolução das redes neurais artificiais, pontuando seus principais fatores de performance, sendo eles: aplicação de uma metodologia interligada entre topologias; se utilizam de especiação para a proteção de estruturas novas; o crescimento a partir de uma estrutura base. Comprovando através de estudos que cada um dos componentes apresentados é necessário para o sistema. Já o artigo “Evolving Neural Networks through Augmenting Topologies“, também escrito pelos autores, tem como enfoque a descrição e desenvolvimento da metodologia do algoritmo NEAT. Reforçando que esta metodologia se demonstra superior a outras metodologias de topologia fixa devido a, principalmente, benchmarks em aprendizagem de tarefas, sendo notável o avanço no campo de pesquisa de algoritmos genéticos (STANLEY; MIIKKULAINEN, 2002).

## **2.2 Problemas de classificação.**

O site da UCI - Machine Learning Repository (DUA; GRAF, 2019) será um dos referenciais estudados neste artigo, abrangendo diversas bases de dados previamente estabelecidas e devidamente autorizadas. Dentre elas, destacam-se três tipos: classificação, agrupamento e regressão. Este trabalho de conclusão de curso será voltado ao tema de classificação de indivíduos, se apropriando de bases que classifiquem entrevistados/fatores em questionamento a partir de parâmetros específicos, sejam informações ou questionários, que permitam separá-los em grupos.

## **2.3 Trabalhos Correlatos**

CHEN E ALAHAKOON (2006) publicaram o artigo “NeuroEvolution of Augmenting Topologies with Learning for Data Classification”, abordando topologias e pesos de conexão como os dois principais pilares para a boa análise de uma rede neural artificial. Neste documento é citado o treinamento do L-NEAT (Learning-NEAT) para problemas de classificação de dados, realizando a subdivisão de tarefas e aprendendo-as, para por fim incorporá-las ao algoritmo NEAT tradicional. Tal artigo foi selecionado principalmente pelos estudos e incorporações do NEAT que podem ser utilizados durante a realização deste trabalho.

ISHIBUCHI et. al (1995) propuseram a construção de um método baseado em um algoritmo genético, para filtrar um pequeno grupo de componentes e construir um grupo compacto de alta classificação. Os dois principais objetivos desse projeto eram a classificação do maior número de indivíduos de forma correta e minimizar o erro de agrupamentos. O fitness de cada indivíduo é especificado por código por dois objetos, tendo assim como treinar e testar os dados com o método.

### **3. Metodologia da Pesquisa**

Para alcançar o objetivo do trabalho e observar as características do algoritmo NEAT em bases de classificação, uma estratégia de execução foi elaborada. Dividida em cinco fases, tal estratégia consistiria em Seleção, Estudo, Teste, Análise e Conclusão.

Sendo a primeira etapa, a fase de Seleção consistiu na identificação de bases destinadas à classificação de indivíduos. Para facilitar desenvolvimentos futuros, optou-se por bases sem dados categóricos e sem valores faltantes. Inicialmente foram separadas 15 bases classificativas, retiradas do site da UCI (DUA; GRAF, 2019), todas apresentando menos de mil atributos e apenas valores inteiros ou reais.

A partir da seleção de tais fases, a etapa seguinte consistiu em encontrar um modelo funcional que aplicasse o algoritmo NEAT em um código padrão. Tinha-se por objetivo o estudo e adaptação do programa para executar as bases encontradas. O código selecionado para cumprir tal tarefa se tratava da aplicação da lógica booleana XOR, alterando a definição de duas entradas seria possível alcançar quatro possíveis saídas (MCINTYRE et. al, 2015 ).

A partir do modelo obtido, o código foi adaptado para suportar as bases desejadas. Dentre as modificações, adaptou-se o programa original para sair do modelo “Hardcoded” da lógica XOR e aceitar bases de qualquer tamanho. Também foi elaborada uma função para calcular a acurácia dos resultados ‘Outputs’ que seriam obtidos.

Com as modificações realizadas no código, cinco das quinze bases previamente selecionadas foram escolhidas para testes e estudo. Para corresponder com as modificações no código mencionadas previamente, tais bases apresentavam classes (resultados da classificação) binárias, ou seja, resultados classificados em apenas 0 ou 1, e um número crescente de atributos (colunas).

As bases a serem utilizadas na fase de Testes:

- Breast Cancer - 570 linhas, 32 colunas (WOLBERG et. al, 1995)
- Fertility - 101 linhas, 9 colunas (GIL et. al, 2013)
- Caesarean section - 90 linhas, 5 colunas (AMIN; ALI, 2018)
- Haberman - 306 linhas, 4 colunas (LIM, 1999)
- Tic-Tac-Toe - 958 linhas, 9 colunas (AHA, 2019)

As bases acima foram rodadas vinte vezes para cada conjunto de testes no algoritmo adaptado NEAT em um computador macbook air m1 2020, 8gb ram, com processador M1 da Apple, CPU de 8 núcleos (4 de desempenho e 4 de eficiência), GPU de 7 núcleos, GPU de 8 núcleos e Neural Engine de 16 núcleos (APPLE, 2022).

A acurácia obtida em cada base era calculada conforme o número de resultados positivos encontrados em seu treinamento após o número de gerações atribuídas. Caso o algoritmo esperasse um valor, porém, o output NEAT gerasse um resultado diferente do mesmo, considera-se um resultado negativo. Para centralizar os resultados em uma medida de fácil análise, outputs que gerassem um valor menor ou igual a '0.5' eram arredondados para '0', maiores eram considerados '1'.

A partir do cálculo de acurácia estabelecida, cinco tabelas foram montadas com base nos resultados obtidos. Tais tabelas apresentam o número de gerações utilizado, o tempo que demorou para sua execução e os resultados obtidos nas vinte execuções, como demonstrado na seção seguinte.

Para realizar a comparação e buscar algumas conclusões quanto a eficiência do NEAT, cada base escolhida foi executada por um algoritmo de alta performance na classificação de indivíduos, o KNN. Tal programa rodou com um valor padrão de K=3 vizinhos mais próximos, e sua acurácia foi registrada para fins de comparação.

Por fim, uma das bases foi selecionada para o estudo de evolução do NEAT. Tal base teria seus parâmetros e configurações constantemente alterados, visando comprovar a possibilidade de evolução do algoritmo a partir de alterações em sua configuração.

A partir do processo e estudo explicado nesta seção, as próximas fases do trabalho consistem na Análise e Conclusão, ambas especificadas nas seções a seguir: Resultados e Conclusões.

## **4. Resultados**

### **4.1 Rodar as cinco bases no algoritmo NEAT e KNN**

Para fazer o uso do algoritmo NEAT, inicialmente é necessário configurar certas variáveis, dentre as quais:

- Número de entradas (inputs);
- Número de saídas (outputs);

- Quantidade de indivíduos que serão criados em cada geração (pop\_size);
- Porcentagem de indivíduos que serão preservados para a próxima geração (survival threshold);
- Facilidade dos indivíduos se agruparem em espécies (compatibility threshold);
- Quantidade de indivíduos que serão mantidos dentro de cada espécie (elitism);
- Limite de gerações que o será mantida uma espécie que não está evoluindo (max stagnation)

Como explicado previamente, a acurácia é calculada com base nos resultados obtidos pelo NEAT em relação ao esperado pela instância da base. Exemplificando a situação, ao rodar o algoritmo com uma base de cem instâncias, um número de ‘Outputs’ é gerado, assim como um número de valores ‘Expected’. Se trinta desses valores esperados forem diferentes das saídas geradas pelo algoritmo, o programa resultaria em uma acurácia de 70%.

As cinco bases selecionadas foram submetidas ao algoritmo NEAT e executadas ao menos vinte vezes para cada conjunto de gerações. As acurácias obtidas em ambos os códigos foram registradas em tabelas específicas para cada uma delas, assim como o tempo médio de sua execução em segundos.

Além de apresentar a Pior e Melhor acurácia obtida nos testes, as tabelas também demonstram as vinte acurácias obtidas em sequência, assim como o resultado obtido ao executá-las no algoritmo KNN.

Para o KNN, estabeleceu-se o  $K = 3$  devido a padronização de seu uso em outros cenários. Na busca de outliers, seu algoritmo foi executado cinco vezes para cada base, não obtendo variação em seu resultado. A estratégia de execução utilizada é popularmente conhecida por ‘conjunto de teste’, em que os dados de entrada foram divididos entre “treino”, para treinar o algoritmo, e “teste”, para fazer a verificação de seu aprendizado.

**Tabela 1. Resultados obtidos com a base de dados “Breast Cancer”**

Breast Cancer (570 linhas / 32 colunas)					
Gerações	Pior Acuracia	Melhor Acuracia	Tempo Médio	Resultados (Na ordem de execução)	Base no KNN com K = 3
100	59%	93%	53s	[93, 79, 78, 92, 78, 79, 79, 92, 92, 78, 92, 78, 92, 78, 78, 59, 92, 79, 79, 92, 78, 92]	90.845
1000	59%	94%	491s	[93, 93, 61, 93, 93, 60, 93, 93, 93, 93, 93, 93, 94, 80, 60, 79, 59, 93, 92, 93, 80]	
10000	92%	95%	67059 s	[95, 95, 94, 93, 94, 95, 95, 94, 95, 95, 94, 95, 92, 94, 94, 95, 92, 94, 95, 95]	

**Tabela 2. Resultados obtidos com a base de dados “Haberman”**

Haberman (306 linhas / 4 colunas)					
Gerações	Pior Acuracia	Melhor Acuracia	Tempo Médio	Resultados (Na ordem de execução)	Base no KNN com K = 3
100	74%	76%	260s	[74, 76, 74, 74, 74, 74, 75, 74, 74, 76, 74, 74, 74, 74, 75, 74, 74, 75, 74, 74]	84.96
1000	74%	76%	2805s	[75, 75, 75, 75, 75, 76, 76, 75, 75, 75, 76, 76, 76, 74, 75, 75, 75, 75, 75]	
10000	75%	77%	41010s	[76, 77, 75, 75, 75, 76, 75, 76, 75, 75, 75, 76, 76, 75, 75, 75, 75, 76, 75, 75]	

**Tabela 3. Resultados obtidos com a base de dados “Tic-Tac-Toe”**



Tic-Tac-Toe (958 linhas / 10 colunas)					
Gerações	Pior Acuracia	Melhor Acuracia	Tempo Médio	Resultados (Na ordem de execução)	Base no KNN com K = 3
100	65%	71%	891s	[71, 65, 71, 71, 71, 71, 65, 71, 71, 71, 71, 65, 65, 71, 71, 65, 65, 71, 71, 65, 65, 71, 71, 65, 65, 71, 71, 71]	81.0
1000	59%	69%	9778s	[65, 65, 65, 69, 61, 65, 69, 65, 65, 65, 65, 65, 59, 65, 65, 65, 67, 65, 65, 65]	
10000	65%	70%	105859 s	[70, 65, 65, 65, 65, 65, 67, 65, 65, 65, 65, 68, 65, 65, 65, 67, 65, 65, 65, 65]	

**Tabela 4. Resultados obtidos com a base de dados “Fertility”**

Fertility (101 linhas / 9 colunas)					
Gerações	Pior Acuracia	Melhor Acuracia	Tempo Médio	Resultados (Na ordem de execução)	Base no KNN com K = 3
100	89%	93%	124s	[93, 92, 89, 92, 92, 93, 91, 93, 92, 92, 92, 91, 92, 92, 92, 90, 93, 91, 92]	91.0
1000	92%	95%	1624s	[93, 94, 94, 93, 93, 94, 92, 94, 93, 94, 95, 93, 93, 94, 94, 94, 94, 95, 93]	
10000	94%	95%	24951s	[95, 95, 95, 95, 95, 94, 95, 95, 95, 95, 95, 95, 95, 95, 95, 95, 95, 95, 95, 95]	

**Tabela 5. Resultados obtidos com a base de dados “Caesarean section”**

Caesarian Section (80 linhas / 5 colunas)					
Gerações	Pior Acuracia	Melhor Acuracia	Tempo Médio	Resultados (Na ordem de execução)	Base no KNN com K = 3
100	65%	72%	72s	[68, 65, 65, 65, 65, 68, 65, 65, 72, 68, 68, 65, 70, 68, 65, 68, 65, 65, 65, 65]	81.25
1000	65%	71%	1004s	[68, 70, 68, 68, 68, 65, 70, 68, 68, 71, 69, 71, 68, 68, 68, 70, 68, 65, 68, 68]	
10000	65%	75%	22414s	[68, 75, 74, 74, 72, 74, 75, 65, 75, 71, 74, 75, 72, 75, 74, 72, 72, 74, 68]	

Em uma observação inicial, as tabelas ‘Haberman’(Tabela 2) e ‘Tic-Tac-Toe Endgame’(Tabela 3) apresentaram uma estagnação em seus resultados, não apresentando uma evolução em relação ao aumento do número de gerações, como evidenciado nas outras tabelas.

Para melhor analisar o estudo do algoritmo NEAT em tais cenários de ‘não evolução’, uma das bases foi selecionada para testar o código com configurações modificadas. A teoria inicial para tal decisão consistia na possibilidade do algoritmo evoluir dependendo das ‘configs’ atribuídas em suas variáveis, necessitando serem únicas para cada caso, como explicado a seguir.

## 4.2 Modificar configurações de uma das bases

Após obter os resultados das tabelas e verificar uma possível estagnação de duas das bases, selecionou-se uma delas para servir de teste e análise do código NEAT utilizado. O algoritmo teria suas configurações padrões modificadas, de forma a analisar a acurácia resultante de sua execução a partir da base selecionada. “Tic-Tac-Toe Endgame” (Tabela 3) foi a escolhida, o teste consistiu em múltiplas modificações, principalmente nas variáveis de seu arquivo ‘config’.

Dentre tais modificações, os setores a seguir foram o foco principal das alterações na configuração do algoritmo: “DefaultSpeciesSet”, “DefaultStagnation”, “DefaultReproduction” e “NEAT”. Esperava-se que, com a alteração das variáveis de cada setor, além de modificações em outros locais da configuração, a acurácia aumentaria e melhor evoluiria dependendo de cada situação. Após testar diferentes cenários e modificações, a acurácia se manteve estável na base testada, sem demonstrar alterações significativas que justificassem uma possível evolução.

## 5. Conclusões

A partir da observação das tabelas, evidenciam-se algumas que estagnaram em seus

resultados de acurácia. Para esses casos, mesmo aumentando o número de gerações, obtém-se acurácia estáveis e sem evidências de evolução (Tabela 4 e 5), porém, observa-se nas outras bases testadas uma linha crescente de acertos conforme o aumento no número de gerações.

A seguir, a tabela de resumo (tabela 6) apresentando a pior e melhor acurácia de cada base de dados analisada, comparando-as com a acurácia obtida na execução do KNN.

**Tabela 6. Resultados das cinco bases de dados e comparativo KNN = 3**

Datasets	Pior Acuracia	Melhor Acuracia	KNN com K = 3
Breast Cancer (570 linhas / 32 colunas )	59%	95%	90.845
Haberman (306 linhas / 4 colunas )	74%	77%	84.96
Tic-Tac-Toe (958 linhas / 10 colunas )	65%	71%	81.0
Fertility (101 linhas / 9 colunas )	89%	95%	91.0
Caesarian Section (80 linhas / 5 colunas)	65%	75%	81.25

Para certos cenários não estagnados, dentre os vinte resultados calculados para cada conjunto de gerações, encontram-se também diversos ‘outliers’ nas acurácias obtidas (Tabela 1). Sendo valores que fogem do padrão na lista de execuções, tais aparições contribuem para a desconfiança no algoritmo neuroevolutivo estudado, porém, observa-se a diminuição delas conforme o aumento no número de gerações processadas. A partir de tal observação, pode-se constatar que: Quanto maior o número de gerações do algoritmo NEAT, maior a confiabilidade dos resultados, aumentando sua acurácia proporcionalmente com o número de gerações, diminuindo assim a chance de resultar em valores muito distantes das acurácias normalmente encontradas.

Quanto às bases que demonstraram estagnação em seus resultados, mesmo alterando suas variáveis de configuração não foi possível aumentar ou diminuir sua acurácia. O algoritmo mantém-se estável e sem uma alteração significativa, levando a conclusão que: Dependendo da base de classificação selecionada e da distribuição dos indivíduos da mesma, o algoritmo NEAT não consegue evoluir nos cenários estudados.

Analisando os resultados ao executar as mesmas bases no algoritmo KNN, observa-se sua acurácia superando a classificação do algoritmo NEAT em quase todos os cenários. Em duas situações (Tabela 1 e Tabela 2), o algoritmo bio-inspirado foi capaz de se sobressair em sua acurácia, porém, levando um tempo de execução significativamente superior à execução instantânea do KNN.

Comparando ambos os algoritmos, evidencia-se que a baixa necessidade de configurações do KNN, assim como seu tempo para gerar um resultado e a possibilidade de configurá-lo, o tornam uma melhor opção para executar bases de classificação. Apesar de ser possível, em certos casos, obter uma acurácia maior, os cenários de classificação de bases pelo NEAT trazem o risco de gerar resultados ‘outliers’, um alto tempo para sua execução, a necessidade de processar múltiplas gerações e a possibilidade de acurácias estagnadas dependendo da base.

A partir do estudo e execução do algoritmo NEAT em diferentes bases, além da comparação com um algoritmo mais usual no cenário analisado (KNN), observa-se a elevada necessidade de mais gerações e um gradativo poder de processamento para se obter uma melhor acurácia.

Devido a possibilidade de se obter resultados melhores com um algoritmo bem menos custoso, observa-se que, no contexto analisado e com o atual poder computacional disponível para a realização deste trabalho, o NEAT não se demonstra eficiente em problemas de classificação.

## Referências Bibliográficas

- DUA, D.; GRAF, C. *UCI Machine Learning Repository*. CA: University of California, School of Information and Computer Science, Irvine, 2019.
- AMIN, M. Z; ALI, A. Caesarian Section Classification Dataset Data Set, 2018. *UCI Machine Learning Repository*, 2019. Disponível em: <https://archive.ics.uci.edu/ml/datasets/Caesarian+Section+Classification+Dataset>. Acesso em 18 out. 2022.
- GIL, D.; GIRELA, J. L.; DE JUAN, J.; TORRES, J. G; JOHNSON, M. Fertility Data Set, 2013. *UCI Machine Learning Repository*, 2019. Disponível em: <https://archive.ics.uci.edu/ml/datasets/Fertility>. Acesso em 18 out. 2022.
- AHA, D. Tic-Tac-Toe Endgame. *UCI Machine Learning Repository*, 2019. Disponível em: <https://archive-beta.ics.uci.edu/ml/datasets/tic+tac+toe+endgame>. Acesso em 18 out. 2022.
- LIM, T. S. Haberman's Survival Data Set, 1999. *UCI Machine Learning Repository*, 2019. Disponível em: <https://archive.ics.uci.edu/ml/datasets/Haberman%27s+Survival>. Acesso em 18 out. 2022.
- WOLBERG, W. H.; STREET, W. N.; MANGASARIAN, O. L. Breast Cancer Wisconsin (Diagnostic), 1995. Data Set. *UCI Machine Learning Repository*, 2019. Disponível em: <https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Diagnostic%29>. Acesso em 18 out. 2022.
- LIPPMANN, R.P. “Uma introdução à computação com redes neurais”. IEEE ASSP Magazine, 1987.
- CHEN, L.; DAMMINDA, A. “NeuroEvolution of Augmenting Topologies with Learning for Data Classification”. International Conference on Information and Automation, 2006.

- STANLEY, K. O.; MIIKKULAINEN, R. “*Evolving Neural Networks through Augmenting Topologies*”. The University of Texas at Austin, 2002.
- STANLEY, K. O.; MIIKKULAINEN, R. “*Efficient Evolution of Neural Network Topologies*”. The University of Texas at Austin, 2002.
- OSÓRIO, F. S., Canoas, RS. “*Redes Neurais Artificiais: do Aprendizado Natural ao Aprendizado Artificial*”. ULBRA Canoas, Rio Grande do Sul, 1999.
- MATTHEW, C.; MENG, Y. “Toward Evolving Neural Networks using Bio-Inspired Algorithms”. 2008.
- MITCHELL, M.; “An introduction to genetic algorithms”. Massachusetts: MIT, 1996.
- MCINTYRE, A.; KALLADA, M; MIGUEL, C. G.; DA SILVA C. F.; NETTO, M. L. NEAT-python, 2015. Disponível em: <https://github.com/CodeReclaimers/neat-python>. Acesso em 18 out. 2022.
- STANLEY, K. O. "Neuroevolution: A different kind of deep learning". *O'Reilly Media*, 2017.
- RISI, S.; TOGELIUS, J. “Neuroevolution in Games: States of the Art and Open Challenges”. *IEEE Transactions on Computational Intelligence and AI in Games*, 2017.
- KASSAHUN, Y.; SOMMER, G.; EDGINGTON, M.; METZEN, J. H.; KIRCHNER, F. “Common genetic encoding for both direct and indirect encodings of networks”. *Genetic and Evolutionary Computation Conference*, 2007.
- TURING, A. M. “On Computable Numbers, with an Application to the Entscheidungsproblem”. *Mathematical Society*, 1936.
- HOFSTADTER, D. R. “Godel, Escher, Bach: an eternal golden braid”. *Basic Books*, 1979.
- MCCULLOCH, W.; PITTS, W. “A Logical Calculus of the Ideas Immanent in Nervous Activity”, 1943. *Ideas That Created the Future*, The MIT Press, 2021.
- APPLE. MacBook Air (M1, 2020) - Especificações técnicas. Apple (Brasil), 2022. Disponível em: [https://support.apple.com/kb/SP825?locale=pt\\_BR](https://support.apple.com/kb/SP825?locale=pt_BR). Acesso em 11 nov. 2022.