http://journal.unoeste.br/index.php/ce/index
DOI: 10.5747/ce.2021.v13.n1.e344
ISSN on-line 2178-8332

Colloquium Exactarum

Submetido: 29/09/2020 Revisado: 17/03/2021 Aceito: 01/04/2021

COMPARAÇÃO ENTRE AS REDES NEURAIS ADALINE E PERCEPTRON UTILIZANDO O CONJUNTO DE DADOS IRIS

Comparison between the neural networks Adaline and Perceptron using the IRIS database

Allana dos Santos Campos e César Alberto Bravo Pariente

Universidade Estadual de Santa Cruz – UESC, BA
E-mail: allanacampos50@gmail.com, cabpariente@uesc.br

RESUMO – Inicialmente as redes neurais foram desenvolvidas com o objetivo de criar um sistema computacional que modele o funcionamento do cérebro humano, porém as mesmas passaram a ser utilizadas para resolver tarefas específicas. Adaline e Perceptron são duas redes neurais que calculam uma função de entrada utilizando um conjunto de pesos adaptativos e um bias e, apesar de suas similaridades, é conhecido que o Adaline converge para um resultado mais rapidamente que o Perceptron. Este trabalho foi elaborado como um exercício didático, afim de apresentar como se obtém tais conclusões, utilizando como dados para classificação e treinamento o conjunto de dados IRIS. Ao longo do trabalho utilizou-se as linguagens de programação Processing, para elaboração das redes neurais, e Python para apresentação visual dos resultados. Os resultados encontrados apresentam o desempenho superior da rede neural Adaline sobre o Perceptron, evidenciando as classes do conjunto de dados que podem ser separadas linearmente e as que não podem, a métrica utilizada para avaliar o desempenho entre as redes neurais é definida pela porcentagem de acertos nas classificações dos dados. O Adaline apresentou melhor desempenho na classificação para comprimento e largura da pétala entre as classes Iris-setosa e Iris-virginica dentre todas as demais classificações.
Palavras-chave: Redes Neurais; Adaline; Perceptron.

ABSTRACT – Initially, neural networks were developed with the objective of creating a computational system that models the functioning of the human brain, however they started to be used to solve specific tasks. Adaline and Perceptron are two neural networks that calculate an input function using a set of adaptive weights and a bias, despite their similarities, it is known that the Adaline neural network converges to a result more quickly than the Perceptron neural network. This work was designed as a didactic exercise, in order to present how such conclusions are obtained, using the IRIS database as data for classification and training. Throughout the work, the programming languages Processing, was used to develop neural networks, and Python for visual presentation of results. The results found show the high performance of the Adaline neural network over the Perceptron, showing the database classes that can be linearly separated and those that cannot, the metric used to evaluate the performance between the neural networks is defined by the percentage of correct answers in the data classifications. Adaline showed the best performance in the classification for length and width of the petal

between the Iris-setosa and Iris-virginica classes among all the other classifications.
Keywords: Neural Network; Adaline; Perceptron.

1. INTRODUÇÃO

Redes neurais artificias são sistemas que possuem vários neurônios artificiais interconectados com o propósito de simular o processo de aprendizagem do cérebro humano. Ao implementar uma rede neural artificial, ela se torna capaz de reconhecer padrões e correlações, agrupar e classificar dados, por conta de tais utilidades as redes neurais são principalmente utilizadas para criar sistemas de inteligência artificial.

Uma das redes neurais mais conhecidas é o Perceptron, por possuir uma arquitetura simples é considerado um bom modelo introdutório. Foi desenvolvido nas décadas de 1950 e 1960 pelo cientista Frank Rosenblatt, inspirado em trabalhos anteriores de Warren McCulloch e Walter Pitts [Data Science Academy, 2019].

Um perceptron é um modelo matemático que a partir de vários valores como entrada devolve uma saída binária. Para simular o processo de aprendizagem o perceptron possui pesos sinápticos para cada entrada que são atualizados até que resultem em um valor desejado para a saída.

A rede neural Adaline, do inglês: Adaptive Linear Neuron, foi proposta por Widrow e Hoff, 1960 [Moreira, 2018], e possui a mesma estrutura que o Perceptron, se diferenciando apenas no algoritmo de treinamento conhecido como Regra Delta e possuindo saídas bipolares.

A Regra Delta foi desenvolvida devido à incapacidade do algoritmo de treinamento do Perceptron em gerar pesos que classifiquem dados não linearmente separáveis. O algoritmo para o Adaline realiza iterações locais para obter o ponto mínimo da função de erro e, assim, fixa os valores dos pesos quando o mínimo foi encontrado [Moreira, 2018].

Com intuito de comparar as duas redes neurais, Adaline e Perceptron, este trabalho foi desenvolvido utilizando o conjunto de dados IRIS, que pode ser encontrado em [Dua e Graff, 2019], para treinamento e classificação das redes neurais.

Em [Fisher, 1936] o estatístico e biólogo britânico Ronald Fisher introduziu o conjunto de dados IRIS, que possui 150 amostras de três espécies de Iris, Iris-setosa, Iris-versicolor e Iris-virginica, cada amostra possui quatro características, o comprimento e largura da sépala e o comprimento e largura da pétala.

2. MATERIAIS E MÉTODOS

2.1. Materiais

- Computador Intel® Core™ i7 7500U CPU @ 2.70GHz, 2.90GHz, 8.00GB, 64 bits, Windows 10;
- Python 3.7.4;
- Processing 3.5.4;
- Conjunto de Dados Iris;

2.2. Métodos

Para cada rede neural foram implementados dois algoritmos, que podem ser encontrados em [Autor, 2020], utilizando a linguagem de programação Processing. Os algoritmos foram baseados nos pseudocódigos encontrados no livro de Danilo G. Spatti, Ivan N. da Silva e Rogério A. Flauzino, "Redes Neurais Artificiais Para Engenharias e Ciências Aplicadas" [Silva, 2010], a seguir:

Considere w como o vetor com os valores dos pesos sinápticos. A função 'sinal(u)' implementada foi: se u ≤ -1 então y = 0, caso contrário y = 1.

Figura 1. Pseudocódigo para treinamento da rede neural Perceptron

```
Obter o conjunto de amostras de treinamento {x(k)};
Associar a saida desejada {d(k)} para cada amostra obtida;
Iniciar o vetor w com valores aleatórios pequenos;
Especificar a taxa de aprendizagem {η};
Iniciar o contador de número de épocas {época ← 0};
Repetir as instruções:
        erro ← "inexistente";
        Para todas as amostras de treinamento {x(k), d(k)}; fazer:
                u ← wT * x(k);
                y ← sinal(u);
                Se y ≠ d(k)
                        Então { w ← w + η * (d(k) - y) * x(k)
                                erro ← "existe" }
        época ← época +1;
        Até que: erro = "inexistente"
```

Fonte: (Silva, 2010)

Como a função 'sinal(u)' utilizada não devolve o valor '-1' então na implementação da classificação a amostra x pertence a Classe A se y = 0.

Figura 2. Pseudocódigo para classificação da rede neural Perceptron

Obter uma amostra a ser classificada {**x**};
Utilizar o vetor **w** ajustado durante o treinamento;
Executar as seguintes instruções;
$\mathbf{u} \leftarrow \mathbf{w}^T * \mathbf{x}$;
$y \leftarrow sinal(u)$;
Se y = -1
Então: amostra $\mathbf{x} \in \{Classe\ A\}$
Se y = 1
Então: amostra $\mathbf{x} \in \{Classe\ B\}$

Fonte: (Silva, 2010)

Para o pseudocódigo da Adaline, considere que w é o vetor com os valores dos pesos sinápticos, assim como no algoritmo de treinamento do Perceptron.

Figura 3. Pseudocódigo para treinamento da rede neural Adaline

Obter o conjunto de amostras de treinamento $\{\mathbf{x}^{(k)}\}$;
Associar a saída desejada $\{ d^{(k)} \}$ para casa amostra obtida;
Iniciar o vetor **w** com valores aleatórios pequenos;
Especificar taxa de aprendizagem $\{\eta\}$ e precisão requerida $\{\varepsilon\}$.
Iniciar o contador de número de épocas { época ← 0 };
Repetir as instruções:
$E_{qm}^{anterior} \leftarrow E_{qm}(\mathbf{w})$;
Para todas as amostras de treinamento $\{\mathbf{x}^{(k)}, d^{(k)}\}$, fazer:
$\mathbf{u} \leftarrow \mathbf{w}^T * \mathbf{x}^{(k)}$;
$\mathbf{w} \leftarrow \mathbf{w} + \eta * (d^{(k)} - u) * \mathbf{x}^{(k)}$;
época ← época + 1;
$E_{qm}^{atual} \leftarrow E_{qm}(\mathbf{w})$;
Até que: $| E_{qm}^{atual} - E_{qm}^{anterior} | \leq \varepsilon$

Fonte: (Silva, 2010)

A função do erro quadrático médio (Eqm(w)) encontrada no algoritmo para o treinamento foi baseada no pseudocódigo a seguir:

Figura 4. Pseudocódigo para função do erro quadrático médio da rede neural Adaline

Obter a quantidade de padrões de treinamento {p};
Iniciar a variável E_{qm} com valor zero { $E_{qm} \leftarrow 0$};
Para todas as amostras de treinamento $\{\mathbf{x}^{(k)}, d^{(k)}\}$, fazer:
$\mathbf{u} \leftarrow \mathbf{w}^T * \mathbf{x}^{(k)}$;
$E_{qm} \leftarrow E_{qm} + (d^{(k)} * u)^2$;
$E_{qm} \leftarrow E_{qm} / p$;

Fonte: (Silva, 2010)

Figura 5. Pseudocódigo para classificação da rede neural Adaline

Obter uma amostra a ser classificada {**x**};
Utilizar o vetor **w** ajustado durante o treinamento;
Executar as seguintes instruções:
$\mathbf{u} \leftarrow \mathbf{w}^T * \mathbf{x}$;
$y \leftarrow sinal(u)$;
Se y = -1
Então: amostra $\mathbf{x} \in \{Classe\ A\}$
Se y = 1
Então: amostra $\mathbf{x} \in \{Classe\ B\}$

Fonte: (Silva, 2010)

Foram elaborados três arquivos a partir do conjunto de dados IRIS, um arquivo com todos os valores referentes as classes Iris-setosa e Iris-versicolor (setvers.txt), um arquivo com todos os valores referentes as classes Iris-setosa e Iris-virginica (setvirg.txt) e um arquivo com todos os valores referentes as classes Iris-versicolor e Iris-virginica (virgvers.txt).

Para gerar o conjunto de dados para o treinamento e classificação das redes neurais foi elaborado um programa, 'geradado', utilizando a linguagem de programação Processing, que contabiliza a quantidade de linhas dos arquivos elaborados pelo conjunto de dados IRIS, sorteia um número entre 0 e a quantidade de linhas, copia a linha representada pelo número no arquivo de saída e salva o número sorteado em um vetor para comparar com os próximos sorteios para não repetir a linha.

Como os arquivos de entrada, setvers.txt, setvirg.txt e virgvers.txt, possuem a metade superior de linhas com amostras de uma classe e a metade inferior de linhas com amostras de outra classe, primeiramente o programa sorteia metade das linhas desejadas para o arquivo de saída nas linhas referentes a classe apresentada no início do arquivo e o restante nas linhas referentes a classe apresentada do final do arquivo de entrada. Tal arquivo de saída servirá como arquivo de treinamento. As linhas que não foram sorteadas são copiadas a outro arquivo de saída que servirá como arquivo de classificação.

Para ajudar na comparação e avaliação dos resultados foi elaborado um gráfico 4X4 utilizando a linguagem de programação python, apresentando as combinações das características das amostras encontradas do conjunto de dados. Inicialmente o arquivo com as amostras teve o formato alterado para CSV para que fosse lido com a função 'read_csv' da biblioteca pandas [Python, 2020], as colunas de cada gráfico foram

definidas como as características de cada amostra e com a função 'pairplot' da biblioteca seaborn [Waskom, 2020] foi gerado o gráfico.

Também foram elaborados gráficos utilizando a biblioteca matplotlib [Hunter, 2007] da linguagem de programação python, utilizando os valores dos pesos resultantes do treinamento das redes neurais e a função 'pyplot.plot' para a geração de uma reta, $w_1*x+w_2*y+\theta = 0$, onde w_1 e w_2 representam os valores dos pesos sinápticos e θ o valor do bias, e, os valores resultantes da classificação e a função 'pyplot.scatter' para a geração de pontos no plano.

3. DESENVOLVIMENTO

Os algoritmos para o Adaline e Perceptron apresentam uma única saída como resposta, de modo que classificam apenas duas classes. Como o conjunto de dados IRIS apresenta três classes distintas, Iris-setosa, Iris-versicolor e Iris-virginica, foram definidas três combinações entre as classes para serem classificadas, Iris-setosa e Iris-versicolor, Iris-setosa e Iris-virginica, Iris-versicolor e Iris-virginica.

O conjunto de dados IRIS contém 150 amostras classificadas, 50 amostras de cada classe, para o treinamento das redes neurais sete conjuntos de treinamento foram elaborados contendo de 20 a 80 amostras, com incrementos de 10, que foram escolhidas aleatoriamente do conjunto de dados. O restante das amostras de cada conjunto de dados formaram os sete conjuntos de classificação.

Para o treinamento e classificação das redes neurais, cada conjunto de entrada possui quatro valores que representam o comprimento e largura da sépala e o comprimento e largura da pétala. Com o intuito de visualizar os resultados da classificação no plano, os conjuntos de entrada foram divididos, de modo que sejam treinados e classificados duas características por vez.

Para a saída desejada para cada conjunto de entrada foram utilizados os números 1 e 0 para representar as classes no Perceptron e os números 1 e -1 para representar as classes no Adaline, de forma que na classificação entre Iris-setosa e Iris-versicolor o 1 representa a Iris-setosa e o 0 a Iris-versicolor para o Perceptron e para o Adaline o -1 representa a Iris-versicolor; na classificação entre Iris-setosa e Iris-virginica são os mesmos valores para Iris-setosa e a Iris-virginica é representada pelos mesmos valores da Iris-versicolor, na classificação entre Iris-versicolor e Iris-virginica o valor 1 representa a Iris-virginica e os valores 0 e -1 a Iris-versicolor.

As redes neurais foram treinadas para cada conjunto de dados e com os pesos obtidos no treinamento foram geradas as classificações das amostras. Foram geradas tabelas com o percentual de erros e acertos de cada resultado. Também foram gerados gráficos com os planos de corte e os pontos de cada classificação para comparação das redes neurais.

4. RESULTADOS

Em cada classificação utilizou-se 7 conjuntos de treinamento, nas tabelas a seguir estão o melhor desempenho, com maior porcentagem de acertos, e o pior desempenho, com menor porcentagem de acertos, encontrados em cada classificação. Nos casos em que na mesma classificação foi encontrado o mesmo desempenho para diferentes conjuntos de treinamento sendo ele o pior ou melhor valor encontrado, o valor é adicionado nas tabelas. O(s) melhor(es) desempenho(s) ocupa(m) o início da tabela, enquanto o(s) pior(es) desempenho(s) ocupa(m) o final da tabela

4.1. Perceptron

Tabela 1. Utilizando rede neural Perceptron, classificação para comprimento e largura da sépala das classes Iris-setosa e Iris-versicolor

Conjunto de Treinamento (%)	Acertos (%)	Falsos Setosa (%)	Falsos Versicolor (%)
30.00	100.00	0.00	0.00
70.00	100.00	0.00	0.00
40.00	90.00	10.00	0.00
50.00	90.00	10.00	0.00

Fonte: (Autor, 2020).

Tabela 2. Utilizando rede neural Perceptron, classificação para comprimento e largura da pétala das classes Iris-setosa e Iris-versicolor

Conjunto de Treinamento (%)	Acertos (%)	Falsos Setosa (%)	Falsos Versicolor (%)
20.00	100.00	0.00	0.00
60.00	100.00	0.00	0.00
70.00	100.00	0.00	0.00
80.00	95.00	5.00	0.00

Fonte: (Autor, 2020).

Tabela 3. Utilizando rede neural Perceptron, classificação para comprimento e largura da sépala das classes Iris-setosa e Iris-virgínica

Conjunto de Treinamento (%)	Acertos (%)	Falsos Setosa (%)	Falsos Virginica (%)
70.00	100.00	0.00	0.00
80.00	95.00	5.00	0.00

Fonte: (Autor, 2020).

Tabela 4. Utilizando rede neural Perceptron, classificação para comprimento e largura da pétala das classes Iris-setosa e Iris-virginica

Conjunto de Treinamento (%)	Acertos (%)	Falsos Setosa (%)	Falsos Virginica (%)
40.00	100.00	0.00	0.00
70.00	100.00	0.00	0.00
80.00	100.00	0.00	0.00
20.00	90.00	2.50	7.50

Fonte: (Autor, 2020).

Tabela 5. Utilizando rede neural Perceptron, classificação para comprimento e largura da sépala das classes Iris-versicolor e Iris-virginica

Conjunto de Treinamento (%)	Acertos (%)	Falsos Versicolor (%)	Falsos Virginica (%)
50.00	64.00	26.00	10.00
20.00	43.75	17.50	38.75

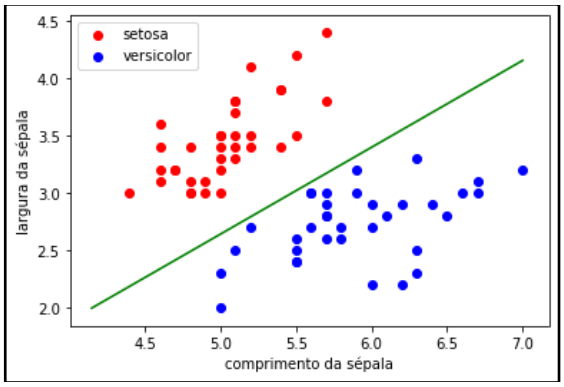
Fonte: (Autor, 2020).

Tabela 6. Utilizando rede neural Perceptron, classificação para comprimento e largura da pétala das classes Iris-versicolor e Iris-virginica

Conjunto de Treinamento (%)	Acertos (%)	Falsos Versicolor (%)	Falsos Virginica (%)
60.00	97.50	0.00	2.50
80.00	80.00	0.00	20.00

Fonte: (Autor, 2020).

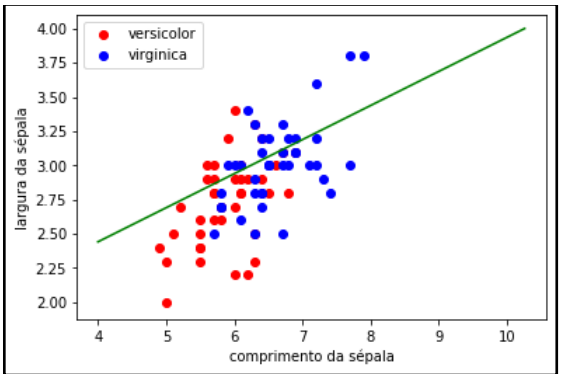
Figura 6. Pesos resultantes para a formação da reta: w0(bias) = 79, w1 = - 52.40036, w2 = 69.20282.



Fonte: (Autor, 2020).

A seguir a Figura 6 apresentando uma classificação com 100% de acertos da rede neural Perceptron, correspondendo a Tabela 1 e a Figura 7 apresentando uma classificação com o pior desempenho da rede neural Perceptron, com 43,75% de acertos, correspondendo a Tabela 5.

Figura 7. Pesos resultantes para a formação da reta: w0(bias) = 148, w1 = 25.499744, w2 = -102.39855.



Fonte: (Autor, 2020).

4.2. Adaline

Tabela 7. Utilizando rede neural Adaline, classificação para comprimento e largura da sépala das classes Iris-setosa e Iris-versicolor

Conjunto de Treinamento (%)	Acertos (%)	Falsos Setosa (%)	Falsos Versicolor (%)
30.00	100.00	0.00	0.00
40.00	100.00	0.00	0.00
80.00	100.00	0.00	0.00
50.00	88.00	12.00	0.00

Fonte: (Autor, 2020).

Tabela 8. Utilizando rede neural Adaline, classificação para comprimento e largura da pétala das classes Iris-setosa e Iris-versicolor

Conjunto de Treinamento (%)	Acertos (%)	Falsos Setosa (%)	Falsos Versicolor (%)
60.00	100.00	0.00	0.00
70.00	100.00	0.00	0.00
80.00	100.00	0.00	0.00
50.00	98.00	2.00	0.00

Fonte: (Autor, 2020).

Tabela 9. Utilizando rede neural Adaline, classificação para comprimento e largura da sépala das classes Iris-setosa e Iris-virginica

Conjunto de Treinamento (%)	Acertos (%)	Falsos Setosa (%)	Falsos Virginica (%)
20.00	100.00	0.00	0.00
30.00	100.00	0.00	0.00
70.00	100.00	0.00	0.00
80.00	90.00	10.00	0.00

Fonte: (Autor, 2020).

Tabela 10. Utilizando rede neural Adaline, classificação para comprimento e largura da pétala das classes Iris-setosa e Iris-virginica

Conjunto de Treinamento (%)	Acertos (%)	Falsos Setosa (%)	Falsos Virginica (%)
20.00	100.00	0.00	0.00
30.00	100.00	0.00	0.00
40.00	100.00	0.00	0.00
50.00	100.00	0.00	0.00
60.00	100.00	0.00	0.00
70.00	100.00	0.00	0.00
80.00	100.00	0.00	0.00

Fonte: (Autor, 2020).

Tabela 11. Utilizando rede neural Adaline, classificação para comprimento e largura da sépala das classes Iris-versicolor e Iris-virginica

Conjunto de Treinamento (%)	Acertos (%)	Falsos Versicolor (%)	Falsos Virginica (%)
30.00	72.90	15.70	11.40
20.00	46.25	11.25	42.50

Fonte: (Autor, 2020).

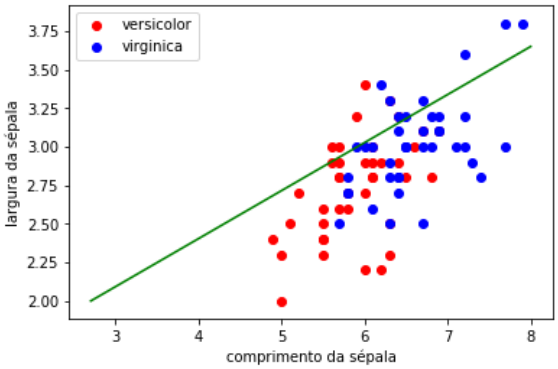
Tabela 12. Utilizando rede neural Adaline, classificação para comprimento e largura da pétala das classes Iris-versicolor e Iris-virginica

Conjunto de Treinamento (%)	Acertos (%)	Falsos Versicolor (%)	Falsos Virginica (%)
40.00	96.70	1.70	1.70
60.00	65.00	0.00	35.00

Fonte: (Autor, 2020).

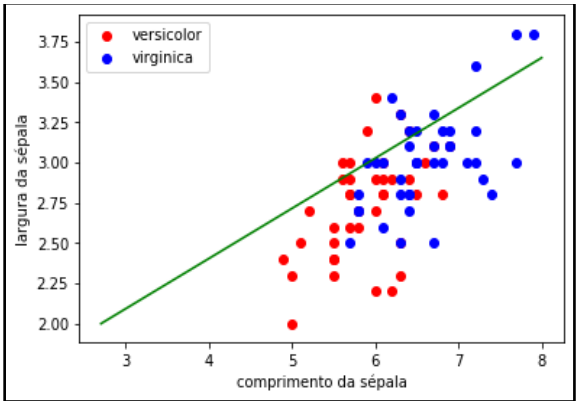
A Figura 8 apresenta uma classificação com 100% de acertos da rede neural Adaline, correspondendo a Tabela 7, e a Figura 9 apresenta uma classificação com o pior desempenho da rede neural Adaline, com 46.25% de acertos, correspondendo a Tabela 11.

Figura 8. Pesos resultantes para a formação da reta: w0(bias) = -0.9220933, w1 = -0.94848824, w2 = 1.3466693.



Fonte: (Autor, 2020)

Figura 9. Pesos resultantes para a formação da reta: w0(bias) = -1.8686919, w1 = 0.5037405, w2 = -1.6156605.



Fonte: (Autor, 2020)

5. CONSIDERAÇÕES FINAIS

Nas classificações para comprimento e largura da pétala observa-se nas Tabelas 2, 4, 8 e 10, que ambas redes neurais obtiveram 100% de acertos para o conjunto de

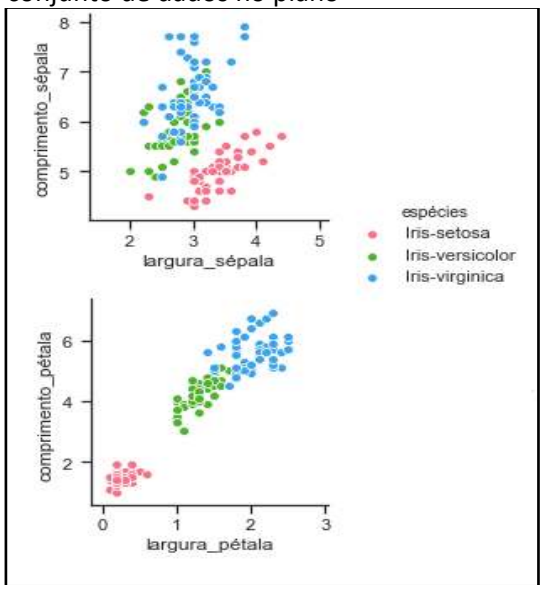
treinamento utilizando 70% das amostras nas classificações entre as classes Iris-setosa e Iris-versicolor e nas classificações entre Iris-setosa e Iris-virginica.

O melhor desempenho foi obtido pela rede neural Adaline nas classificações de comprimento e largura da pétala entre as classes Iris-setosa e Iris-virginica, visto na Tabela 10, onde apresentou 100% de acertos para todas os conjuntos de treinamento.

O Perceptron apresentou melhor desempenho nas classificações de comprimento e largura da pétala entre as classes Iris-setosa e Iris-virginica quando comparadas com as demais classificações, com 100% de acertos para dos conjuntos de treinamento utilizando 40%, 70% e 80% das amostras, vistos na Tabela 4.

Os melhores desempenhos encontrados nas classificações entre as classes Iris-setosa e Iris-virginica ocorreram por conta dos pontos representando tais classes estarem mais afastados entre si do que os pontos representando Iris-setosa e Iris-versicolor ou Iris-virginica e Iris-versicolor. Pode-se visualizar tal fato comparando a posição dos pontos referentes as classes Iris-setosa e Iris-virginica com os pontos da classe Iris-versicolor na Figura 10.

Figura 10. Características das amostras do conjunto de dados no plano



Fonte: (Autor, 2020).

A comparação entre os melhores desempenhos das redes neurais vistos nas Tabelas 4 e 10 mostra que o Adaline classifica melhor que o Perceptron, pois apresentou 100% de acertos para todas os conjuntos de treinamento enquanto o Perceptron apresentou 100% de acertos para 3 conjuntos de treinamento. A comparação entre as Tabelas 1 e 7, também mostra um melhor desempenho do Adaline com 3 conjuntos de treinamento apresentando 100% de acertos enquanto o Perceptron possui apenas 2 conjuntos de treinamento apresentando 100% de acertos.

Ambas as redes neurais apresentaram o pior desempenho para a classificação de comprimento e largura da sépala entre Iris-versicolor e Iris-virginica para o conjunto de treinamento utilizando 20% das amostras, vistos nas Tabelas 5 e 11, onde o Perceptron apresentou 43.75% de acertos e o Adaline 46.25%.

O pior desempenho visto nas classificações entre as classes Iris-versicolor e Iris-virginica deve-se ao fato de que os pontos representando tais classes não podem ser linearmente separados principalmente para os valores de comprimento e largura da sépala, que podem ser visualizados no primeiro gráfico da Figura 10.

A comparação entre os piores desempenhos das redes neurais vistos nas Tabelas 5 e 11 mostra que o Perceptron classifica pior que o Adaline, onde o melhor desempenho apresentado pelo Perceptron em sua pior classificação é de apenas 64.00% de acertos enquanto o Adaline apresentou 72.90% de acertos.

Conclui-se que para o experimento realizado considerando como valor decisivo a quantidade de acertos totais entre as redes neurais, o Adaline teve melhor desempenho frente ao Perceptron. Sendo que o Adaline apresentou melhor desempenho na classificação para comprimento e largura da pétala entre as classes Iris-setosa e Iris-virginica dentre todas as demais classificações.

REFERÊNCIAS

Autor (2020). **Redes Neurais**. Disponível em: https://github.com/AllanaCampos/Redes-Neurais**.** Acesso: 19 ago. 2020.

Data Science Academy (2019). **Deep Learning Book**. Disponível em: http://www.deeplearningbook.com.br/. Acesso: 02 jul. 2020.

Dua, D. e Graff, C. (2019). **UCI Machine Learning Repository**. Irvine, CA: University of California, School of Information and Computer Science. Disponível em: <**http://archive.ics.uci.edu/ml**>. Acesso: 02 jul. 2020.

HUNTER, J.. *et al.* **Matplotlib:** Visualization with Python. 2020. Disponível em: https://matplotlib.org. Acesso: 02 jul. 2020.

MATOS, M. de. *et al.* **Implementação de redes neurais perceptron e adaline em ambiente labview**. Disponível em: http://repositorio.roca.utfpr.edu.br/jspui/bitstream/1/10215/1/PG_COELE_2018_1_04.pdf. Acesso: 02 jul. 2020.

MOREIRA, S. **Rede Neural Perceptron Adaline**. 2018. Disponível em: https://medium.com/ensina-ai/rede-neural-perceptron-adaline-8f69dc419d4e**.** Acesso: 17 Jul. 2020.

PYTHON (2020). **Pandas**. Disponível em: https://pypi.org/project/pandas/ Acesso em: 17 jul. 2020.

FISHER, R. A. The Use of Multiple Measurements in Taxonomic Problems. **Annals of Eugenics**, v. 7, n. 2, p. 179–188., 1936. Disponível em: http://www.comp.tmu.ac.jp/morbier/R/Fisher-1936-Ann._Eugen.pdf. Acesso: 25 jul. 2020. https://doi.org/10.1111/j.1469-1809.1936.tb02137.x

SHARMA, R. **SIGGRAPH Now | Hands-on Workshop: Machine Learning and Neural Networks – Lecture 3**. 2020. Disponível em: <**https://www.youtube.com/watch?v=anASnKYImSI&t=1670s**>. Acesso: 02 jul. 2020.

Silva, IN da, Danilo Hernane Spatti, e Rogério Andrade Flauzino (2010). **Redes neurais artificiais para engenharia e ciências aplicadas**. São Paulo: Artliber.

Waskom, Michael (2020). **Seaborn**. Disponível em: <**https://seaborn.pydata.org**>. Acesso: 17 jul. 2020.