



**Universidade
Federal de
Uberlândia**

Computação Bioinspirada

Implementação de um perceptron para classificar dados da
base Iris

Ellen Christina Amaral Santana - 12011BSI208

Uberlândia, 2023

Sumário

Sumário	2
Introdução	3
O Código	3
Single Layer Perceptron com 2 classes	3
Treinamento da rede neural usando diferentes percentuais da base original	4
Utilizando 20% da base original no treinamento da rede neural	4
Utilizando 50% da base original no treinamento da rede neural	5
Utilizando 80% da base original no treinamento da rede neural	5
Conclusão sobre percentual de utilização da base para treinamento	6
Diferentes valores para a taxa de aprendizagem (η)	7
Taxa de aprendizagem de 0,05 e treinamento com 20% da base	7
Taxa de aprendizagem de 0,05 e treinamento com 50% da base	7
Conclusão sobre os diferentes valores da taxa de aprendizagem	8
Diferentes valores para o total de iterações (epoch)	9
Total de interações igual à 50	9
Total de interações igual à 200	10
Conclusão sobre os diferentes valores do total de integrações	11
Single Layer Perceptron com três classes	11

Introdução

O objetivo deste trabalho é explorar a implementação e os diferentes parâmetros da rede neural perceptron. Para isso, será utilizada a base de dados Iris, disponível no link (<https://archive.ics.uci.edu/dataset/53/iris>). Trata-se de um conjunto de dados de uma espécie de planta florida. Essa base contém as características de três variedades de íris: Setosa, Versicolour e Virginica. Para cada variedade, são apresentadas quatro características: comprimento da sépala (sepal length), largura da sépala (sepal width), comprimento da pétala (petal length) e largura da pétala (petal width). A partir dessas características, é possível classificar as plantas em uma das três variedades.

Inicialmente, iremos implementar uma rede neural perceptron de uma camada (single layer perceptron) e utilizaremos duas classes de íris para fazer sua classificação. A rede será treinada com diferentes porcentagens da base original (20%, 50% e 80%) tentando prever o restante. Também iremos avaliar diferentes valores da taxa de aprendizado (η) e diferentes valores para o total de iterações (epoch).

Em seguida, vamos incluir uma terceira classe da íris nos experimentos. A rede será treinada utilizando diferentes porcentagem da base original (20%, 50% e 80%) tentando prever o restante. Também utilizará diferentes valores da taxa de aprendizado (η) e diferentes valores para o total de iterações (epoch).

Utilizaremos os resultados obtidos em cada teste para avaliar como chegar ao melhor resultado.

O Código

Para realizar os testes a seguir foi utilizado o código em R que pode ser acessado pelo meu repositório no GitHub através do link (https://github.com/Ellen172/UFU-CompBio/blob/main/trabalho2/perceptron_iris.R).

Os Resultados

Todos os resultados obtidos podem ser visualizados através da tabela a seguir, pelo link (https://1drv.ms/x/s!AkWz4NO3x7obiXd_KTDeUbnlcmf3?e=98XlAj).

Single Layer Perceptron com 2 classes

Para os testes a seguir utilizaremos apenas as classes: Iris-setosa e Iris-versicolor, o que deixa o banco com 100 registros ao todo. No código, definimos a classe Iris-setosa como 1 e a classe Iris-versicolor como -1.

Treinamento da rede neural usando diferentes percentuais da base original

Para os testes a seguir, utilizaremos os seguintes valores fixos:

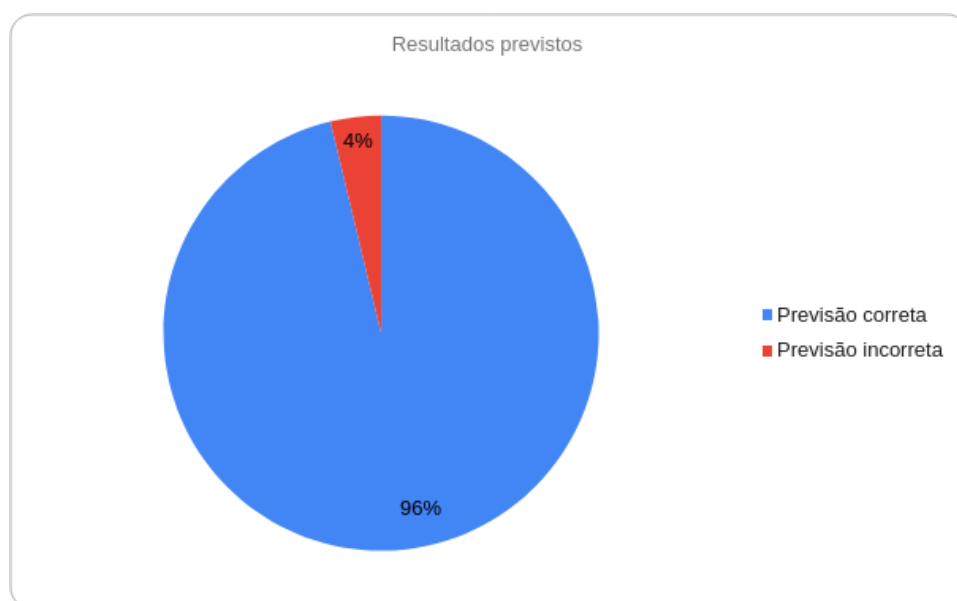
- taxa de aprendizado (η) = 0,01;
- total de iterações (epoch) = 100.

Realizaremos testes utilizando 20% da base para o treinamento, depois, 50% da base e por fim, 80% da base.

Utilizando 20% da base original no treinamento da rede neural

Tendo a base 100 registros, utilizamos 20 para o treinamento e tentamos prever a classe dos 80 restantes. O programa foi executado 3 vezes. Abaixo, temos os resultados obtidos:

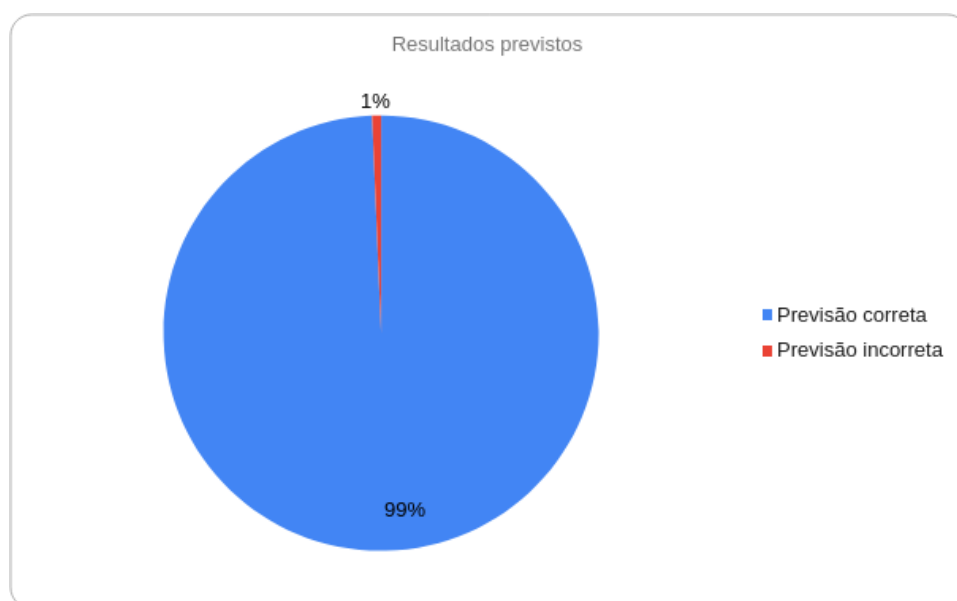
	soma	execução 1	execução 2	execução 3
Previsão correta	231,00	74	80	77
Previsão incorreta	9,00	6	0	3



Utilizando 50% da base original no treinamento da rede neural

Tendo a base 100 registros, utilizamos 50 para o treinamento e tentamos prever a classe dos 50 restantes. O programa foi executado 3 vezes. Abaixo, temos os resultados obtidos:

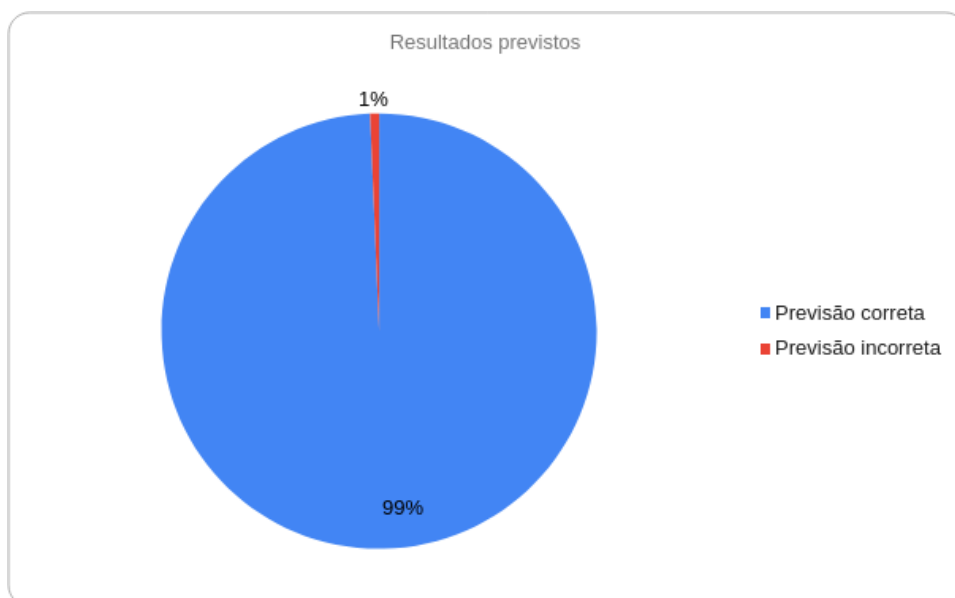
	soma	execução 1	execução 2	execução 3
Previsão correta	149,00	49	50	50
Previsão incorreta	1,00	1	0	0



Utilizando 80% da base original no treinamento da rede neural

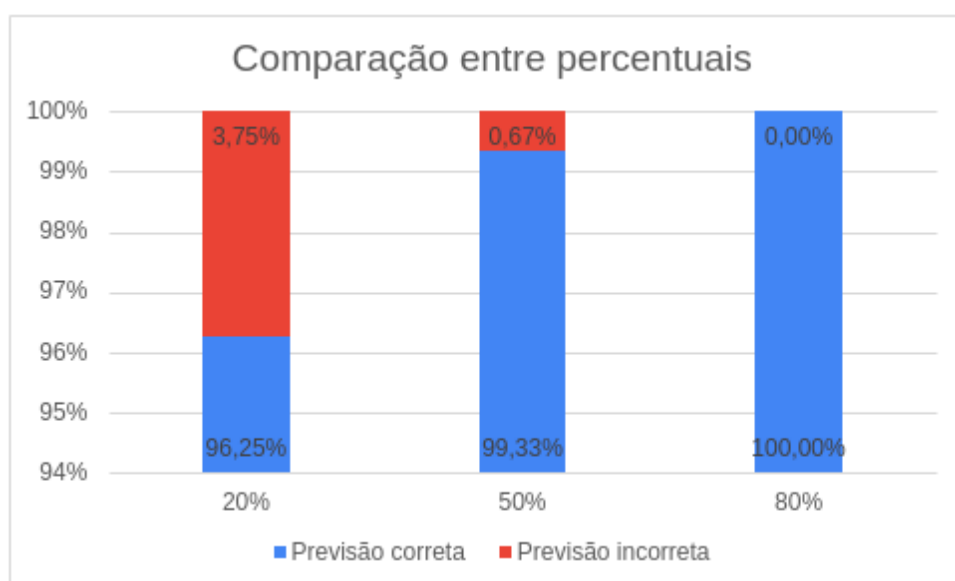
Tendo a base 100 registros, utilizamos 80 para o treinamento e tentamos prever a classe dos 20 restantes. O programa foi executado 3 vezes. Abaixo, temos os resultados obtidos:

	soma	execução 1	execução 2	execução 3
Previsão correta	60,00	20	20	20
Previsão incorreta	0,00	0	0	0



Conclusão sobre percentual de utilização da base para treinamento

De acordo com os resultados obtidos, alterando apenas o percentual de utilização da base usada para o treinamento da rede neural, as previsões com maior número de acertos foram encontradas quando utilizamos 80% da base para o treinamento, seguidos pela utilização de 50% da base e, por fim, 20% da base. Portanto, podemos concluir que quanto maior o percentual utilizado para o treinamento da rede neural, melhor ela será em prever os resultados corretamente.



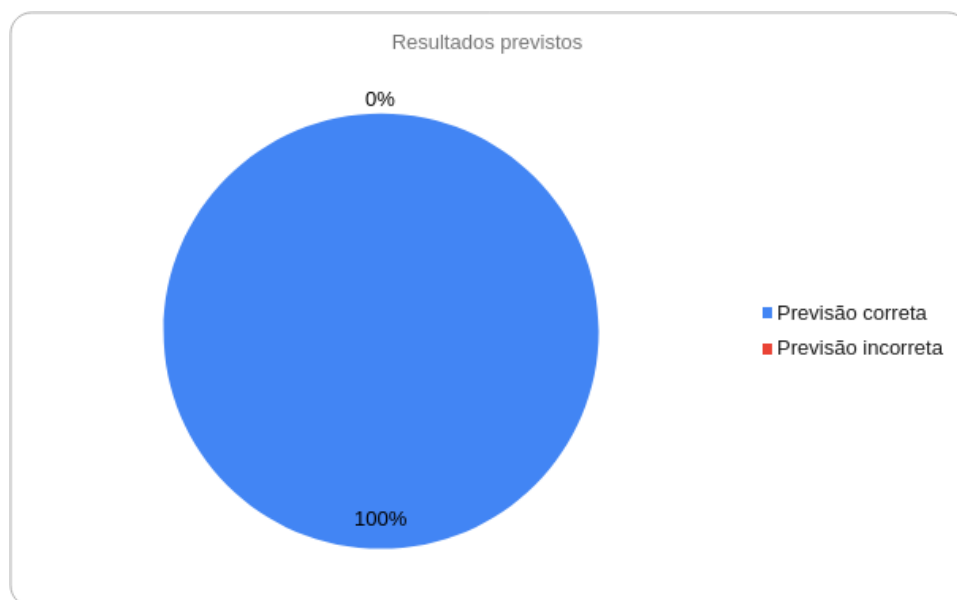
Diferentes valores para a taxa de aprendizagem (η)

Para testar os diferentes valores da taxa de aprendizagem, utilizaremos os valores obtidos no teste anterior (Treinamento da rede neural usando diferentes percentuais da base original) com a taxa igual à 0,01. Iremos testar também a taxa de aprendizagem igual a 0,05, utilizando 20% da base para o treinamento e 50% da base para o treinamento.

Taxa de aprendizagem de 0,05 e treinamento com 20% da base

Com a taxa de aprendizagem igual à 0,05 e o percentual utilizado para o treinamento da rede neural igual à 20%, os resultados obtidos foram:

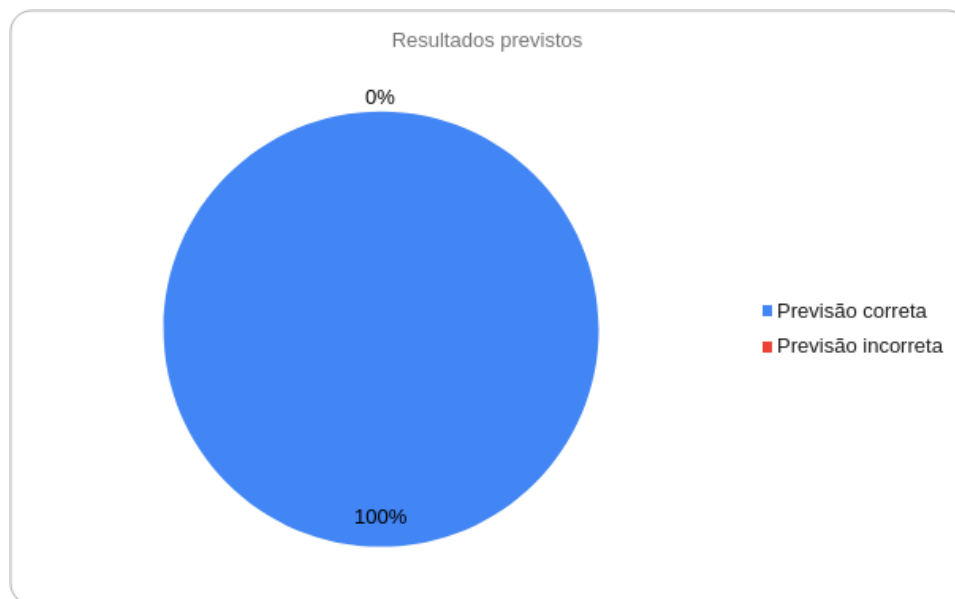
	soma	execução 1	execução 2	execução 3
Previsão correta	240,00	80	80	80
Previsão incorreta	0,00	0	0	0



Taxa de aprendizagem de 0,05 e treinamento com 50% da base

Com a taxa de aprendizagem igual à 0,05 e o percentual utilizado para o treinamento da rede neural igual à 50%, os resultados obtidos foram:

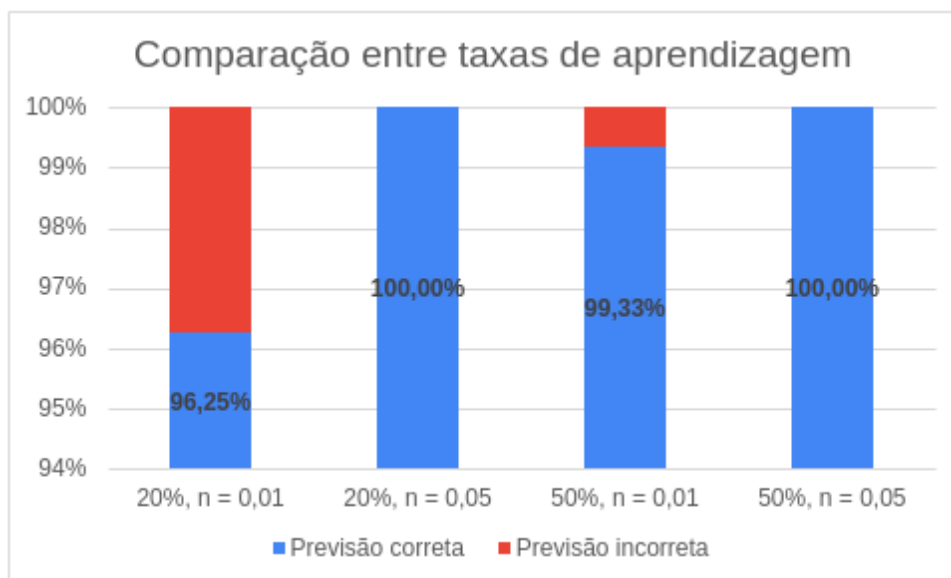
	soma	execução 1	execução 2	execução 3
Previsão correta	150,00	50	50	50
Previsão incorreta	0,00	0	0	0



Conclusão sobre os diferentes valores da taxa de aprendizagem

Observando os resultados obtidos, chegamos à conclusão de que à medida que aumenta a taxa de aprendizagem, aumenta também o percentual de acertos em relação às previsões realizadas pela rede.

	20%, n = 0,01	20%, n = 0,05	50%, n = 0,01	50%, n = 0,05
Previsão correta	96,25%	100,00%	99,33%	100,00%
Previsão incorreta	3,75%	0,00%	0,67%	0,00%
Total	240	240	150	150



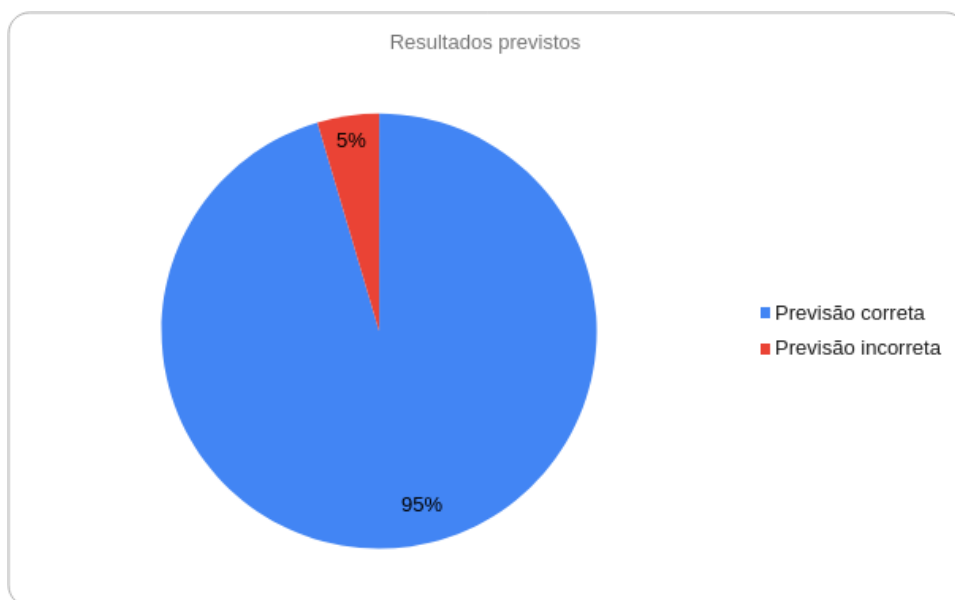
Diferentes valores para o total de iterações (epoch)

Para os testes com diferentes valores para o epoch, utilizaremos os resultados obtidos anteriormente, com epoch igual à 100 e iremos testar também com os valores 50 e 200. Utilizaremos a taxa de aprendizagem igual à 0,01 e 20% da base no treinamento.

Total de interações igual à 50

Utilizando o total de interação igual à 50, obtivemos os seguintes resultados:

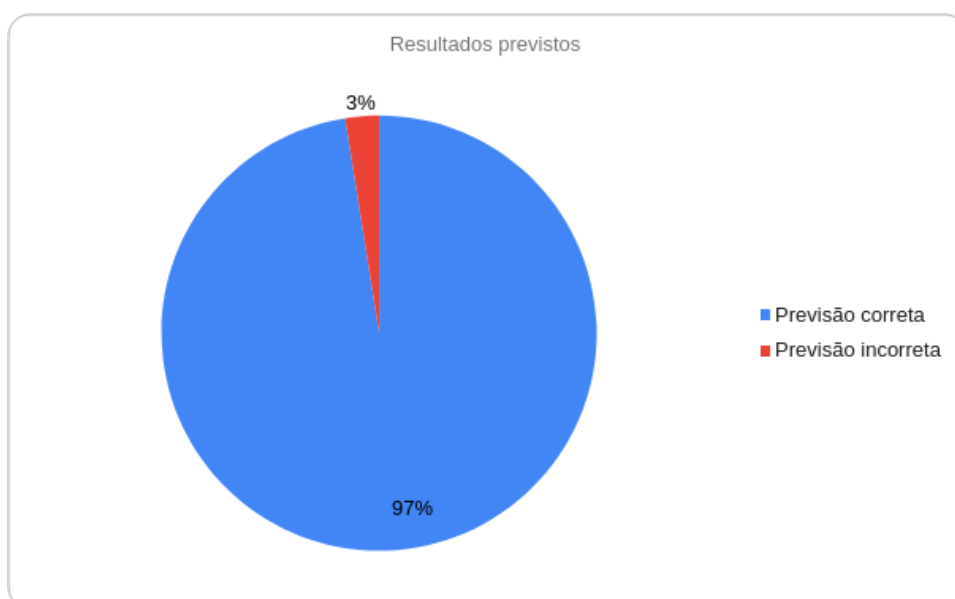
	soma	execução 1	execução 2	execução 3
Previsão correta	229,00	80	78	71
Previsão incorreta	11,00	0	2	9



Total de interações igual à 200

Utilizando o total de interação igual à 200, obtivemos os seguintes resultados:

	soma	execução 1	execução 2	execução 3
Previsão correta	234,00	79	80	75
Previsão incorreta	6,00	1	0	5



Conclusão sobre os diferentes valores do total de integrações

Comparando os resultados obtidos para os diferentes valores do total de integração, podemos perceber que quanto maior seu valor, maior a precisão de acerto quanto às previsões da rede neural.

	50	100	200
Previsão correta	95,42%	96,25%	97,50%
Previsão incorreta	4,58%	3,75%	2,50%
Total	240	240	240

