

Duas classes (Iris-setosa e Iris-versicolor)

Proporção de Treinamento = 10%	Taxa de aprendizado $\eta = 0.001$	Total de iterações = 100	Precisão 98.89%
Proporção de Treinamento = 10%	Taxa de aprendizado $\eta = 0.001$	Total de iterações = 1000	Precisão 98.89%
Proporção de Treinamento = 10%	Taxa de aprendizado $\eta = 0.001$	Total de iterações = 10000	Precisão 98.89%
Proporção de Treinamento = 10%	Taxa de aprendizado $\eta = 0.01$	Total de iterações = 100	Precisão 98.89%
Proporção de Treinamento = 10%	Taxa de aprendizado $\eta = 0.01$	Total de iterações = 1000	Precisão 98.89%
Proporção de Treinamento = 10%	Taxa de aprendizado $\eta = 0.01$	Total de iterações = 10000	Precisão 98.89%
Proporção de Treinamento = 10%	Taxa de aprendizado $\eta = 0.1$	Total de iterações = 100	Precisão 98.89%
Proporção de Treinamento = 10%	Taxa de aprendizado $\eta = 0.1$	Total de iterações = 1000	Precisão 98.89%
Proporção de Treinamento = 10%	Taxa de aprendizado $\eta = 0.1$	Total de iterações = 10000	Precisão 98.89%
Proporção de Treinamento = 30%	Taxa de aprendizado $\eta = 0.001$	Total de iterações = 100	Precisão 100.00%
Proporção de Treinamento = 30%	Taxa de aprendizado $\eta = 0.001$	Total de iterações = 1000	Precisão 100.00%
Proporção de Treinamento = 30%	Taxa de aprendizado $\eta = 0.001$	Total de iterações = 10000	Precisão 100.00%
Proporção de Treinamento = 30%	Taxa de aprendizado $\eta = 0.01$	Total de iterações = 100	Precisão 100.00%
Proporção de Treinamento = 30%	Taxa de aprendizado $\eta = 0.01$	Total de iterações = 1000	Precisão 100.00%
Proporção de Treinamento = 30%	Taxa de aprendizado $\eta = 0.01$	Total de iterações = 10000	Precisão 100.00%
Proporção de Treinamento = 30%	Taxa de aprendizado $\eta = 0.1$	Total de iterações = 100	Precisão 100.00%
Proporção de Treinamento = 30%	Taxa de aprendizado $\eta = 0.1$	Total de iterações = 1000	Precisão 100.00%
Proporção de Treinamento = 30%	Taxa de aprendizado $\eta = 0.1$	Total de iterações = 10000	Precisão 100.00%
Proporção de Treinamento = 50%	Taxa de aprendizado $\eta = 0.001$	Total de iterações = 100	Precisão 100.00%
Proporção de Treinamento = 50%	Taxa de aprendizado $\eta = 0.001$	Total de iterações = 1000	Precisão 100.00%
Proporção de Treinamento = 50%	Taxa de aprendizado $\eta = 0.001$	Total de iterações = 10000	Precisão 100.00%
Proporção de Treinamento = 50%	Taxa de aprendizado $\eta = 0.01$	Total de iterações = 100	Precisão 100.00%
Proporção de Treinamento = 50%	Taxa de aprendizado $\eta = 0.01$	Total de iterações = 1000	Precisão 100.00%
Proporção de Treinamento = 50%	Taxa de aprendizado $\eta = 0.01$	Total de iterações = 10000	Precisão 100.00%
Proporção de Treinamento = 50%	Taxa de aprendizado $\eta = 0.1$	Total de iterações = 100	Precisão 100.00%
Proporção de Treinamento = 50%	Taxa de aprendizado $\eta = 0.1$	Total de iterações = 1000	Precisão 100.00%
Proporção de Treinamento = 50%	Taxa de aprendizado $\eta = 0.1$	Total de iterações = 10000	Precisão 100.00%

Com base nos resultados do experimento, podemos observar que o desempenho do perceptron na classificação das classes "Iris-setosa" e "Iris-versicolor" da base de dados Iris é notável. As porcentagens de dados de treinamento usadas foram 10%, 30% e 50%, e a precisão alcançada foi surpreendentemente alta.

Ao usar apenas 10% dos dados para treinamento, o perceptron ainda alcançou uma precisão de 98.89%, o que é muito impressionante. Isso sugere que as características selecionadas são altamente discriminantes entre as duas classes.

Ao aumentar a porcentagem de dados de treinamento para 30% e 50%, a precisão atinge 100% em ambos os casos. Isso indica que, com mais exemplos de treinamento, o perceptron pode aprender ainda melhor a fronteira de decisão entre as duas classes, tornando-se capaz de classificar com precisão os exemplos de teste.

É importante notar que isso pode ser devido à natureza das classes escolhidas, que podem ser linearmente separáveis no espaço de características.

Ao adicionar a terceira classe (Iris-virginica)

Proporção de Treinamento = 10%	Taxa de aprendizado $\eta = 0.001$	Total de iterações = 100	Precisão 67.41%
Proporção de Treinamento = 10%	Taxa de aprendizado $\eta = 0.001$	Total de iterações = 1000	Precisão 67.41%
Proporção de Treinamento = 10%	Taxa de aprendizado $\eta = 0.001$	Total de iterações = 10000	Precisão 67.41%
Proporção de Treinamento = 10%	Taxa de aprendizado $\eta = 0.01$	Total de iterações = 100	Precisão 67.41%
Proporção de Treinamento = 10%	Taxa de aprendizado $\eta = 0.01$	Total de iterações = 1000	Precisão 67.41%
Proporção de Treinamento = 10%	Taxa de aprendizado $\eta = 0.01$	Total de iterações = 10000	Precisão 67.41%
Proporção de Treinamento = 10%	Taxa de aprendizado $\eta = 0.1$	Total de iterações = 100	Precisão 67.41%
Proporção de Treinamento = 10%	Taxa de aprendizado $\eta = 0.1$	Total de iterações = 1000	Precisão 67.41%
Proporção de Treinamento = 10%	Taxa de aprendizado $\eta = 0.1$	Total de iterações = 10000	Precisão 67.41%
Proporção de Treinamento = 30%	Taxa de aprendizado $\eta = 0.001$	Total de iterações = 100	Precisão 68.57%
Proporção de Treinamento = 30%	Taxa de aprendizado $\eta = 0.001$	Total de iterações = 1000	Precisão 68.57%
Proporção de Treinamento = 30%	Taxa de aprendizado $\eta = 0.001$	Total de iterações = 10000	Precisão 68.57%
Proporção de Treinamento = 30%	Taxa de aprendizado $\eta = 0.01$	Total de iterações = 100	Precisão 68.57%
Proporção de Treinamento = 30%	Taxa de aprendizado $\eta = 0.01$	Total de iterações = 1000	Precisão 68.57%
Proporção de Treinamento = 30%	Taxa de aprendizado $\eta = 0.01$	Total de iterações = 10000	Precisão 68.57%
Proporção de Treinamento = 30%	Taxa de aprendizado $\eta = 0.1$	Total de iterações = 100	Precisão 68.57%
Proporção de Treinamento = 30%	Taxa de aprendizado $\eta = 0.1$	Total de iterações = 1000	Precisão 68.57%
Proporção de Treinamento = 30%	Taxa de aprendizado $\eta = 0.1$	Total de iterações = 10000	Precisão 68.57%
Proporção de Treinamento = 50%	Taxa de aprendizado $\eta = 0.001$	Total de iterações = 100	Precisão 64.00%
Proporção de Treinamento = 50%	Taxa de aprendizado $\eta = 0.001$	Total de iterações = 1000	Precisão 64.00%
Proporção de Treinamento = 50%	Taxa de aprendizado $\eta = 0.001$	Total de iterações = 10000	Precisão 64.00%
Proporção de Treinamento = 50%	Taxa de aprendizado $\eta = 0.01$	Total de iterações = 100	Precisão 64.00%
Proporção de Treinamento = 50%	Taxa de aprendizado $\eta = 0.01$	Total de iterações = 1000	Precisão 64.00%
Proporção de Treinamento = 50%	Taxa de aprendizado $\eta = 0.01$	Total de iterações = 10000	Precisão 64.00%
Proporção de Treinamento = 50%	Taxa de aprendizado $\eta = 0.1$	Total de iterações = 100	Precisão 64.00%
Proporção de Treinamento = 50%	Taxa de aprendizado $\eta = 0.1$	Total de iterações = 1000	Precisão 64.00%
Proporção de Treinamento = 50%	Taxa de aprendizado $\eta = 0.1$	Total de iterações = 10000	Precisão 64.00%

Os resultados mostram que, ao incluir a terceira classe "Iris-virginica" no conjunto de dados, a precisão do modelo de Perceptron diminui em comparação com o caso anterior em que apenas duas classes ("Iris-setosa" e "Iris-versicolor") foram consideradas.

A queda na precisão pode ser explicada pela maior complexidade do problema quando a terceira classe é adicionada. O Perceptron é um classificador linear simples e funciona melhor em problemas linearmente separáveis, onde as classes podem ser separadas por uma linha reta. Quando a classe adicional é introduzida, a separação das classes torna-se mais desafiadora, já que é necessário encontrar um hiperplano que possa separar eficazmente as três classes.

Além disso, ao adicionar mais classes, o conjunto de dados torna-se mais desequilibrado, uma vez que há menos exemplos para cada classe. Isso pode tornar o treinamento do modelo mais suscetível a variações devido à falta de dados de treinamento para algumas classes.

Portanto, a queda na precisão ao adicionar a terceira classe "Iris-virginica" pode ser atribuída à complexidade adicional do problema e à falta de dados de treinamento, tornando a tarefa de classificação mais desafiadora para o modelo de Perceptron. Para abordar esse problema, métodos mais avançados de classificação, como algoritmos de aprendizado de máquina não linear, podem ser necessários.