Redes Neurais Artificais - Exercício Perceptron

Marcio R. A. Souza Filho - 2015104105

Agosto 2020

1 Introdução

Para resolver os exercícios 2 e 3 foi feita a implementação do Perceptron e seu treinamento seguindo exatamente o pseudo algoritmo apresentado nos materiais do professor Antônio P. Braga.

2 Exercício 1

Foram gerados dois grupos de dados de entrada a partir de distribuições normais com média [2,2] para o grupo 1, e média [4,4], para o grupo 2. Para ambos os grupos foi utilizado $sigma = 0.4$ e gerado 200 pontos. Foi considerado a saída igual à 1 para os pontos do grupo 1 e igual à 0 para os pontos do grupo 2.

Os dados de entrada possuem dimensão 2, x_0 e x_1. Portanto, a dimensão do Perceptron também deve ser 2, acrescentado de mais uma dimensão responsável pela polarização do resultado, totalizando dimensão 3. Então, é caracterizado por um vetor de pesos $w = [w_0, w_1, w_2]$, sendo w_0 o termo de polarização.

Nesse exercício o Perceptron não será treinado, ao invés disso será usado um vetor de pesos conhecido para avaliar os pontos de entrada. Esse vetor é:

$$w = [-6, 1, 1]$$

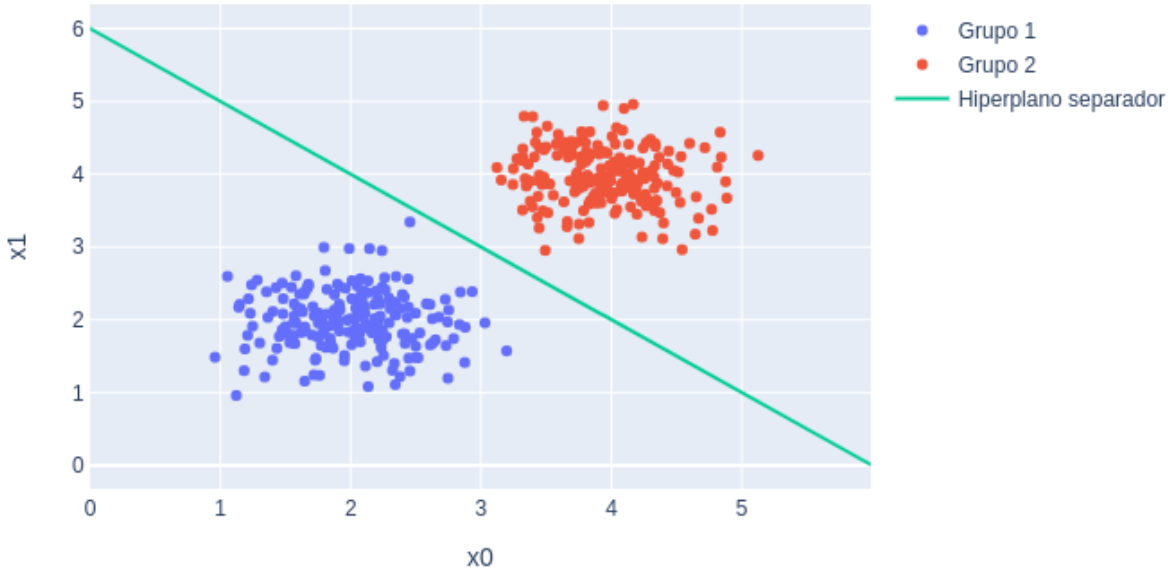

Figura 1: Pontos de entrada e reta de separação.

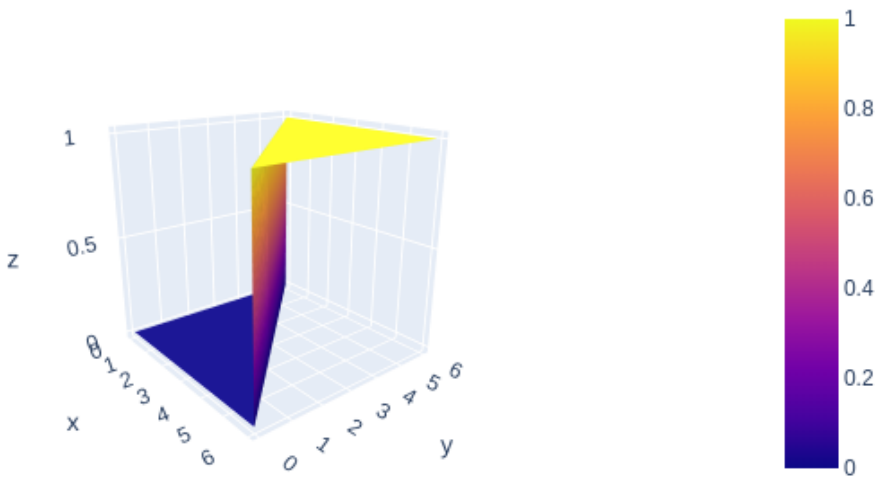

Figura 2: Superfície de separação.

3 Exercício 2

Nesse exercício foram utilizados os mesmos pontos utilizados no exercício anterior.

Os parâmetros definidos para o treinamento do Perceptron foram:

$$tolerancia = 1e-6$$

$$nEpocas = 1000$$

O vetor de pesos w encontrado ao final do treinamento foi:

$$w = [0.35 - 0.05428391 - 0.0556711]$$

À primeira vista esse os pesos obtidos parecem bem diferentes dos pesos utilizados no Exercício 1. Se multiplicarmos o vetor por um escalar, não mudamos a classificação das entradas. Multiplicando o vetor de pesos obtidos por $6/0.35$, obtemos:

$$w' = [1, -0.9306, -0.9543]$$

Utilizando esse novo vetor, fica aparente a similaridade em relação ao vetor de pesos do Exercício 1. A diferença que existe se deve ao fato de que as regiões dos pontos dos grupos 1 e 2, apresentam um espaçamento que permite diferentes retas de separação, mas que também separam corretamente todos os pontos.

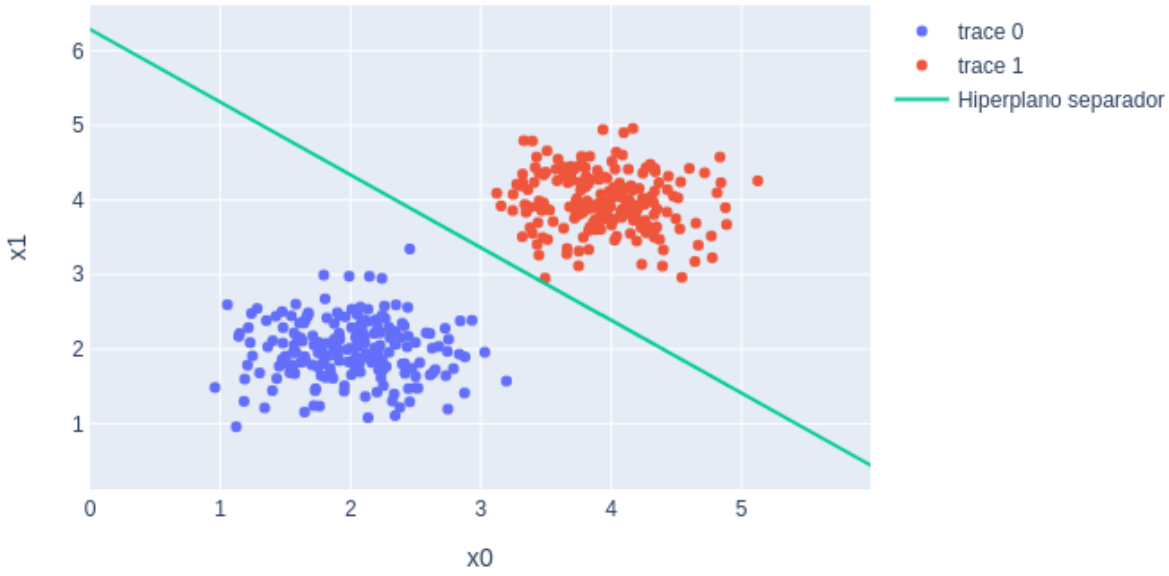

Figura 3: Pontos de entrada e reta de separação.

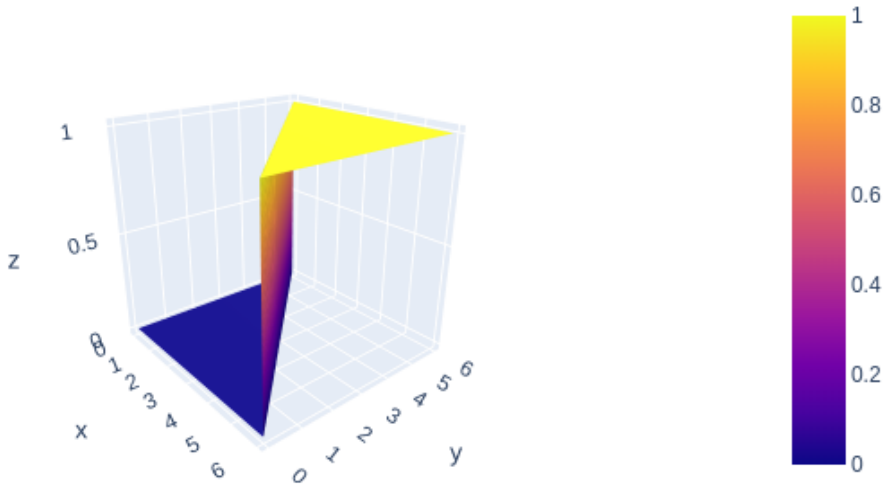

Figura 4: Superfície de separação.

4 Exercício 3

Para esse exercício foi utilizado o conjunto de dados de Câncer de Mama(BreastCancer), disponível no scikit-learn. Esse conjunto de dados é uma versão do conjunto de dados desse mesmo problema encontrado no *mlbench*. Foi utilizada a versão do scikit-learn pela facilidade de importar no Python. Ele possui 569 amostras, 130 menos do que a versão para R, desconsiderando as amostras com dados faltantes no R. No entanto como o fim desse exercício é apenas didático, considerei que não haveria danos causados por esse número menor de amostras.

Para o treinamento do Perceptron foi utilizada a técnica de validação cruzada com 10 *folds*. Os parâmetros de treinamento escolhidos foram:

$$tolerancia = 1e-6$$

$$nEpocas = 10000$$

Para cada *fold* foi calculada a acurácia do modelo treinado. Foi calculado também o *recall*. Decidi calcular o *recall*, pois para o problema de classificação estudado, os resultados Falsos-Negativos, isto é, a classificação obtida pelo modelo é benigno enquanto a classificação real é maligno, são extremamente importantes de serem analisados, já que podem causar sérios danos aos pacientes, caso as respostas do modelo sejam consideradas no diagnóstico.

Fold	Acurácia	*Recall*
1	85.7%	78.4%
2	83.9%	80.0%
3	76.8%	66.7%
4	60.7%	48.7%
5	83.9%	73.6%
6	73.2%	56.3%
7	85.7%	78.9%
8	89.3%	97.1%
9	89.3%	97.1%
10	80.3%	100%

Tabela 1: Acurácias e *recall* para cada *fold*.

A partir dos dados da Tabela 1, foram calculadas as médias e desvios padrões para as acurácias e *recalls*. Esses resultados estão dispostos na Tabela 2.

Parâmetro	Acurácia	*Recall*
Média	80.1%	77.7%
Desvio Padrão	8.6%	17.3%

Tabela 2: Médias e desvios padrões.

Os resultados para esse experimento mostram a capacidade de classificação que pode ser obtida utilizando o Perceptron, mesmo para um problema complexo de dimensão 9.

Esse exercício também instigou uma curiosidade em saber quão bom seria a classificação do algoritmo se mais dados estivessem disponíveis, pois a quantidade usadas para testes é extremamente pequena se compararmos com os casos que ocorrem no mundo, no conjunto de dados utilizados existem apenas 569 amostras enquanto, só no Brasil, é estimado que sejam diagnosticados mais de 66 mil casos em 2020 [https://www.inca.gov.br/controle-do-cancer-de-mama/conceito-e-magnitude].

Além da acurácia, foi possível verificar como a escolha dos dados de treinamento e teste podem impactar no resultado do modelo obtido. No *fold 1* por exemplo, a acurácia do modelo foi 60.7% e o *recall*, ou seja se consideramos que os testes foram realmente diagnósticos, cerca de 52% dos casos seriam Falsos-Negativos, i.e., 52% dos pacientes seriam diagnosticados como benigno, enquanto eles teriam tumores malignos.