Talita Halboth Cunha Fernandes
GRR20165399
talitahalboth@gmail.com

Trabalho Prático 2
Inteliência Artificial 2020/2021

2021-01-11

1 Implementação do Perceptron

A implementação da função de atualização de pesos do perceptron foi bastante simples. Usando a biblioteca Numpy, é possível encontrar o produto escalar de 2 vetores facilmente, assim como realizar multiplicação por um escalar e fazer soma de vetores, que são as operações necessárias nessa função.

2 Classificação

Foram realizados diversos testes com o MLP, para encontrar uma combinação com um alto valor de acurácia. Foram usadas 2, 3 e 4 camadas, cada camada com 10, 100 ou 1000 neurônios. Os melhores resultados são apresentados na tabela 1.

Acurácia	1a camada	2a camada	3 camada	4a camada
0.969	1000	10	10	100
0.969	1000	1000	100	100
0.97	1000	1000	1000	0
0.972	1000	1000	10	100
0.976	1000	1000	100	1000

Tabelle 1: Maiores acurácias encontradas

Usando um modelo com 3 camadas, todas com 1000 neurônios cada (que teve uma acurária de 97% no teste com outros parâmetros default), foram testados diferentes funções de ativação (identity, tahn, logistic e relu) e diferentes solucioinadores (lbfgs, sgd e adam), e os resultados encontrados podem ser encontradas na tabela 2. Podemos ver que todas as instâncias que usaram a função de ativação identity tiveram um desempenho pior que as outras. Isto acontece por esta função simplesmente retorna $f(x) = x$. Exceto pela função de ativação, a utilização de diferentes parâmetros não demonstrou diferenças muitos significativas na acurácia do classificador, levando-se em consideração que pequenas diferenças são esperadas já que o classificador é um algoritmo probabilístico.

Acurácia	Activation	Solver
0.508	identity	sgd
0.836	identity	lbfgs
0.856	identity	adam
0.89	logistic	sgd
0.913	tanh	sgd
0.932	logistic	adam
0.938	tanh	adam
0.945	tanh	lbfgs
0.956	relu	lbfgs
0.958	logistic	lbfgs
0.963	relu	sgd
0.967	relu	adam

Tabelle 2: Acurácia de diferentes Funções de Ativação vs Solucionadores

3 Conclusões

Na base xor, o MLP teve sempre 100% de acurácia, enquanto o perceptron teve apensar 50%. Isso é de se esperar levando em consideração a forma como funciona o perceptron e a distribuição de elementos de cada classe. O perceptron implementado é um método de classificação binária linear. Isto significa que ele é capaz de classificar elementos dado que estes sejam linearmente separáveis. Pensando de forma gráfica, isto significa que é possível desenhar uma reta que divide os elementos em dois grupos, como por exemplo na figura 1, onde a reta separa os elementos x dos elementos • .

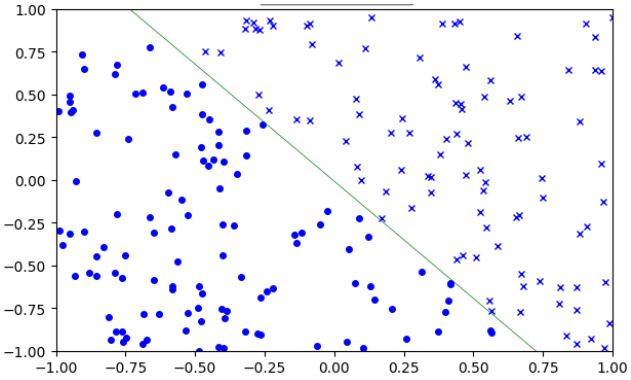

Abbildung 1: Elementos linearmente separáveis

Porém, a base xor não tem esta característica. Por causa da forma como estão aglomerados os dados de cada tipo, é impossível desenhar uma única reta capaz de separar os dados em duas categorias diferentes. O melhor que é possível ser feito é separar os dados de forma a acertar a classificação apensar metade das vezes, como na figura 2. Há várias retas que podem fazer essa mesma separação, de forma que os dados separados contenham 50% dos elementos de cada classe de cada lado.

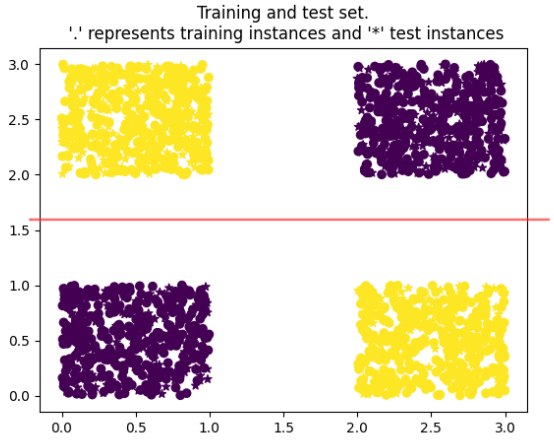

Abbildung 2: Separação dos elementos do conjunto xor

Já no caso da base de dados MNIST, o perceptron conseguiu ter por volta de 85% de acurácia, enquanto o MLP chega a ter até 97% nos testes realizados. A distribuição dos dados na base MNIST, diferentemente dos dados da base XOR, são mais possíveis de serem separadas por uma linha, apesar de não haver nenhuma reta que faça isso de forma perfeita. Por isso o perceptron tem mais acurácia quando trabalhando com essa base. Porém, os dados não estão distribuídos de forma tão homogênea como na base XOR, o que faz com que a acurácia do MPL seja reduzida.