

Universidade da Beira Interior

Departamento de Informática

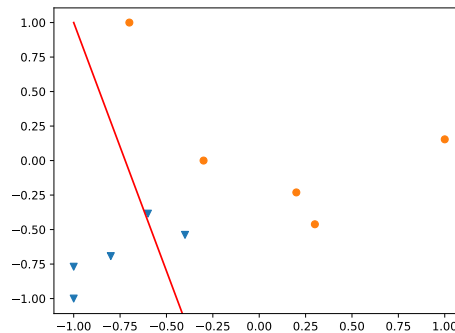
Inteligência Computacional

Ficha prática 1

Luís A. Alexandre

- 1 O objetivo deste exercício é obter uma implementação do neurónio artificial.
A classe deve chamar-se **Neuronio**. Use o gerador de números aleatórios do **Numpy**.
Pretende-se que a classe possua as seguintes características:
 - um construtor `__init__()` que recebe o número de entradas do neurónio e uma semente aleatória para inicializar o gerador de números aleatórios
 - o construtor deve criar o vetor de pesos de dimensão adequada
 - deve ainda inicializar o vetor de pesos com valores aleatórios no intervalo $[-0.1, 0.1]$
 - um método `sigmoid()` que implemente a função de ativação sigmóide
 - um método `calcula()` que devolva a saída do neurónio dado um vetor com as entradas
 - um método `mostra_pesos()` que mostre os pesos do neurónio no ecrã
- 2 Escreva um programa que abra um ficheiro de dados (use a biblioteca **csv**, mas note que o **delimiter** é um espaço), crie um neurónio de dimensão adequada e mostre o resultado da saída para cada ponto do conjunto de dados **exemplo1.dat**.
Se inicializar o gerador de números aleatório usando `seed(53217)` deve obter como saídas os valores do ficheiro **out1.dat**.
- 3 Altere a classe **Neuronio** para que a função de ativação possa ser escolhida aquando da criação do neurónio através da passagem do nome da mesma ao construtor. Deve disponibilizar as funções de ativação vistas na aula teórica. Volte a achar as saídas para os pontos do conjunto de dados **exemplo1.dat** para cada uma das funções. Compare os resultados que obteve com os dos seus colegas.
- 4 Acrescente métodos à classe **Neuronio** que permitam gravar e ler os pesos a partir de um ficheiro.

5. Implemente um método na classe `Neuronio` que permita efetuar a aprendizagem de problemas usando a regra da descida do gradiente em modo estocástico. Acrescente ao construtor da classe `Neuronio` mais um parâmetro que corresponde à taxa de aprendizagem.
6. Usando os dados do ficheiro `exemplo1N.dat`, valide a sua implementação do exercício 5 verificando graficamente a posição da fronteira de decisão do neurónio após cada iteração do algoritmo de aprendizagem implementado. Use o `matplotlib` para o efeito. Exemplo:



7. Considere os seguintes parâmetros: $\eta = 0.05$, inicializar o gerador de números aleatórios a 10 e a função de ativação sigmóide com $\lambda = 1$ usando a descida do gradiente com aprendizagem em modo estocástico. Quantas épocas necessita para obter uma solução (erro de classificação = 0) usando os dados do ficheiro `exemplo1N.dat`?
8. Implemente uma classe que permita efetuar aprendizagem não supervisionada usando a quantificação vetorial.
Não considere a existência de vizinhança no algoritmo: apenas são atualizados os pesos do neurónio que vence para cada ponto de entrada.
9. Use os dados no ficheiro `exemplo2.dat`, para verificar o resultado do seu algoritmo. Faça testes com 2 e 3 grupos. Visualize os resultados com o `matplotlib`.
10. Modifique o exercício 8 para considerar a existência de vizinhança no algoritmo. Veja o impacto que tem, repetindo o exercício 9 com 4 e 5 grupos, usando valores diferentes para a vizinhança.
11. Use os dados no ficheiro `exemplo2.dat`. Fixe o número de grupos em 2 e o número de épocas em 6. Faça variar o parâmetro η no algoritmo de forma a que o seu valor inicial seja sempre 0.9 mas a taxa de atualização $X > 0$ em cada época t varie de acordo com:

$$\eta(t) = \eta(t - 1) - X$$

Construa uma tabela com o valor do erro de quantização ao longo das 6 épocas versus os valores que usou para X . Compare com os resultados obtidos pelos seus colegas.