

TP555 - AI/ML

Lista de Exercícios #11

Redes Neurais Artificiais (Parte 1)

1. Usando-se o modelo do neurônio de McCulloch e Pitts, qual seria o valor do limiar de ativação, θ , para classificar a função booleana abaixo (dada pela tabela abaixo)? Desenhe a função de ativação e o neurônio, indicando quais entradas são inibitórias, caso haja alguma.

(**Dica:** entradas inibitórias são entradas que tem seus valores '*negados*'. Vocês podem precisar ter uma ou mais entradas inibitórias para encontrar o valor de θ).

x1	x2	x3	y
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0

2. Baseado no que você aprendeu até aqui sobre o modelo do neurônio de McCulloch e Pitts e do modelo perceptron. Seria possível classificar a função booleana XOR com algum desses dois modelos? Explique os motivos pelos quais pode-se ou não realizar tal classificação.
3. Por que geralmente é preferível usar um **classificador de regressão logística** em vez de um **perceptron**? Quais modificações você deve aplicar à um **perceptron** para torná-lo equivalente a um **classificador de regressão logística**?
4. Implemente um **perceptron** que classifique com 100% de precisão os dados das funções lógicas dadas a seguir utilizando a **regra de aprendizado do perceptron**. Em seguida, para cada uma das funções lógicas, plote uma figura mostrando:
 - a. a **fronteira de decisão** que separa as 2 classes.
 - b. o número de épocas versus o erro quadrático médio (MSE) por época.

(**Dica:** encontre o valor ótimo do passo de aprendizagem, α)

(**Dica:** use `np.random.permutation` para embaralhar os dados a cada nova época)

(**Dica:** execute a regra de aprendizagem por um número predefinido de épocas, e.g, 2000, e sempre armazene o **vetor de pesos sinápticos**, w , que resulte no menor **erro quadrático médio** (MSE))

a)			b)		
AND			OR		
x1	x2	y	x1	x2	y
0	0	0	0	0	0
0	1	0	0	1	1
1	0	0	1	0	1
1	1	1	1	1	1

5. Neste exercício você irá comparar a performance de classificação de um **perceptron** com a de um **regressor logístico**. Use o código abaixo para gerar os dados pertencentes a duas classes

```
# Number of examples.
```

```
N = 1000
```

```
centers = [[-0.5, 0], [0, 1.5]]
```

```
X, y = make_blobs(n_samples=N, centers=centers, random_state=42)
```

Em seguida, faça o seguinte

- Plote os dados do conjunto de treinamento em relação às classes a que pertencem. Ou seja, defina marcadores diferentes para identificar cada um das classes na figura. Por exemplo, use círculos para denotar exemplos que pertencem à classe 0 e quadrados para denotar exemplos que pertencem à classe 1.
- Instancie um objeto da classe **Perceptron**, use a linha abaixo para realizar a instanciação.

```
per = Perceptron(random_state=42)
```

Em seguida treine o modelo, faça a predição com X e calcule a precisão deste modelo como mostrado abaixo.

```
# Calculate and return the accuracy on the test data
```

```
accuracy = accuracy_score(y, y_pred)
```

```
print('accuracy: ',accuracy)
```

- Plote a matriz de confusão e a figura com a fronteira de decisão para a classificação com o **Perceptron**.
- Instancie um objeto da classe **LogisticRegression**, use a linha abaixo para realizar a instanciação.

```
per = LogisticRegression(solver='lbfgs', random_state=42)
```

Em seguida treine o modelo, faça a predição com X e calcule a precisão deste modelo como mostrado abaixo.

```
# Calculate and return the accuracy on the test data
accuracy = accuracy_score(y, y_pred)

print('accuracy: ',accuracy)
```

- E. Plote a matriz de confusão e a figura com a fronteira de decisão para a classificação com o ***regressor logístico***.
- F. Baseado nos resultados obtidos, qual dos 2 classificadores apresenta melhor performance? Você conseguiria explicar porque ele apresenta melhor performance?