

Deep Learning - Introdução a Modelagem Matemática

10.1 Exercício: Rede MLP - Multilayer Perceptron

Considere o conjunto de dados abaixo constituído pelos dados de entrada nas três primeiras colunas (x_1 , x_2 , x_3) da matriz seguido pelo respectivo rótulo (0, 1 ou 2).

#	x_1	x_2	x_3	$classe$
dataset =	[1.02153588,	-1.04584554,	-0.96943922,	0],
	[1.07472359,	-0.81372418,	-0.50227571,	0],
	[1.03845087,	-0.85529440,	-1.07551718,	0],
	[1.08671323,	-0.39041877,	-1.06912290,	0],
	[1.81386050,	-1.03705351,	-1.01491796,	0],
	[1.60285424,	-0.53666876,	-0.68868644,	0],
	[1.67743150,	-0.68302534,	-0.58474262,	0],
	[1.34016832,	-0.33036141,	-0.45972834,	0],
	[1.15715279,	-0.57687815,	-0.86905314,	0],
	[1.19312552,	-0.74858765,	-0.26682942,	0],
	[1.32535135,	-0.51619190,	-0.50434856,	0],
	[1.17080360,	-0.29582611,	-0.31267014,	0],
	[-0.60098243,	1.13155099,	-0.77290891,	1],
	[-0.78660704,	1.22050116,	-1.05339234,	1],
	[-0.42366120,	1.29704384,	-0.94234171,	1],
	[-0.66885149,	1.43052978,	-1.00206341,	1],
	[-0.46490586,	1.76682434,	-0.74631286,	1],
	[-1.02489359,	1.35547338,	-0.38458331,	1],
	[-0.96717853,	1.46557232,	-0.68402367,	1],
	[-0.57114584,	1.15404176,	-0.87468506,	1],
	[-0.94714779,	1.13832305,	-0.58694270,	1],
	[-0.40102286,	1.46159431,	-0.69792237,	1],
	[-0.26437944,	1.34796154,	-0.73774277,	1],
	[-0.27293769,	1.59309487,	-1.04274151,	1],
	[-0.30283821,	-0.73600306,	1.44039980,	2],
	[-0.58290259,	-0.25539981,	1.50781368,	2],

```

[-1.08018241, -0.25738737, 1.09149484, 2],
[-0.76039084, -0.97361097, 1.28860632, 2],
[-1.04975329, -0.77085457, 1.78097885, 2],
[-0.40436016, -0.52396243, 1.51026685, 2],
[-0.83580163, -0.84298958, 1.05583722, 2],
[-0.84797325, -0.47850486, 1.55482311, 2],
[-0.27698582, -0.67465935, 1.74025219, 2],
[-0.94195261, -0.72946186, 1.79217650, 2],
[-0.48707297, -0.86887812, 1.20341262, 2],
[-0.82242448, -0.75027166, 1.49045897, 2]]

```

Implemente o treinamento supervisionado de uma rede MLP considerando todo o conjunto de dados acima como conjunto de treinamento.

Considerações:

- O modelo deve possuir uma camada com quatro neurônios recebendo os dados de entrada (x1, x2, x3 e o bias) seguida pela camada de saída com três neurônios (um neurônio para cada classe) recebendo as saídas da camada escondida;
- Utilize a técnica "One-Hot-Encoding" para converter os rótulos de saída
- Repita o laço de treinamento 20 vezes
- Utilize

```

LEARNING_RATE = 0.01
np.random.seed(7)

```

Entrada: Nova instância (x1, x2, x3) para classificar.

Saída: A classe da instância de entrada. Por exemplo, a saída dos neurônios do primeiro caso teste é:

```
[0.46114314 0.4329263 0.43107695]
```

porém a classe é dada pela posição do maior valor, ou seja:

```
output_layer_y.argmax()
```

resultando na classe:

```
>> 0
```