



UFC – UNIVERSIDADE FEDERAL DO CEARÁ
CAMPUS DE SOBRAL
CURSO DE ENGENHARIA COMPUTAÇÃO
INTELIGÊNCIA COMPUTACIONAL
PROFESSOR: JARBAS JOACI

TRABALHO 2

FRANCISCO GILSON PEREIRA ALMEIDA FILHO – 401066

SUMÁRIO

1	Questão 1	3
2	Questão 2	5
	REFERÊNCIAS.....	7

1. Questão 1

Questão 1: Implemente um neurônio Perceptron com valores de pesos inicialmente aleatórios para resolver o problema do AND com duas variáveis. O programa deverá plotar os pontos da tabela-verdade e a reta de separação obtida.

Alguns pontos:

- Na matriz de entrada a primeira coluna são os valores do 'bias'.
- Usei a função `rand()` para fornecer os valores aleatórios dos pesos.
- Usei a equação " $w(i) = w(i) + \eta * \text{erro} * x(i)$ " fornecida no slide para o treinamento.
- Usei 0.5 na taxa de aprendizado, já que $0 < \eta < 1$.
- Usei a equação da reta fornecida no slide " $x_2 = -(w(2)/w(3)) * x_1 + w(1)/w(3)$ " para o plot.
 - $W(1)$ é o peso do 'bias' (θ), $W(2)$ e $W(3)$ são os pesos da entrada do AND.
- Coloquei um delay de um segundo para cada operação, então, como são quarenta operações o programa demora quarenta segundos para executar por completo, e ao final faz o plote, os valores podem ser acompanhados no console, a cada segundo imprime as informações referentes a um treinamento (Imagem 1).

Fonte: Autor.

```
"-----"

"Pesos:"

1.2113249
0.7560439
0.5002211

"Entrada:"

-1.    1.    1.

"Alvo:"

1.

"net:"

1.
```

Imagem 1: Impressão no console das informações do treinamento.

Resultados e discussões sobre a questão 1:

O treinamento roda dez vezes para os valores da porta AND de duas variáveis, um total de quarenta operações. Fiz a impressão no console dos principais valores a serem observados: (pesos, entrada, alvo(saída) e net).

No final foi feito o plot dos pontos da tabela verdade de duas variáveis e da reta de separação.

Na imagem abaixo temos um exemplo com os pesos: $W1= 1.2113249$ $W2= 0.7560439$ $W3= 0.5002211$.

Fonte da imagem: Autor.

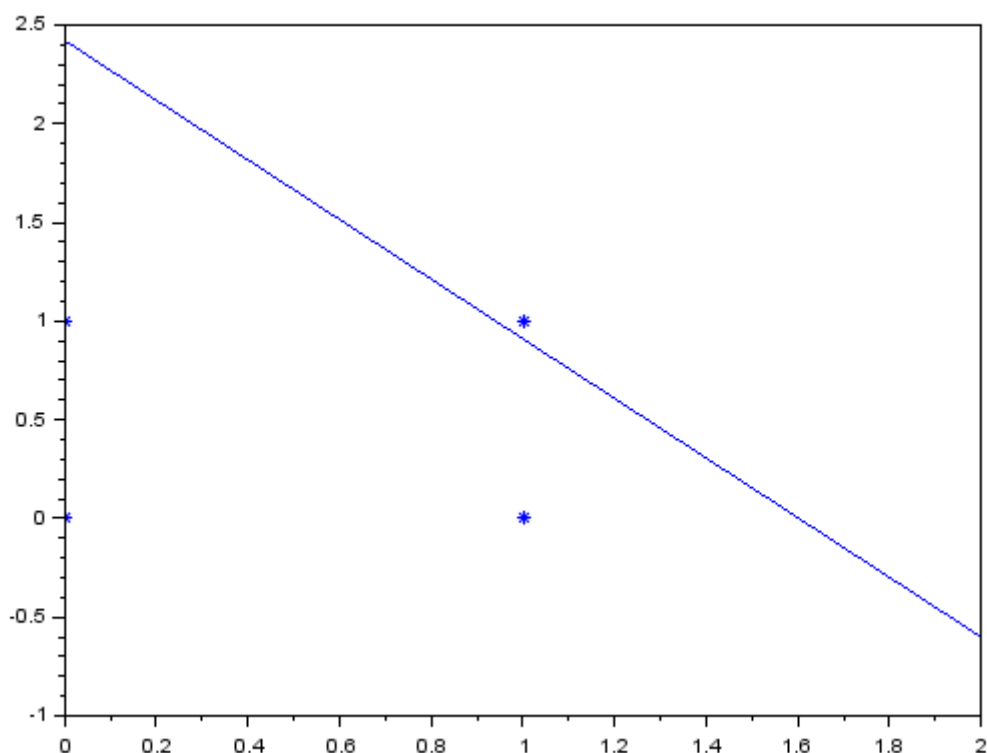


Imagem 2: Plote dos pontos da tabela verdade e da reta de separação para os pesos mencionados acima.

Percebe-se que, a reta sempre vai ser diferente devido a “aleatoriedade” dos valores dos pesos, mas sempre será abaixo do ponto (1,1).

2. Questão 2

Questão 2: Implemente uma rede RBF para criar uma curva que se ajuste ao conjunto de dados do aerogerador. Plote as curvas obtidas (e mostre os valores de R2 correspondentes) para diferentes quantidades de neurônios ocultos.

Resultados e discussões sobre a questão 1:

O programa a cada segundo plota uma curva e imprime no console o valor de R2 com base no valor da quantidade de neurônios ocultos, que está indo de dez a vinte neurônios.

Usei um loop para realizar essa ação, e a cada passagem a quantidade de neurônios ocultos aumenta + 2, então são seis plots com valores de 10,12,14,16,18 e 20 neurônios ocultos.

No gráfico, a cada um segundo, uma nova curva aparece com base no valor de qN (Imagem 3).

Fonte: Autor.

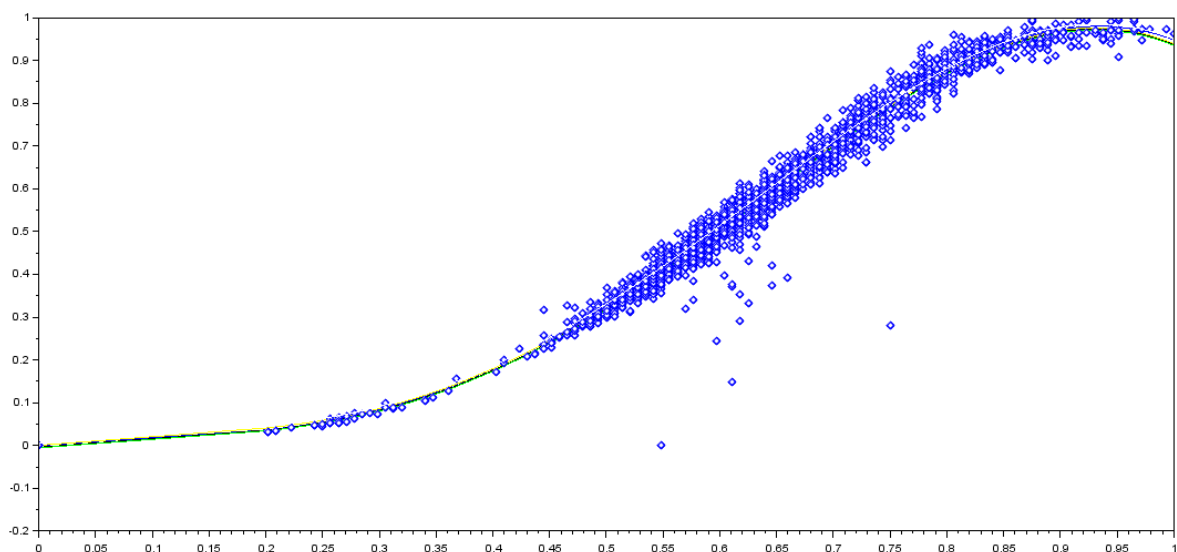


Imagem 3: Plote das 6 curvas com base na quantidade de neurônios.

A mesma coisa para o valor de R2 no console (Imagem 4).

Fonte: Autor.

"Quantidade de neuronios:"	"Quantidade de neuronios:"
18.	20.
"Valor de R2"	"Valor de R2"
1.1492234	0.7188219

Imagem 3: Exemplos de valores de R2 com base nas quantidades de neurônios 18 e 20.

Configurei usando alguns IFS para que cada curva tenha sua cor específica para facilitar a visualização. (Tabela 1).

Fonte: Autor.

Quantidade de neurônios	Cor da curva
10	Vermelho
12	Azul
14	Amarelo
16	Preto
18	Verde
20	Branco

Tabela 1: Relação quantidade de neurônios – cor da curva.

Referências

Ajuda do Scilab. Disponível em: https://help.scilab.org/docs/5.5.0/pt_BR/index.html. Acesso em: 25 de ago.2020.