



## **Redes Neurais Artificiais**

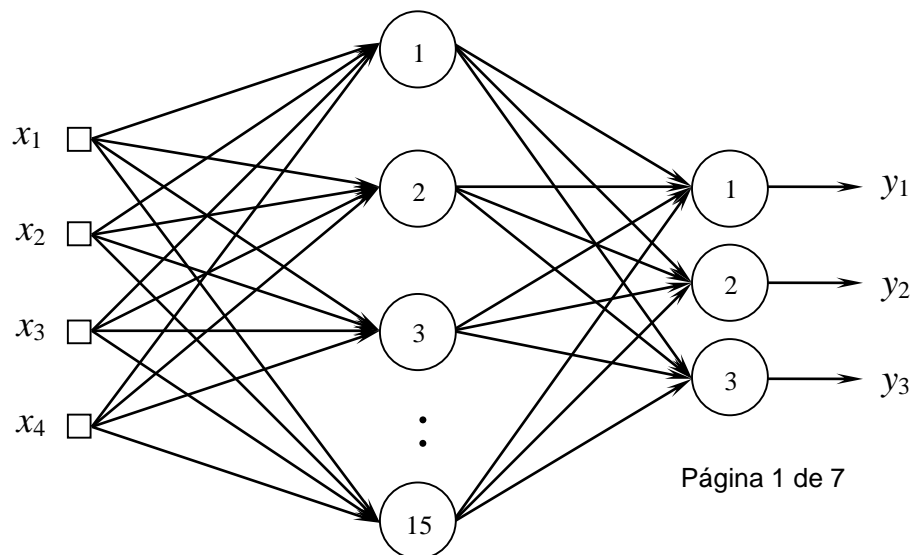
(Prof. Ivan Nunes da Silva)

LUIZ FELIPE MACHADO VOTTO

### **EPC-5**

No processamento de bebidas, a aplicação de um determinado conservante é efetuada em função da combinação de 04 variáveis reais, definidas por  $x_1$  (teor de água),  $x_2$  (grau de acidez),  $x_3$  (temperatura) e  $x_4$  (tensão superficial). Sabe-se que existem apenas três tipos de conservantes que podem ser aplicados, os quais são categorizados por tipo A, B e C. A partir destas variáveis, realizam-se ensaios em laboratório para especificar que tipo de conservante deve ser aplicado em determinada bebida.

Por intermédio de 148 desses ensaios experimentais, a equipe de engenheiros e cientistas resolveu aplicar uma rede perceptron multicamadas como classificadora de padrões, a fim de que esta identifique qual conservante será aplicado em determinado lote de bebida. Por questões operacionais da própria linha de produção, utilizar-se-á aqui uma rede perceptron com três saídas, conforme apresentado na figura abaixo.





A padronização para a saída, representando o conservante a ser aplicado, ficou definida da seguinte forma:

Tipo de Conservante	$y_1$	$y_2$	$y_3$
Tipo A	1	0	0
Tipo B	0	1	0
Tipo C	0	0	1

Utilizando os dados de treinamento apresentados no Anexo, execute o treinamento de uma rede perceptron multicamadas (04 entradas, 15 neurônios na camada escondida e 03 saídas) que possa classificar, em função apenas dos valores medidos de  $x_1$ ,  $x_2$ ,  $x_3$  e  $x_4$  (já normalizados), qual o tipo de conservante que deverá ser aplicado em determinada bebida. Para tanto, faça as seguintes atividades:

1. Execute cinco treinamentos da rede perceptron multicamadas ilustrada na Figura 1, por meio do algoritmo de aprendizagem **backpropagation**, inicializando-se todas as matrizes de pesos com valores aleatórios entre 0 e 1. Utilize a função de ativação logística para todos os neurônios, taxa de aprendizado  $\eta = 0.1$  e precisão  $\varepsilon = 10^{-6}$ . Meça também o tempo de processamento envolvido com cada um desses treinamentos.
2. Dado que o problema se configura como um típico processo de classificação de padrões, implemente a rotina que faz o pós-processamento das saídas fornecidas pela rede (números reais) para números inteiros. Utilize o critério do arredondamento simétrico, isto é:

$$y_i^{\text{pós}} = \begin{cases} 1, & \text{se } y_i \geq 0.5 \\ 0, & \text{se } y_i < 0.5 \end{cases}, \text{ utilizado apenas no pós-processamento do conjunto de teste.}$$



Universidade de São Paulo  
Escola de Engenharia de São Carlos  
Departamento de Engenharia Elétrica e de Computação



- 
3. Para cada um dos cinco treinamentos, faça então a validação aplicando o conjunto de teste fornecido na tabela abaixo. Forneça a taxa de acerto (%) entre os valores desejados e os valores fornecidos pela rede (após o pós-processamento) em relação a todas as amostras de teste.

**Nas páginas seguintes, estão a saída de um programa Python que usa treinou e avaliou as cinco redes. A primeira coluna diz respeito ao número da amostra, a segunda dá a resposta esperada, a terceira, a resposta recebida pós-processada seguida pela quarta coluna que contém as saídas sem pós-processamento. Ao fim de cada tabela, há a taxa de acertos da rede correspondente.**



NET 0

0	:	[0, 0, 1]	[0, 0, 1]	['0.00', '0.01', '1.00']
1	:	[1, 0, 0]	[1, 0, 0]	['1.00', '0.00', '0.00']
2	:	[0, 0, 1]	[0, 0, 1]	['0.00', '0.01', '1.00']
3	:	[0, 1, 0]	[0, 1, 0]	['0.01', '0.91', '0.00']
4	:	[0, 0, 1]	[0, 0, 1]	['0.00', '0.02', '0.99']
5	:	[1, 0, 0]	[1, 0, 0]	['0.99', '0.00', '0.00']
6	:	[0, 1, 0]	[0, 1, 0]	['0.00', '0.99', '0.00']
7	:	[0, 1, 0]	[0, 1, 0]	['0.00', '0.95', '0.03']
8	:	[1, 0, 0]	[1, 0, 0]	['0.99', '0.00', '0.00']
9	:	[1, 0, 0]	[1, 0, 0]	['1.00', '0.00', '0.00']
10	:	[0, 1, 0]	[0, 1, 0]	['0.00', '0.99', '0.00']
11	:	[1, 0, 0]	[1, 0, 0]	['1.00', '0.00', '0.00']
12	:	[0, 0, 1]	[0, 0, 1]	['0.00', '0.00', '1.00']
13	:	[0, 0, 1]	[0, 0, 1]	['0.00', '0.00', '1.00']
14	:	[0, 0, 1]	[0, 0, 1]	['0.00', '0.00', '1.00']
15	:	[1, 0, 0]	[1, 0, 0]	['0.94', '0.05', '0.00']
16	:	[0, 0, 1]	[0, 1, 0]	['0.00', '0.74', '0.15']
17	:	[0, 1, 0]	[0, 1, 0]	['0.00', '0.99', '0.00']

0.9444444444444444

NET 1

0	:	[0, 0, 1]	[0, 0, 1]	['0.00', '0.02', '0.99']
1	:	[1, 0, 0]	[1, 0, 0]	['1.00', '0.00', '0.00']
2	:	[0, 0, 1]	[0, 0, 1]	['0.00', '0.01', '0.99']
3	:	[0, 1, 0]	[0, 1, 0]	['0.02', '0.91', '0.00']
4	:	[0, 0, 1]	[0, 0, 1]	['0.00', '0.02', '0.99']
5	:	[1, 0, 0]	[1, 0, 0]	['0.99', '0.00', '0.00']
6	:	[0, 1, 0]	[0, 1, 0]	['0.00', '0.99', '0.00']
7	:	[0, 1, 0]	[0, 1, 0]	['0.00', '0.95', '0.05']
8	:	[1, 0, 0]	[1, 0, 0]	['1.00', '0.00', '0.00']
9	:	[1, 0, 0]	[1, 0, 0]	['1.00', '0.00', '0.00']
10	:	[0, 1, 0]	[0, 1, 0]	['0.00', '0.99', '0.01']
11	:	[1, 0, 0]	[1, 0, 0]	['1.00', '0.00', '0.00']
12	:	[0, 0, 1]	[0, 0, 1]	['0.00', '0.00', '1.00']
13	:	[0, 0, 1]	[0, 0, 1]	['0.00', '0.01', '1.00']
14	:	[0, 0, 1]	[0, 0, 1]	['0.00', '0.00', '1.00']
15	:	[1, 0, 0]	[1, 0, 0]	['0.96', '0.05', '0.00']
16	:	[0, 0, 1]	[0, 1, 0]	['0.00', '0.86', '0.05']
17	:	[0, 1, 0]	[0, 1, 0]	['0.00', '0.99', '0.00']

0.9444444444444448



NET 2

0 : [0, 0, 1] [0, 0, 1] ['0.00', '0.02', '0.99']  
1 : [1, 0, 0] [1, 0, 0] ['1.00', '0.00', '0.00']  
2 : [0, 0, 1] [0, 0, 1] ['0.00', '0.01', '0.99']  
3 : [0, 1, 0] [0, 1, 0] ['0.02', '0.92', '0.00']  
4 : [0, 0, 1] [0, 0, 1] ['0.00', '0.02', '0.98']  
5 : [1, 0, 0] [1, 0, 0] ['0.98', '0.00', '0.00']  
6 : [0, 1, 0] [0, 1, 0] ['0.00', '0.99', '0.00']  
7 : [0, 1, 0] [0, 1, 0] ['0.00', '0.93', '0.05']  
8 : [1, 0, 0] [1, 0, 0] ['0.99', '0.00', '0.00']  
9 : [1, 0, 0] [1, 0, 0] ['1.00', '0.00', '0.00']  
10 : [0, 1, 0] [0, 1, 0] ['0.00', '0.99', '0.01']  
11 : [1, 0, 0] [1, 0, 0] ['1.00', '0.00', '0.00']  
12 : [0, 0, 1] [0, 0, 1] ['0.00', '0.00', '1.00']  
13 : [0, 0, 1] [0, 0, 1] ['0.00', '0.00', '1.00']  
14 : [0, 0, 1] [0, 0, 1] ['0.00', '0.00', '1.00']  
15 : [1, 0, 0] [1, 0, 0] ['0.94', '0.04', '0.00']  
16 : [0, 0, 1] [0, 1, 0] ['0.00', '0.81', '0.06']  
17 : [0, 1, 0] [0, 1, 0] ['0.00', '0.99', '0.00']

0.9444444444444448

NET 3

0 : [0, 0, 1] [0, 0, 1] ['0.00', '0.02', '0.98']  
1 : [1, 0, 0] [1, 0, 0] ['1.00', '0.00', '0.00']  
2 : [0, 0, 1] [0, 0, 1] ['0.00', '0.01', '0.99']  
3 : [0, 1, 0] [0, 1, 0] ['0.04', '0.98', '0.00']  
4 : [0, 0, 1] [0, 0, 1] ['0.00', '0.03', '0.98']  
5 : [1, 0, 0] [1, 0, 0] ['0.99', '0.00', '0.00']  
6 : [0, 1, 0] [0, 1, 0] ['0.00', '0.99', '0.00']  
7 : [0, 1, 0] [0, 1, 0] ['0.00', '0.95', '0.04']  
8 : [1, 0, 0] [1, 0, 0] ['0.99', '0.00', '0.00']  
9 : [1, 0, 0] [1, 0, 0] ['1.00', '0.00', '0.00']  
10 : [0, 1, 0] [0, 1, 0] ['0.00', '0.99', '0.00']  
11 : [1, 0, 0] [1, 0, 0] ['1.00', '0.00', '0.00']  
12 : [0, 0, 1] [0, 0, 1] ['0.00', '0.00', '1.00']  
13 : [0, 0, 1] [0, 0, 1] ['0.00', '0.00', '1.00']  
14 : [0, 0, 1] [0, 0, 1] ['0.00', '0.00', '1.00']  
15 : [1, 0, 0] [1, 0, 0] ['0.96', '0.01', '0.00']  
16 : [0, 0, 1] [0, 1, 0] ['0.00', '0.86', '0.08']  
17 : [0, 1, 0] [0, 1, 0] ['0.00', '1.00', '0.00']

0.9444444444444448



\_\_\_\_\_ NET 4 \_\_\_\_\_

0 : [0, 0, 1] [0, 0, 1] ['0.00', '0.02', '0.99']  
1 : [1, 0, 0] [1, 0, 0] ['1.00', '0.00', '0.00']  
2 : [0, 0, 1] [0, 0, 1] ['0.00', '0.01', '0.99']  
3 : [0, 1, 0] [0, 1, 0] ['0.02', '0.97', '0.00']  
4 : [0, 0, 1] [0, 0, 1] ['0.00', '0.01', '0.98']  
5 : [1, 0, 0] [1, 0, 0] ['0.99', '0.01', '0.00']  
6 : [0, 1, 0] [0, 1, 0] ['0.00', '0.99', '0.00']  
7 : [0, 1, 0] [0, 1, 0] ['0.00', '0.93', '0.05']  
8 : [1, 0, 0] [1, 0, 0] ['0.99', '0.01', '0.00']  
9 : [1, 0, 0] [1, 0, 0] ['1.00', '0.00', '0.00']  
10 : [0, 1, 0] [0, 1, 0] ['0.00', '1.00', '0.00']  
11 : [1, 0, 0] [1, 0, 0] ['1.00', '0.00', '0.00']  
12 : [0, 0, 1] [0, 0, 1] ['0.00', '0.00', '1.00']  
13 : [0, 0, 1] [0, 0, 1] ['0.00', '0.00', '1.00']  
14 : [0, 0, 1] [0, 0, 1] ['0.00', '0.00', '1.00']  
15 : [1, 0, 0] [1, 0, 0] ['0.97', '0.02', '0.00']  
16 : [0, 0, 1] [0, 1, 0] ['0.00', '0.90', '0.10']  
17 : [0, 1, 0] [0, 1, 0] ['0.00', '0.99', '0.00']

0.94444444444444448



4. Explique qual foi o motivo de se realizar cinco treinamentos para uma mesma configuração topológica de rede perceptron multicamadas. **Pois cada rede se inicializa a uma posição aleatória no espaço de busca. O método de descida gradiente usado no back-propagation pode resultar em uma rede estabilizando num mínimo local do espaço de busca, por isso o fato de inicializar cada rede em locais aleatórios é importante.**
5. Para o melhor dos cinco treinamentos realizados acima, trace o respectivo gráfico dos valores de erro quadrático médio (EQM) em função de cada época de treinamento.

