

Disciplina ACH 2016 – Exercícios (2020).

Esses exercícios foram adaptados das seguintes referências:

- Haykin, S. (1999). Neural Networks: A Comprehensive Foundation. Prentice Hall.
- Fausett, V. F. (1994) Fundamentals of Neural Networks: Architectures, Algorithms, and Applications. Prentice Hall
- Silva, L.A., Peres, S. M. Boscarioli, C. (2016) Introdução à Mineração de Dados: Com aplicações em R. Elsevier.
- Costa, J. A. F. (1999) Classificação automática e análise de dados por redes neurais auto-organizáveis. Tese de Doutorado. Unicamp.
- Han, J., Kamber, M., Pei, J. (2011) Data Mining: Concepts and Techniques. Morgan Kaufmann Publishers

A adaptação foi realizada de forma que o conteúdo ministrado na disciplina fosse suficiente para a resolução de cada exercício.

1- **Perceptron Simples (Fausett, 1994)** Encontre os pesos requeridos para executar a classificação dos seguintes vetores de dados, considerando a regra de aprendizado do Perceptron:

- 1, 1, 1, 1
- -1, 1, -1, -1

que pertencem à classe e, portanto, possuem valor alvo = 1, e

- 1, 1, 1, -1
- 1, -1, -1, 1

que não pertencem à classe, portanto possuem valor alvo = -1.

Use a taxa de aprendizado = 1 e pesos inicializados com 0. Use cada vetor como treinamento e também como teste de sua rede.

2- **Multilayer Perceptron (Fausett, 1994)** Encontre os pesos novos quando a rede ilustrada na figura abaixo recebe:

- a entrada (0, 1) e a saída esperada é 1. Use a taxa de aprendizado = 0,25, e a função de ativação sigmoide binária.
- a entrada (-1, 1) e a saída esperada é 1. Use a taxa de aprendizado = 0,25 e a função de ativação sigmoide bipolar.
- a entrada (0, 1) e a saída esperada é 0,8. Use a taxa de aprendizado = 0,25, e a função de ativação sigmoide binária.
- a entrada (-1, 1) e a saída esperada é 0,8. Use a taxa de aprendizado = 0,25 e a função de ativação sigmoide bipolar.

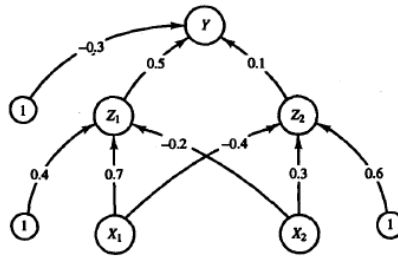


Figure 6.12 Neural network for Exercises 6.1-6.5.

- 3- **Multilayer Perceptron (Fausett, 1994)** Repita o exercício 2 usando um parâmetro de inclinação da função sigmoide em 3,0. Isto aumenta ou diminui a quantidade de aprendizado (tamanho das mudanças nos pesos)?
- 4- Uma rede neural está sendo treinada nos dados do problema XOR. A arquitetura e os valores dos pesos e biases são mostrados abaixo:

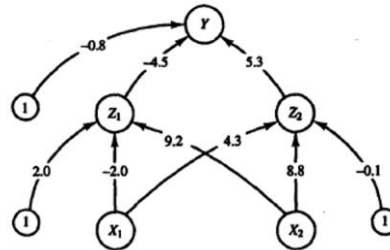


Figure 6.13 Neural network for Exercise 6.6.

- Usando uma sigmoide binária, compute as ativações para cada uma das unidades quando o vetor de entrada (0, 1) é apresentado. Encontre o fator delta (informação do erro δ) das unidades de saída e das unidades escondidas. Use uma taxa de aprendizado = 0,25, compute as correções de pesos. Encontre os novos pesos e biases.
 - Repita para o vetor de entrada (1,0).
 - Analise as diferenças entre as mudanças nas conexões para as unidades de entrada e para as unidades escondidas em ambos os casos anteriores.
- 5- **Multilayer Perceptron (Fausett, 1994)** Teste a sua implementação para o MLP, assumindo a existência de unidades de biases. Use uma função de ativação sigmoide bipolar. Para cada caso de teste, imprima os pesos iniciais, pesos finais, taxa de aprendizado, número de épocas e resposta da rede para cada vetor de entrada no fim do treinamento. Os dados de treinamento são referentes à função XOR bipolar (entradas e saídas bipolares):

$1, -1 \rightarrow 1$
 $-1, 1 \rightarrow 1$
 $1, 1 \rightarrow -1$
 $-1, -1 \rightarrow -1$

Use pesos iniciais aleatoriamente distribuídos em $[-0,5;0,5]$ e a taxa de aprendizado igual a:

- i) 0,05
- ii) 0,25
- iii) 0,5.

Para cada taxa de aprendizado execute 1.000, 10.000 e 25.000 épocas de treinamento (usando os mesmos pesos iniciais em cada caso). Use duas unidades de entrada, duas unidades escondidas e uma unidade de saída.

- 6- **Multilayer Perceptron (Fausett, 1994)** Teste a sua implementação para o MLP considerando uma camada escondida. Use bias nas unidades escondidas e nas unidades de saída, 10 neurônios de entrada, 4 neurônios escondidos; 2 neurônios de saída e função de ativação sigmoide bipolar. Os vetores de entrada para usa rede são:

```

0 1 2 3 4 5 6 7 8 9 → -1 -1
9 8 7 6 5 4 3 2 1 0 → 1 1
0 9 1 8 2 7 3 6 4 5 → -1 1
4 5 6 3 2 7 1 8 0 9 → 1 -1
3 8 2 7 1 6 0 5 9 4 → 1 1
1 6 0 7 4 8 3 9 2 5 → 1 -1
2 1 3 0 4 9 5 8 6 7 → -1 1
9 4 0 5 1 6 2 7 3 8 → -1 -1

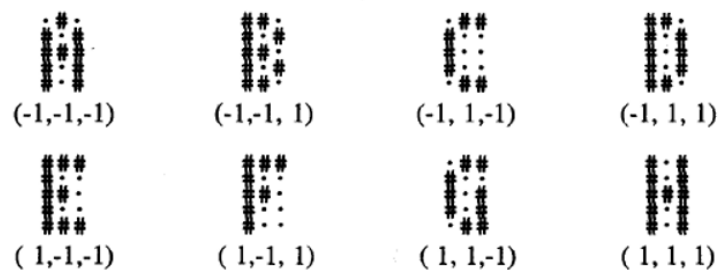
```

Inicialize os pesos aleatoriamente em $[-0,5; 0,5]$ e use taxa de aprendizado em:

- i) 0,05
- ii) 0,5.

Para cada taxa de aprendizdos, execute 5.000 e 50.000 épocas de treinamento (usando o mesmo conjunto de pesos inicial em todos os casos).

- 7- **Multilayer Perceptron (Fausett, 1994)** Teste a sua implementação para o MLP para o problema determinado pela figura abaixo:



Experimente com diferentes quantidades de neurônios escondidos (não mais do que 15), diferentes taxas de aprendizado e diferentes inicializações de pesos.

- 8- **Multilayer Perceptron (Fausett, 1994)** Teste a sua implementação para o MLP para o problema determinado de aprender a função abaixo:

$$y = f(x_1, x_2) = \text{seno}(2\pi x_1) * \text{seno}(2\pi x_2)$$

para $0 \leq x_1 \leq 1$, $0 \leq x_2 \leq 1$. O número de neurônios escondidos deve variar como parte do experimento.

- Tente espaçar os vetores de treinamento igualmente ($i/5, j/5$) para $i = 0, \dots, 5$, $j = 0, \dots, 5$. Misture (scramble) a ordem de apresentação dos vetores de treinamento, e compute o valor esperado

para cada vetor. Teste a rede nos pontos $(i/10, j/10)$ para $i = 0, \dots, 10, j = 0, \dots, 10$. Mostre os resultados. A resposta é melhor depois 10.000 épocas do que depois de 1.000 épocas?

* Este exemplo é, talvez surpreendentemente, difícil segundo a autora.

- Tente usar mais vetores de treinamento para os quais a resposta esperada é diferente de zero, e poucos, se houver, vetores para os quais a resposta esperada é zero.
- Tente usar vetores de treinamento gerados aleatoriamente.

9- **Multilayer Perceptron (Fausett, 1994)** Teste a sua implementação para o MLP para o problema de compressão de dados. Use representação bipolar para os vetores. As respostas esperadas são os próprios vetores de entrada. Você usará 63 unidades de entrada e 63 unidades de saída. Use uma tolerância de 0,2 para considerar a resposta do neurônio de saída correta, ou seja, saídas de 0,8 a 1 são consideradas 1; saídas de -0,8 a -1 são consideradas -1. É sugerido usar 24 neurônios na camada escondida.

* Backpropagation pode ser usado para fazer compressão de dados por meio do treinamento de uma rede para atuar como uma rede autoassociativa (o vetor de treinamento e o vetor de saída resposta esperada – são iguais), com menos neurônios da camada escondida do que nas camadas de entrada e saída.

Esses são os vetores (letras) que devem ser compactados.

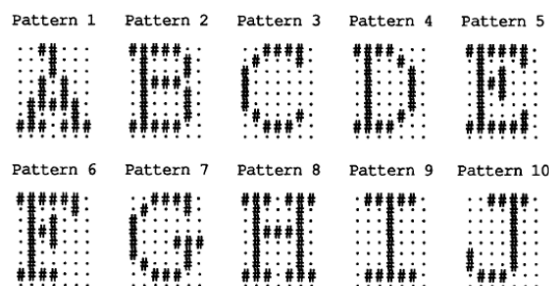
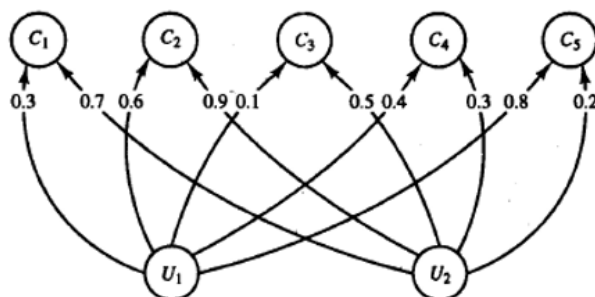


Figure 6.6 Total patterns for Example 6.5.

10- **Kohonen (Fausett, 1994)** Dado um mapa auto-organizável de Kohonen com pesos como mostrado na figura abaixo, use o quadrado da distância euclidiana para encontrar a unidade C_j que é a mais próximo do vetor de entrada $(0,5; 0,2)$. Então, usando uma taxa de aprendizado = 0,2, encontra os novos pesos para C_j . Se as unidades C_{j-1} e C_{j+1} tem permissão para aprender o vetor de entrada, encontre seus novos pesos.



11- **Kohonen (Fausett, 1994)** Repita o procedimento do exercício anterior para o vetor de entrada $(0,5; 0,5)$ e taxa de aprendizado em 0,1.

12- **Kohonen (Fausett, 1994)** Considere um mapa de Kohonen com duas unidades de saída e cinco unidades de entrada. Os vetores de pesos para as unidades de saída são:

$$w_1 = (1,0 \ 0,8 \ 0,6 \ 0,4 \ 0,2)$$

$$w_2 = (0,2 \ 0,4 \ 0,5 \ 0,8 \ 1,0)$$

Use o quadrado da distância euclidiana para encontrar a unidade de saída vencedora para o vetor de entrada:

$$X = (0,5 \ 1,0 \ 0,5 \ 0,0 \ 0,0)$$

Assuma que a taxa de aprendizado é 0,2 e encontre os novos pesos da unidade vencedora.

13- **Kohonen (Fausett, 1994)** Automatize o processo de treinamento de um SOM com base no seguinte exercício: use duas unidades de entrada, 50 unidades de saída, topologia linear. Permita que a unidade vencedora e seus vizinhos topológicos aprendam – se a unidade J é a vencedora, as unidades J-1 e J+1 também aprendem, a menos que J-1 < 1 ou J+1 > 50. Use uma taxa de aprendizado inicial de 0,5 e graduação a decresça para 0,01 (considerando 100 épocas). Os pesos iniciais das unidades de saída devem ser estabelecidos aleatoriamente com números no intervalo [-1, 1]. Gere os dados de treinamento como segue:

- Escolha dois números aleatórios entre -0,5 e 0,5, e os chame de x_1 e x_2 .
- Coloque o vetor (x_1, x_2) no conjunto de treinamento se $x_1^2 + x_2^2 < 0,25$.
- Repita até que você tenha 100 vetores de treinamento.

A cada 10 épocas de treinamento, plote as unidades de saída (usando seus vetores de pesos como posições em um plano euclidiano bi-dimensional); desenha uma linha conectando C1 a C2, C2 a C3 e assim por diante, para mostrar as relações topológicas.

* Você deverá começar com uma desorganização grande. Mas ela irá gradualmente se tornando uma linha na região onde existem vetores de dados.

14- **Clustering (Han et al., 2011)** Dado dois objetos (dados, vetores) representados pelas tuplas (22, 1, 42, 10) e (20, 0, 36, 8):

- Compute a distância Euclidiana entre os dois objetos;
- Compute a distância Manhattan entre os dois objetos.

15- **Clustering (Han et al., 2011)** Suponha que a tarefa é agrupar (clustering) os seguintes oito objetos (com (x,y) representando localização no espaço vetorial) em três grupos (clusters):

$$A_1(2,10) \ A_2(2,5) \ A_3(8,4) \ B_1(5,8) \ B_2(7,5) \ B_3(6,4) \ C_1(1,2) \ C_2(4,9)$$

A função de distância utilizada é a distância Euclidiana. Suponha, inicialmente, que nós associamos A_1 , B_2 e C_1 como centro de cada um dos grupos. Use o algoritmo **k-means** para mostrar somente:

- Os três centros dos grupos depois da primeira iteração do algoritmo;
- Os três grupos (clusters) finais.

16- **Clustering (Han et al., 2011)** Agrupamento tem sido popularmente reconhecido como uma tarefa importante de análise de dados para vários domínios de aplicação.

- Uma aplicação que usa agrupamento como o principal meio de resolução de problemas;
- Uma aplicação que usa agrupamento como uma ferramenta de pré-processamento para preparar dados para outras técnicas que resolverão o problema.

17- **Rede Neural (Haykin, 1999)** Uma rede feedforward totalmente conectada com 10 neurônios de entrada, 2 camadas escondidas, uma com 4 neurônios e outra com 3 neurônios, e um único neurônio de saída. Construa o gráfico da arquitetura desta rede.

18- **Rede Neural (Haykin, 1999)** A figura abaixo mostra um gráfico de fluxo de sinal de uma rede feedforward 2-2-2-1. A função de ativação $\varphi(\cdot)$ denota a função logística. Escreva o mapeamento de entrada-saída definido para esta rede. Então, suponha que o neurônio de saída do gráfico opera em sua região linear. Escreva o mapeamento de entrada-saída definido para esta rede.

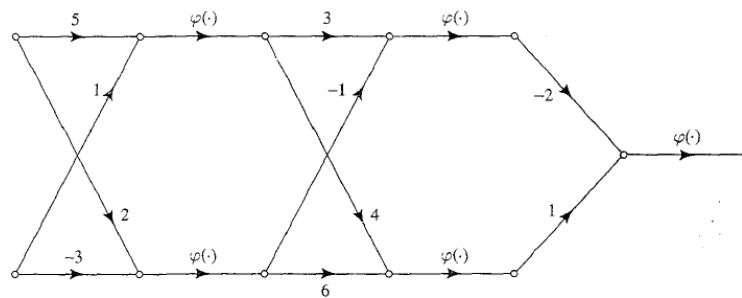


FIGURE P1.13

19- **Rede Neural (Haykin, 1999)** Usando a mesma figura do exercício anterior, suponha que biases iguais a -1 e +1 sejam aplicados para os neurônios superior e inferior da primeira camada escondida, e biases iguais a +1 e -2 são aplicados para os neurônios superior e inferior da segunda camada escondida. Escreva uma nova forma para o mapeamento entrada-saída da rede.

20- **Perceptron (Haykin, 1999)** O Perceptron pode ser usado para executar várias funções lógicas. Mostre a implementação das funções lógicas binárias AND, OR e COMPLEMENTO. Uma limitação básica do Perceptron é que ele não pode implementar a função lógica OR EXCLUSIVO. Explique as razões para essa limitação.

21- **XOR problem (Haykin, 1999)** A figura abaixo mostra uma rede neural com um neurônio escondido, para resolução do problema XOR. Mostre que essa rede resolver o problema XOR (a) construindo regiões de decisão, e (b) uma tabela verdade para a rede.

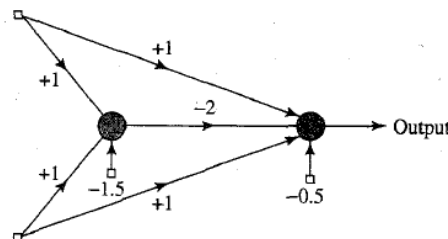


FIGURE P4.1

22- **Backpropagation (Haykin, 1999)** Investigue o uso do algoritmo de aprendizado backpropagation usando funções sigmoide para construir os seguintes mapeamentos:

$$f(x) = 1/x, \text{ para } 1 \leq x \leq 100$$

$$f(x) = \log_{10} x, \text{ para } 1 \leq x \leq 10$$

$$f(x) = \exp(-x), \text{ para } 1 \leq x \leq 10$$

$$f(x) = \sin(x), \text{ para } 0 \leq x \leq \pi/2$$

Para cada mapeamento, faça o seguinte:

- Determine dois conjuntos de dados, um para treinar a rede e outro para testar a rede.
- Use o conjunto de dados de treinamento para computação os pesos sinápticos da rede, assumido que ela tem uma camada escondida.
- Avalie a acurácia de computação da rede usando os conjuntos de dados de teste.

Em todos os casos, use uma camada escondida, mas varie o número de neurônios escondidos. Investigue como o desempenho da rede é afetado pelo tamanho da camada escondida.

23- Kohonen (Haykin, 1999) Diz-se que o algoritmo SOM baseado em aprendizado competitivo não possui nenhuma tolerância contra falhas de hardware, mas o algoritmo é tolerante a erros, uma vez que uma pequena perturbação aplicada ao vetor de entrada faz com que a saída pule do neurônio vencedor para o vizinho. Discuta a implicação destas afirmações.

24- Kohonen (Haykin, 1999) Neste experimento, nós usamos simulações computacionais para investigar o algoritmo SOM aplicado a lattice uni-dimensional com entradas bi-dimensionais. O lattice é formado por 65 neurônios. As entradas consistem em pontos (vetores) aleatórios uniformemente distribuídos dentro de uma área triangular (veja a figura). Compute o mapa produzido pelo algoritmo SOM depois de 0, 20, 100, 1000, 10.000, e 25.000 iterações.

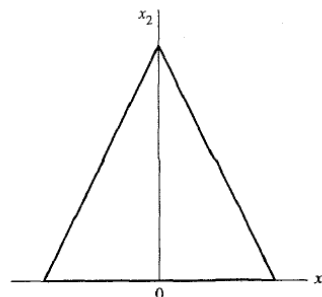


FIGURE P9.9

25- Kohonen (Haykin, 1999) Considere um lattice bi-dimensional de neurônios treinados com entradas distribuídas no espaço tri-dimensional. O lattice consiste em 10 x 10 neurônios. A entrada é uniformemente distribuída em um volume “magro” definido por $\{(0 < x_1 < 1), (0 < x_2 < 1), (0 < x_3 < 0,2)\}$. Use o algoritmo SOM para computar a projeção bi-dimensional do espaço de entrada depois de 50, 1.000 e 10.000 épocas.

26- Kohonen (Haykin, 1999) A propriedade de ordenamento topológico do algoritmo SOM pode ser usada para formar uma representação bi-dimensional abstrata de um espaço de entrada de dimensão alta. Para investigar esta forma de representação, considere um lattice bi-dimensional consistindo em 10 x 10 neurônios que é treinado com uma entrada consistindo em quatro nuvens gaussianas, C_1 , C_2 , C_3 e C_4 , em um espaço oito-dimensional. Todas as nuvens têm variância unitária, mas diferentes centros. Os centros são alocados nos pontos $(0, 0, 0, \dots, 0)$, $(4, 0, 0, \dots, 0)$, $(4, 4, 0, \dots, 0)$ e $(0, 4, 0, \dots, 0)$. Compute o mapa

produzido pelo algoritmo SOM, com cada neurônio do mapa sendo rotulado com a classe mais representada pelos pontos de entrada ao redor dela.

27- Modelagem de rede neural (Silva et al, 2016) Proponha a arquitetura de uma rede neural MLP que é capaz de implementar a classificação dos dados do conjunto:

- *Optical Recognition of Handwritten Digits*,
- *Car Evaluation Data Set*
- *20 Newsgroup Data Set*

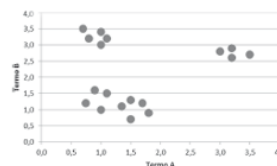
disponíveis no repositório UCI Machine Learning (<http://archive.ics.uci.edu/ml/index.php>).

28- Kohonen (Silva et al, 2016) Na figura abaixo são apresentados: (a) um conjunto de dados; (b) o gráfico de dispersão dos exemplares desses conjuntos; (c) os espaços matricial e vetorial (inicial) de um SOM. A partir dessas informações e da escolha de um exemplar do conjunto de dados, responda:

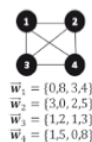
- Qual é o neurônio vencedor? Mostre os cálculos que levam à descoberta do neurônio vencedor.
- Apresente os cálculos para atualização do neurônio vencedor e de um de seus vizinhos topológicos, considerando uma função de vizinhança gaussiana e taxa de aprendizado = 1.
- Apresente o gráfico de dispersão referente ao espaço vetorial, considerando os exemplares, os neurônios com seus pesos iniciais e, para o caso dos dois neurônios atualizados, com seus pesos atualizados – use cores diferentes na plotagem dos neurônios para indicar as mudanças de seus posicionamentos.

	Termo A	Termo B
\vec{x}_1	1,0	1,0
\vec{x}_2	0,8	1,2
\vec{x}_3	1,1	1,5
\vec{x}_4	0,9	1,6
\vec{x}_5	1,4	1,1
\vec{x}_6	1,5	0,7
\vec{x}_7	1,0	3,0
\vec{x}_8	0,8	3,2
\vec{x}_9	0,7	3,5
\vec{x}_{10}	1,0	3,4
\vec{x}_{11}	1,1	3,2
\vec{x}_{12}	3,0	2,8
\vec{x}_{13}	3,2	2,9
\vec{x}_{14}	3,5	2,7
\vec{x}_{15}	3,2	2,6
\vec{x}_{16}	1,5	1,3
\vec{x}_{17}	1,8	0,9
\vec{x}_{18}	1,7	1,2

(a) Conjunto de dados



b) gráfico de dispersão



c) Espaço matricial e vetorial (iniciais) de uma rede SOM