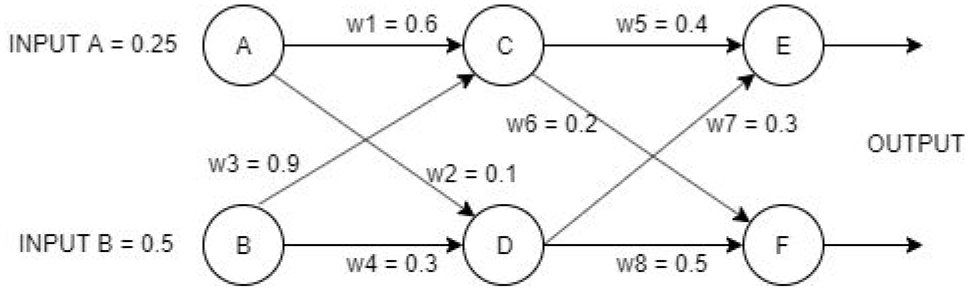
Take-Home Practice on Neural Networks (Non-Graded) -- Solutions

1. Backpropagation example with 2 inputs, 2 hidden units and 2 outputs



Assume that the neurons have sigmoid activation function, learning rate of 1 and answer the following questions:

a) Perform a forward pass on the network and determine the outputs at E and F.
b) Perform back-propagation on the output layer neurons($target_E$=1 and $target_F$=0) and determine the updated weights for w5, w6, w7 and w8.
c) Perform back-propagation on the hidden layer neurons and determine the updated weights for w1, w2, w3, and w4.

Useful sigmoid values:

x	0.175	0.4	0.42	0.5	0.6	0.73	0.82
sigmoid(x)	0.543	0.598	0.6	0.622	0.645	0.67	0.694

Solution 1a)

in_C = (0.25*0.6)+(0.5*0.9) = 0.15 + 0.45 = 0.6

out_C = sigmoid(0.6) = 0.645

in_D = (0.25*0.1)+(0.5*0.3) = 0.025 + 0.15 = 0.175

out_D = sigmoid(0.175) = 0.543

in_E = (0.645*0.4)+(0.543*0.3) = 0.258 + 0.162 = 0.42

out_E = sigmoid(0.42) = 0.6

in_F = (0.645*0.2)+(0.543*0.5) = 0.129 + 0.271 = 0.4

out_F = sigmoid(0.4) = 0.598

Solution 1b)

$$\delta_E = out_E(1 - out_E)(target_E - out_E) = 0.6(1 - 0.6)(1 - 0.6) = 0.096$$

$$\delta_F = out_F(1 - out_F)(target_F - out_F) = 0.598(0 - 0.598)(1 - 0.598) = -0.14$$

$$w_5^* = w_5 + \eta \cdot \delta_E \cdot out_C = 0.4 + 1 * (0.096 \cdot 0.645) = 0.46$$

$$w_6^* = w_6 + \eta \cdot \delta_F \cdot out_C = 0.2 + 1 * (-0.14 \cdot 0.645) = 0.11$$

$$w_7^* = w_7 + \eta \cdot \delta_E \cdot out_D = 0.3 + 1 * (0.096 \cdot 0.543) = 0.352$$

$$w_8^* = w_8 + \eta \cdot \delta_F \cdot out_D = 0.5 + 1 * (-0.14 \cdot 0.543) = 0.42$$

Solution 1c)

$$\delta_C = out_C(1 - out_C)(\delta_E \cdot w_5 + \delta_F \cdot w_6) = 0.645(1 - 0.645)(0.096 \cdot 0.46 + -0.14 \cdot 0.11) = 0.007$$

$$\delta_D = out_D(1 - out_D)(\delta_E \cdot w_7 + \delta_F \cdot w_8) = 0.543(1 - 0.543)(0.096 \cdot 0.352 + -0.14 \cdot 0.43) = -0.007$$

$$w_1^* = w_1 + \eta \cdot \delta_C \cdot in_A = 0.6 + 1 * (0.007 \cdot 0.25) = 0.601$$

$$w_2^* = w_2 + \eta \cdot \delta_D \cdot in_A = 0.1 + 1 * (-0.007 \cdot 0.25) = 0.09825$$

$$w_3^* = w_3 + \eta \cdot \delta_C \cdot in_B = 0.9 + 1 * (0.007 \cdot 0.5) = 0.904$$

$$w_4^* = w_4 + \eta \cdot \delta_D \cdot in_B = 0.3 + 1 * (-0.007 \cdot 0.5) = 0.297$$

2. The following matrix represents the weights of a hopfield network with the vectors $V_1(0, 1, 0, 1)$, $V_2(1, 0, 0, 1)$ stored. Use this matrix and answer the following questions.

$$W = \begin{matrix} 0 & -2 & 0 & 0 \\ -2 & 0 & 0 & 0 \\ 0 & 0 & 0 & -2 \\ 0 & 0 & -2 & 0 \end{matrix}$$

   a. What is the weight matrix when a new vector V_3 (0, 1, 1, 0) is added to this network?
   b. Use the weight matrix obtained in part a, and assume that the order of node updates is 2, 3, 4, 1. What memory does the network converge to if Vin is (1, 0, 1, 0)? (Show the input vector after each update and the final attractor that the network converges to)

Solution 2a):

The weight matrix for storing just V_3 is:

$$W_3 = \begin{matrix} 0 & -1 & -1 & 1 \\ -1 & 0 & 1 & -1 \\ -1 & 1 & 0 & -1 \\ 1 & -1 & -1 & 0 \end{matrix}$$

This is calculated using $W_3^{i,j} = (2*V_3^i - 1) x (2*V_3^j - 1)$.

The final weight matrix $W = W + W_3$

$$W = \begin{matrix} 0 & -3 & -1 & 1 \\ -3 & 0 & 1 & -1 \\ -1 & 1 & 0 & -3 \\ 1 & -1 & -3 & 0 \end{matrix}$$

Solution2b):

Iteration 1:
Node updates order: [2, 3, 4, 1]

Updating node 2:
V_2in: W[2] * V_2 = (-3, 0, 1, -1) * (1, 0, 1, 0) = -2 <0 => V_2in = 0 (didn't change), Vin = (1, 0, 1, 0)

Updating node 3:
V_3in: (-1, 1, 0, -3) * (1, 0, 1, 0) = -1 <0 => V_3in = 0 (changed), Vin = (1, 0, 0, 0)
Updating node 4:
V_4in: (1, -1, -3, 0) * (1, 0, 0, 0) = 1 >=0 => V_4in = 1 (changed), Vin = (1, 0, 0, 1)

Updating node 1:
V_1in: (0, -3, -1, 1) * (1, 0, 0, 1) = 1 >=0 => V_1in = 1 (didn't change), Vin = (1, 0, 0, 1)

Since there were updates made to the input vector, we need to perform another iteration to test for convergence.

Iteration 2:
Node updates order: [2, 3, 4, 1]

Updating node 2:
V_2in: (-3, 0, 1, -1) * (1, 0, 0, 1) = -4 <0 => V_2in = 0 (didn't change), Vin = (1, 0, 0, 1)

Updating node 3:

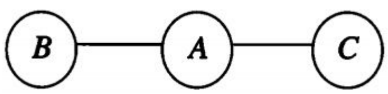
V_3in: (-1, 1, 0, -3) * (1, 0, 0, 1) = -4 <0 => V_3in = 0 (didn't change), Vin = (1, 0, 0, 1)

Updating node 4:
V_4in: (1, -1, -3, 0) * (1, 0, 0, 1) = 1 >=0 => V_4in = 1 (didn't change), Vin = (1, 0, 0, 1)

Updating node 1:
V_1in: (0, -3, -1, 1,) * (1, 0, 0, 1,) = 1 >=0 => V_1in = 1 (didn't change), Vin = (1, 0, 0, 1)

Since there were no updates made to the input vector, we conclude that the network converged to the stored memory V_2 (1, 0, 0, 1).

3. Consider the input vectors
I_1 = (1.1, 1.7, 1.8), I_2 = (0,0,0), I_3 = (0,0.5, 1.5), I_4 = (1,0,0), I_5 = (0.5,0.5,0.5), I_6 = (1, 1, 1).
We are using a 3 node Self-Organizing Map network with initial weights W(0) as shown below:



$$W(0) = \begin{pmatrix} w_A: & 0.2 & 0.7 & 0.3 \\ w_B: & 0.1 & 0.1 & 0.9 \\ w_C: & 1 & 1 & 1 \end{pmatrix}$$

Using the above information, answer the following questions:

a. If the neighborhood radius is R = 1 (ie. we consider neighbors at distance 1 of a node when updating weights), learning rate η = 0.5, and we consider the inputs in order (I_1 to I_6), **what is the weight matrix after the first epoch** (after all inputs are presented once)**?**

b. What clusters (A, B or C) are the inputs assigned to at the end of the first epoch?
c. Assuming a geometric decrease in learning rate of 0.5, what is the learning rate for the second and third epochs?

Solution 3a):

For the first input I_1 (1.1, 1.7, 1.8):
Euclidean Distance of A from I_1 = $D^2(A, I_1) = \sum_{j=1 \text{ to } 3} (I_1^{(j)} - W_A^{(j)})^2$
$= (1.1 - 0.2)^2 + (1.7 - 0.7)^2 + (1.8 - 0.3)^2 = 4.1$
Similarly, $D^2(B, I_1) = 4.4$ and $D^2(C, I_1) = 1.1$

Since C has the shortest distance to I_1, we pick C as the winning neuron, and update the weights of C and all the neurons at a distance of 1 (since R=1) from it, which is only neuron A, thus
We update W_C and W_A:

$W_{C1} = W_{C1} + \eta * (I_{11} - W_{C1})) = 1 + 0.5 *(1.1 - 1) = 1.05$
$W_{C2} = W_{C2} + \eta * (I_{12} - W_{C1})) = 1 + 0.5 *(1.7 - 1) = 1.35$
$W_{C3} = W_{C3} + \eta * (I_{13} - W_{C3})) = 1 + 0.5 *(1.8 - 1) = 1.4$
Thus the updated weight for W_C is:
$W_C(1) = [1.05, 1.35, 1.4]$

We can similarly compute $W_A(1)$:
$W_A(1) = [0.65, 1.2, 1.05]$

Since B was not updated, $W_{\mathbf{B}}(1) = W_B(0)$ (no change).

The new weight matrix W(1) is then:

$$W(1) = \begin{pmatrix} w_A: & 0.65 & 1.2 & 1.05 \\ w_B: & 0.1 & 0.1 & 0.9 \\ w_C: & 1.05 & 1.35 & 1.4 \end{pmatrix}$$

For the next input I_2 (0, 0, 0):
We compute the distances of I_2 from A, B and C using the updated weight matrix from the last instance, ie. W(1).

$D^2(A, I_2) = 3$, $D^2(B, I_2) = 0.8$, $D^2(C, I_2) = 4.9$

Since B is the closest, we pick B as the winning neuron and update weights of B and A.

$$W(2) = \begin{pmatrix} w_A: & 0.325 & 0.6 & 0.525 \\ w_B: & 0.05 & 0.05 & 0.45 \\ w_C: & 1.05 & 1.35 & 1.4 \end{pmatrix}$$

For the next input I_3 (0, 0.5, 1.5):
We compute the distances using W(2), and get $D^2(A, I_2) = 1.1$, $D^2(B, I_2) = 1.3$, $D^2(C, I_2) = 1.7$
Hence we pick A as the winner, and update weights of A along with its neighbors with distance 1, which includes both B and C. (Thus in this case the weights of all 3 neurons get updated).

$$W(3) = \begin{pmatrix} w_A: & 0.16 & 0.55 & 1.01 \\ w_B: & 0.025 & 0.275 & 0.975 \\ w_C: & 0.525 & 0.925 & 1.45 \end{pmatrix}$$

Students can perform similar computations for $\mathbf{I_4}$, $\mathbf{I_5}$, **and** $\mathbf{I_6}$, to get the final weight matrix after the first epoch W(6):

$$W(6) = \begin{pmatrix} w_A: & 0.77 & 0.69 & 0.75 \\ w_B: & 0.51 & 0.32 & 0.49 \\ w_C: & 0.76 & 0.86 & 0.99 \end{pmatrix}$$

Solution 3b):

To determine the clusters we compute the euclidean distances of the 6 inputs from the 3 neurons using the weight matrix W(6), and assign the inputs to the cluster represented by their closest neuron.

The following table gives the values of distances and cluster assignments for all the inputs:

Input	Distance to A	Distance to B	Distance to C	Assigned Cluster
I_1	2.2	4.0	1.5	C
I_2	1.6	0.6	2.3	B
I_3	1.2	1.3	1.0	C
I_4	1.1	0.6	1.8	B
I_5	0.2	0.03	0.4	B
I_6	0.2	1.0	0.07	C

Solution 3c):

The learning rate for first epoch is $\eta(1) = 0.5$.
Since the rate decreases at a rate of 0.5, the learning rate for the second epoch is $\eta(2) = 0.5\ \eta(1) = 0.5 * 0.5 = 0.25$
Similarly, the learning rate for the third epoch is $\eta(3) = 0.5\ \eta(2) = 0.125$