Identikey: ___________________________

Artificial Intelligence (F19)
Quiz 4 (QZ4)

First Name: ___________________________

Last Name: ___________________________

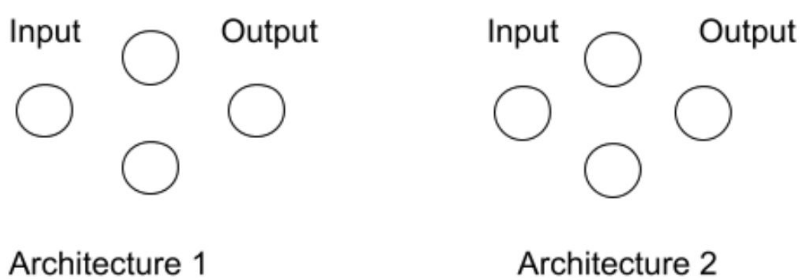
Overview
7.5 Point Quiz on Search (7.5% of final grade)
30 minute closed book quiz

Learning Objective
This assignment satisfies learning objective 1 (LO1) as specified in the syllabus. You will demonstrate conceptual understanding of the core AI topics.

The quiz is worth 7.5 points total but there are 8 points available. This means that you can lose 0.5 points, but still earn a perfect score of 7.5 points on the quiz. If you score above 7.5 points, your total score will be rounded down to 7.5 points.

Questions

1. [Total Points: 3.0] Multiple choice/short answer questions
   a. [Points: 0.5] What are the two main types of neural network architectures? Write the names of each below the diagram and illustrate the connections among neurons in the diagram.



Solution:
The two types are Feed-Forward networks and Recurrent Networks.
One of the architectures should have self-connections/loops while the other should not.

   a. [Points: 0.25] Which of the following gives non-linearity to a neural network?
      1. Gradient descent
      2. Sigmoid activation function
      3. Recurrent connections
      4. None of the above

Solution. (2)

   b. [Points: 0.25] Which of the following neural network architectures preserves topological (spatial) properties of the input space in the output space?

1. Multilayer Perceptron
2. Hopfield Network
3. Elman Network
4. Kohonen's Self Organizing Map

Solution: iv) Kohonen's Self Organizing Map

c. [Points: 0.25] Which of the following techniques can help overcome the local minima problem in training neural networks?

1. Increasing the number of neurons in the hidden layer
2. Adding more hidden layers
3. Adding a momentum term to the weight update
4. Using a different activation function other than a sigmoid
5. Increasing the learning rate

Solution: iii) Adding momentum term to the weight update

d. [Points: 0.5] Consider the following equation used to calculate error in the final layer of a neural network using back propagation. Identify the different components of the formula. *Hint – there are two components of interest.*

$$\delta = (\text{target}_a - \text{out}_a)\,\text{out}_a(1_a - \text{out}_a)$$

Solution: The $\text{out}_a(1_a - \text{out}_a)$ term comes from the derivative of the activation function. In this case the sigmoid activation. The other term $(\text{target}_a - \text{out}_a)$ is the error in predicted output.

e. [Points: 0.5] Consider the following steps for backpropagation.
1. Calculate error between the actual value and the predicted value
2. Reiterate until you find the best weights of network
3. Pass an input through the network and get values from output layer
4. Initialize random weight
5. Change the weights of each neuron that contribute to the error

Arrange these steps in the correct order.

Solution: 4, 3, 1, 5, 2

f. [Points 0.25] Assume memories A, B, and C are stored in a Hopfield network. Is the network always guaranteed to return one of the stored memories during retrieval? *Provide a brief justification to get credit.*

Solution: No. Hopfield networks may converge to other attractor basins that are different from the stored memories. These are "spurious memories".

g. [Points: 0.5] What is the role of the hidden layers in a neural network?

Solution: The hidden layers recode the input into different levels of abstraction. This allows the neural network to model increasingly complex data.

2. [Total Points: 1.0] Consider the following weights of a 3 neuron Hopfield network:
$W_{12} = 1$
$W_{23} = -1$

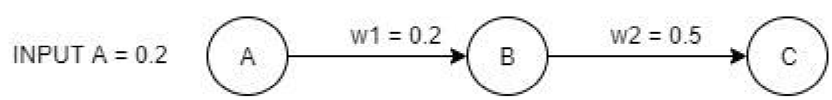
a. [Points: 0.25] What is the value of W_{21}?
Solution: 1

b. [Points: 0.25] What is the value of W_{22}?
Solution: 0

c. [Points: 0.5] What was the likely input to the network. You can assume that the weights were initialized to 0 and that only a single memory has been stored. *Hint: Remember that the Hopfield network uses hebbian learning for encoding memories.*

Solution:
Multiple possible answers. Any that show agreement between neurons 1&2 and disagreement between 2&3 are acceptable. Eg. [1, 1, 0] or [0, 0, 1]

3. [Total Points: 1.5] Backpropagation
Consider the following network with given weights and no bias to nodes. Assume sigmoid activation and learning rate of 1.



Use the following values for sigmoid that have been rounded for simplicity:

X	0.1	0.2	0.3	0.4
sigmoid(x)	0.5	0.5	0.6	0.6

a. [Points: 0.5] After processing input A (0.2), hidden node B outputs 0.6. What is the output of node C? *Show the calculations.*

$out_C =$

Solution:

$in_C = (0.6*0.5) = 0.3$

$out_C = \text{sigmoid}(0.3) = 0.6$

b. [Points: 1.0] Assume that the updated weight for $w_2 = 0.6$, and error on output $\delta_c = 0.1$, compute the updated weight w_1 using backpropagation. *You can simply plug the correct values in the equation..*

Solution:

$\delta_C = 0.1,\ W_2 = 0.6$

$\delta_B = out_B(1 - out_B)(\delta_C \cdot w_2) = 0.6\ (1\text{-}0.6)(0.1 * 0.6) = 0.6 * 0.4 * 0.06 = 0.014$

$W_1 = W_1 + \eta * \delta_B * in_A = 0.2 + (0.6)(1\text{-}0.6)(0.1*0.6) * 0.2 = 0.2 + 0.014*0.2 = 0.2028$

4. [Total Points: 2.5] Kohonen's Self Organizing Maps

a. [Points: 0.5] Consider a Self-Organizing Map (SOM) with 5 neurons used to model data where the input vectors have 500 attributes (dimensions):
   1. How many layers does the SOM have?
   2. How many total weights are in the SOM?

Solution: It has only 2 layers - 1 input layer and 1 output layer. Since each attribute of input is connected to all the 5 neurons, the number of weights is 5*500 = 2500 weights.

b. [Points: 0.5] Consider a Self-Organizing map (SOM) with two neurons and weights shown below. Which neuron will be chosen for updating for input vector [1, 1, 0, 1]? *Show your work to get credit.*

Neuron A	2	0	1	0
Neuron B	1	2	1	0

Solution:
D(I, A) = (2-1)^2 + (0-1)^2 + (1-0)^2 + (0-1)^2 = 4
D(I, B) = (1-1)^2 + (2-1)^2 + (1-0)^2 + (0-1)^2 = 3
Neuron B is closer hence will be chosen as the winner.

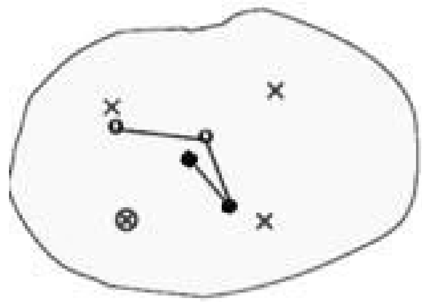
c. [Points: 0.5] The weights of an SOM neuron are [0.5, 0, 0, 0.25]. What is the new weight for $W_{11} = 0.5$ after processing input I = [1 0 0 1] and learning rate $\alpha = 0.1$. *You can simply plug the correct values in the equation..*

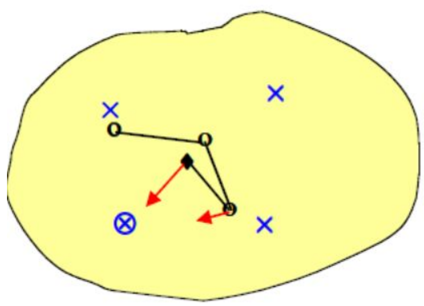
Solution: $W_{ij} = W_{ij} + \alpha(I_i - W_{ij})$

$W_{11} = W_{11} + 0.1\ (I_{11} - W_{11}) = 0.5 + 0.1\ (1 - 0.5) = 0.5 + 0.1*0.5 = 0.55$

Identikey: ___________________________

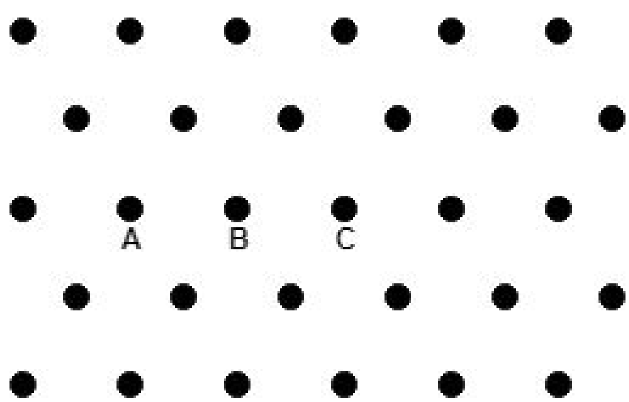
d. [Points: 0.5] Graphically illustrate the concept of a weight update in an SOM using the following diagram where the X with the circle is the input being processed, and the solid square and solid circle represent the neuron it is close to and its neighbor (R=1) respectively.





Solution:

e. [Points: 0.5] Consider the following SOM (neurons are dots) with neighborhood radius R = 1. Circle the neurons that will be updated if neuron 'B' is the winner and a hexagonal neighborhood us used.



Solution: All nodes at distance 1 from B must be circled.

Equations Sheet

Kohonen Self Organizing Maps Algorithm

Step 0. Initialize weights w_{ij}. (Possible choices are discussed below.)
Set topological neighborhood parameters.
Set learning rate parameters.

Step 1. While stopping condition is false, do Steps 2–8.

Step 2. For each input vector **x**, do Steps 3–5.

Step 3. For each j, compute:

$$D(j) = \sum_i (w_{ij} - x_i)^2.$$

Step 4. Find index J such that $D(J)$ is a minimum.

Step 5. For all units j within a specified neighborhood of J, and for all i:

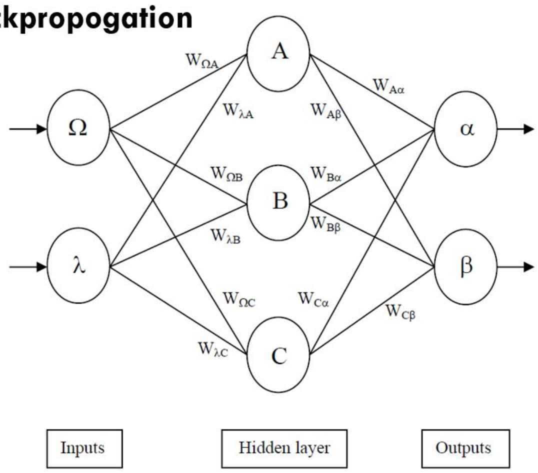
$$w_{ij}(\text{new}) = w_{ij}(\text{old}) + \alpha[x_i - w_{ij}(\text{old})].$$

Step 6. Update learning rate.

Step 7. Reduce radius of topological neighborhood at specified times.

Step 8. Test stopping condition.

Backpropogation



$$\delta_\alpha = \text{out}_\alpha (1 - \text{out}_\alpha) (\text{Target}_\alpha - \text{out}_\alpha)$$

$$W^{+}_{A\alpha} = W_{A\alpha} + \eta\delta_{\alpha}\, out_{A}$$

$$\delta_{A} = out_{A}\,(1 - out_{A})\,(\delta_{\alpha}W_{A\alpha} + \delta_{\beta}W_{A\beta})$$

$$W^{+}_{\lambda A} = W_{\lambda A} + \eta\delta_{A}\, in_{\lambda}$$

Encoding memories in a Hopfield network

$$W_{ij} = \sum_{s=1}^{n}\left(2V_s^i - 1\right)\left(2V_s^j - 1\right)$$

s is number of patterns
Vs are input patterns