F/G/TY

Reg. No:

VIT
Vellore Institute of Technology

Summer Term Final Assessment Test – July 2025

Course: **MDI4009 - Neural Networks and Deep Learning**

Class NBR(s): **0199**

Slot: **G1+TG1+G2+TG2**

Time: **Three Hours**

Max. Marks: **100**

- **KEEPING MOBILE PHONE/ANY ELECTRONIC GADGETS, EVEN IN 'OFF' POSITION IS TREATED AS EXAM MALPRACTICE**
- **DON'T WRITE ANYTHING ON THE QUESTION PAPER**

Answer ALL Questions

(10 X 10 = 100 Marks)

1. a) What is the purpose of using activation function? Explain about various activation functions. **[4]**

   b) Describe any three categories from the taxonomy of neural networks and illustrate them with suitable diagrams. **[6]**

2. a) Discuss the discrete and single continuous perceptron models with its training algorithms. **[6]**

   b) Explain the concept of the credit assignment problem in deep learning. **[4]**

3. You are designing a neural network model for a binary classification problem using a sigmoid activation function. Derive the backpropagation algorithm step-by-step for the following two cases:

   Case 1: Training rule for output unit weights

   Case 2: Training rule for hidden unit weights

   For both cases, apply the stochastic gradient descent rule to update the weights. Clearly indicate the flow of derivatives, chain rule usage, and final weight update formulae.

4. Let us consider a hetero-associative memory neural network for pattern association. The task is to store input–output pattern pairs using Outer Product Rule and Hebbian Learning Rule. The training data is given below:

Input	S1	S2	S3	S4	T1	T2
1st	1	0	1	0	1	0
2nd	1	0	0	1	1	0
3rd	1	1	0	0	0	1
4th	0	0	1	1	0	1

Using the given input-output pattern pairs, apply the Outer Product Rule and Hebbian Learning Rule to compute the weight matrix W for a hetero-associative memory network.

5. Implement ART1 learning on input vectors (1,1,0,0), (1,0,0,0), (0,0,1,1), (0,0,0,1) using a vigilance parameter of 0.6 and learning rate 1.5. Show step-by-step weight updates and final cluster assignments. Assume all necessary initial weights and parameters if required.

6. Design a Kohonen Self-Organizing Map (SOM) for pattern clustering. The SOM has three cluster units (U1, U2, and U3), and you are given five training samples, each of dimension 3. The initial weight vectors for each unit are:

U1 = [0.2 0.4 0.6 0.8]

U2 = [0.5 0.3 0.7 0.9]

Apply the square of the Euclidean distance to find the winning cluster unit for the training samples

Z1: (1,0,1,1) Z2: (1,0,1,0) Z3: (0,1,0,1) Z4: (0,1,1,0)

Utilize the learning rate of 0.6 and calculate the new weights for the winning unit.

7. Describe the types of regularization methods in deep learning and outline how deep neural networks are trained through the backpropagation algorithm.

8. Discuss the core concepts behind Generative Adversarial Networks (GANs) and elaborate the loss functions of the generator and discriminator during the training phase. How does a GAN work?

9.a) Analyze the architectural design and functionalities of AlexNet and GoogLeNet. Illustrate each model with appropriately labeled diagrams, and apply your understanding by explaining how these models can be utilized in a real-world image classification application.

OR

9.b) Construct a Convolutional Neural Network (CNN) to process a grayscale image of size 8×8 pixels. Start with a 3×3 convolutional kernel applied using a stride of 1 and padding of 1, and compute the dimensions of the resulting feature map. Then apply average pooling using a 2×2 window with a stride of 2 to reduce the spatial dimensions. Flatten the pooled output and connect it to two fully connected layers with 128 neurons each. Use a ReLu activation function at the output layer to perform classification.

10.a) i. Narrate the various approaches to implement reinforcement learning algorithms. **[4]**

ii. Elaborate the phases of the Monte Carlo Tree Search (MCTS) algorithm. Illustrate each phase with a simple example. **[6]**

OR

10.b) Provide a detailed derivation of the Q-learning method for action-value estimation. Explain how the Bellman equation guides the update mechanism, and solve a sample problem to demonstrate the algorithm in action.

⇔⇔⇔ **F/G/TY** ⇔⇔⇔