# Question Paper A - Formative Test I

**Course Code**: 21CSE558T **Course Title**: Deep Neural Network Architectures **Test Duration**: 50 minutes **Total Marks**: 25 **Coverage**: Modules 1-2 (Weeks 1-6)

## Instructions:

1. Answer ALL questions in Part A (MCQs)
2. Answer ANY THREE questions from Part B (SAQs)
3. Each MCQ carries 1 mark, each SAQ carries 5 marks
4. Total marks: 25 (Part A: 10 marks + Part B: 15 marks)

# PART A: Multiple Choice Questions (10 × 1 = 10 marks)

### Q1. The XOR problem cannot be solved by a single perceptron because:

**Module**: 1 | **Difficulty**: Easy | **CO**: CO-1 | **Source**: MCQ Q1

1. It requires too many inputs
2. It is not linearly separable
3. It needs multiple outputs
4. It requires complex activation functions

### Q2. Which activation function can output negative values?

**Module**: 1 | **Difficulty**: Easy | **CO**: CO-1 | **Source**: MCQ Q5

1. Sigmoid
2. ReLU
3. Tanh
4. Softmax

### Q3. The softmax activation function is primarily used for:

**Module**: 1 | **Difficulty**: Easy | **CO**: CO-1 | **Source**: MCQ Q10

1. Binary classification
2. Multi-class classification
3. Regression problems
4. Feature extraction

### Q4. The mathematical representation of a perceptron output is:

**Module**: 1 | **Difficulty**: Moderate | **CO**: CO-1 | **Source**: MCQ Q16

1. y = wx + b
2. y = σ(wx + b)
3. y = w²x + b
4. y = log(wx + b)

### Q5. Which gradient descent variant uses the entire dataset for each update?

**Module**: 2 | **Difficulty**: Easy | **CO**: CO-2 | **Source**: MCQ Q21

1. Stochastic Gradient Descent
2. Mini-batch Gradient Descent
3. Batch Gradient Descent
4. Adam Optimizer

### Q6. Overfitting occurs when:

**Module**: 2 | **Difficulty**: Easy | **CO**: CO-2 | **Source**: MCQ Q23

1. Model performs well on both training and test data
2. Model performs poorly on training data
3. Model performs well on training but poorly on test data
4. Model cannot learn from data

### Q7. The purpose of validation set is to:

**Module**: 2 | **Difficulty**: Easy | **CO**: CO-2 | **Source**: MCQ Q30

1. Train the model
2. Tune hyperparameters and monitor overfitting
3. Test final performance
4. Increase training data

### Q8. Which optimizer adapts the learning rate for each parameter individually?

**Module**: 2 | **Difficulty**: Moderate | **CO**: CO-2 | **Source**: MCQ Q35

1. SGD
2. Momentum
3. AdaGrad
4. Standard gradient descent

### Q9. Which problem is addressed by batch normalization?

**Module**: 2 | **Difficulty**: Moderate | **CO**: CO-2 | **Source**: MCQ Q40

1. Overfitting only
2. Internal covariate shift
3. Underfitting only
4. Memory optimization

### Q10. The learning rate finder technique helps to:

**Module**: 2 | **Difficulty**: Difficult | **CO**: CO-2 | **Source**: MCQ Q45

1. Find optimal architecture
2. Determine optimal learning rate range
3. Prevent overfitting
4. Reduce training time

# PART B: Short Answer Questions (Answer any 3 × 5 = 15 marks)

### Q11. Perceptron Forward Pass Calculation

**Module**: 1 | **Week**: 2 | **Difficulty**: Easy | **CO**: CO-1 | **Source**: SAQ Q1

You have a single perceptron with the following parameters: - Input vector: X = [0.8, 0.6] - Weight vector: W = [0.4, -0.7] - Bias: b = 0.3 - Activation function: Sigmoid

**Task:** Calculate the complete forward pass output of this perceptron. Show all intermediate steps including weighted sum calculation and activation function application.

### Q12. MLP Forward Propagation

**Module**: 1 | **Week**: 4 | **Difficulty**: Moderate | **CO**: CO-2 | **Source**: SAQ Q3

Given a 2-layer MLP with the following TensorFlow/Keras structure: - Input layer: 2 neurons - Hidden layer: 3 neurons with ReLU activation - Output layer: 1 neuron with sigmoid activation - Input: X = [1.2, -0.8] - Hidden layer weights: W1 = [[0.5, -0.3, 0.7], [0.2, 0.9, -0.4]] - Hidden layer bias: b1 = [0.1, -0.2, 0.3] - Output layer weights: W2 = [[0.6], [-0.5], [0.8]] - Output layer bias: b2 = [0.2]

**Task:** Perform complete forward propagation through this network. Calculate hidden layer outputs after ReLU activation, then compute the final sigmoid output.

### Q13. Activation Functions Comparison

**Module**: 2 | **Week**: 5 | **Difficulty**: Easy | **CO**: CO-2 | **Source**: SAQ Q5

Compare the computational efficiency of three activation functions in a neural network: - Input values: [-2.0, 0.0, 2.0] - Functions to evaluate: Sigmoid, ReLU, and Tanh

**Task:** Calculate the output of each activation function for all three input values. Show the mathematical formulas used and compute the exact numerical results.

### Q14. Batch Normalization Calculation

**Module**: 2 | **Week**: 6 | **Difficulty**: Moderate | **CO**: CO-2 | **Source**: SAQ Q7

Given a batch normalization layer in TensorFlow with the following statistics: - Input batch: [2.0, 4.0, 6.0, 8.0] - Learned parameters: γ (scale) = 1.5, β (shift) = 0.5 - Small constant: ε = 1e-5

**Task:** Perform batch normalization on this input batch. Calculate the batch mean, batch variance, normalized values, and final output after scaling and shifting. Show all intermediate calculations.

### Q15. Gradient Descent vs Momentum Comparison

**Module**: 2 | **Week**: 6 | **Difficulty**: Moderate | **CO**: CO-2 | **Source**: SAQ Q9

A neural network uses different optimization algorithms. Compare one iteration of gradient descent vs. momentum-based gradient descent: - Current gradient: ∇w = 0.4 - Learning rate: α = 0.1 - Current weight: w = 0.8 - For momentum: previous velocity v\_prev = 0.2, momentum coefficient β = 0.9

**Task:** Calculate the weight update for both standard gradient descent and momentum-based gradient descent. Show the velocity calculation for momentum method and compare the final weight updates.

## Distribution Summary:

* **MCQs**: Module 1: 4, Module 2: 6 | Easy: 4, Moderate: 5, Difficult: 1
* **SAQs**: Module 1: 2, Module 2: 3 | Easy: 2, Moderate: 3
* **Total Marks**: 25 | **Time**: 50 minutes