Mid Semester

Date: September 27, 2024 **Time:** 1030 -1230 hrs (Morning Session)

Subject Code: CS 365 **Subject Name:** Deep Learning

Q1	You are doing full batch gradient descent using the entire training set (not stochastic gradient descent). Is it necessary to shuffle the training data? Justify your answer.	[2]
Q2	You design a fully connected neural network architecture where all activations are sigmoids. You initialize the weights with large positive numbers. Is this a good idea? Explain your answer.	[2]
Q3	You are training a logistic regression model. You initialize the parameters with 0's. Is this a good idea? Explain your answer.	[2]
Q4	You would like to train a dog/cat image classifier using mini-batch gradient descent. You have already split your dataset into train, dev and test sets. The classes are balanced. You realize that within the training set, the images are ordered in such a way that all the dog images come first and all the cat images come after. A friend tells you: "you absolutely need to shuffle your training set before the training procedure." Is your friend right? Explain.	[2]
Q5	Suppose a deep learning model has a loss function $L(\theta) = \theta^2 + 5\theta + 6$ Using gradient descent with a learning rate of 0.1, compute the updated value of θ after one iteration, starting with $\theta_0 = 2$.	[2]
Q6	Consider three classification models with the following performance metrics. Identify which model is experiencing overfitting, underfitting, and which one represents a well-generalized model: • Model 1: Training error = 5%, Test error = 30% • Model 2: Training error = 20%, Test error = 22% • Model 3: Training error = 8%, Test error = 9%	[3]
Q7	In a fighting game, you are training a neural network to classify different moves (e.g.,punch, kick, block) based on visual input from the game. The neural network has a single hidden layer with two neurons. The activation function used is the sigmoid function, and the loss function is Mean Squared Error (MSE). (*At each stage, the values should be rounded to two decimal places.)* Given the following details for a single training example: I. Input features: $x_1 = 1.0$, $x_2 = 0.5$ II. Weights for the hidden layer: $w_{11} = 0.2$,$w_{12} = 0.4$,$w_{13} = 0.1$,$w_{14} = 0.3$ (Note: w_{11} & w_{12} connects x_1; w_{13} & w_{14} connects x_2) III. Weights for the output layer: $v_1 = 0.5$, $v_2 = 0.6$ IV. Biases: $b_1 = 0.1$ for hidden neurons, $b_{output} = 0.2$ for the output neuron V. Target output: $y_{target} = 1.0$ VI. Learning rate: η = 0.01 a) Perform the forward pass to compute the predicted output of the network. b) Calculate the error at the output using the MSE loss function. c) Using backpropagation, compute the gradients of the weights & bias in the output and hidden layers. d) Update the weights & bias using gradient descent.	[17]