Q1 (20M)

A. Perceptron with 2 input features - compute weighted sum, output, updated weights using perceptron learning algorithm (6M)
B. Perceptron Linear separability and its limitations (4M)
C. Generalization and Overfitting reasoning (5M)
D. Python code for perceptron training. For the given snippet, fill in the blanks with missing code/expression. (5M)

Q2 (20M)

A. Linear Regression for the house price prediction problem. Perform batch gradient descent iteration. Compute the predictions, calculate MSE loss, compute the gradient, updated weights. (6M)
B. Python code for batch gradient descent. For the given snippet, fill in the blanks with missing code/expression. (5M)
C. Linear regression for annual salary prediction (3M). RMSE reasoning. (3M)
D. Two linear regression models comparison on MSE reasoning (3M)

Q3. (20M)

A. Binary Classifier - compute weighted summation, predict probability using sigmoid, compute gradient, updated weights (6M)
B. Python code for the mini-batch logistic regression. (5M)
C. Logistic regression - using TP, FN, FP, TN - Compute the overall Accuracy (3M), Explain Recall vs Precision impact (3M)
D. Two model evaluation - interpretation based (3M)

Q4. (20M)

A. Three-class classifier - Compute softmax probability(2M), Categorical cross-entropy (2M), Identify model prediction(1M), Model prediction is correct or not (1M)
B. Python code for mini-batch softmax training code. (5M)
C. 3 Class classifier - compute precision and recall using TP, FP, FN, TN from confusion matrix (3M), Accuracy reasoning (3M)
D. Two models comparison and reasoning (3M)

Q5. (20M)

A. Perform forward propagation using a 2-layer DFFN using ReLU, Binary Cross-Entropy loss (6M)
B. Python code for 2-layer DFFN (5M)
C. Design a Deep Feedforward Neural Network for sentiment classification (6M)
D. Compute the Deep Feedforward Neural Network architecture parameters (3M)