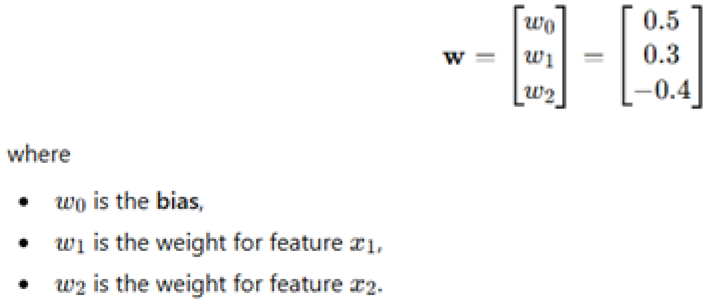
Q1 - Perceptron & Spam Classification

(a) Consider a perceptron with 2 input features. Current weights are:



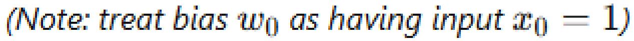
$$\mathbf{w} = \begin{bmatrix} w_0 \\ w_1 \\ w_2 \end{bmatrix} = \begin{bmatrix} 0.5 \\ 0.3 \\ -0.4 \end{bmatrix}$$

where

- w_0 is the **bias**,
- w_1 is the weight for feature x_1,
- w_2 is the weight for feature x_2.

A training example has features x1 = 1, x2 = 2 with true label y = 1. Learning rate η = 0.1.
The step activation function outputs 1 if z ≥ 0, otherwise outputs 0.

Compute: (i) weighted sum z, (ii) predicted output ŷ using step activation, (iii) if the example is misclassified, calculate the updated weights using the perceptron learning rule for each weight



(Note: treat bias w_0 as having input $x_0 = 1$)

[6 Marks]

(b) A spam classifier perceptron has learned weights w = [0.2, 0.8, 0.9, -0.5]^T (bias, suspicious words, links, length).

i. Which feature most strongly indicates spam? Explain using weight magnitudes. **[2 Marks]**

ii. The model achieves 75% accuracy but stops improving with more training. Explain what this reveals about the data's linear separability and perceptron limitations. **[4 Marks]**

Q1 - Perceptron & Spam Classification

(c) A company is building a perceptron-based classifier to categorize customer reviews as positive or negative. They have collected 500 training reviews and are comparing two approaches:

Approach A: Uses 5 carefully selected features:
count of positive sentiment words,
count of negative sentiment words,
presence of exclamation marks,
review length (word count),
and …..

<<<incomplete question + 2 sub part missing ☹ >>>

Q2 - Linear Regression & Gradient Descent

(a) A linear regression model predicts house price (in 1000s) from area (in 100s of sq ft). The Training data is :

Bias	Area	Price
1	10	150
1	20	250

Current weights: bias w0 = 50, area weight w1 = 8, learning rate η = 0.01.

Perform one batch gradient descent iteration:

(i) compute predictions for both examples, (ii) calculate MSE loss, (iii) compute gradient, (iv) update weights.

[6 Marks]

(b) Complete the code for batch gradient descent. For the following Python code, write only the code/expression that should replace each blank (Blank 1, Blank 2, etc.).
Do not write the entire program.

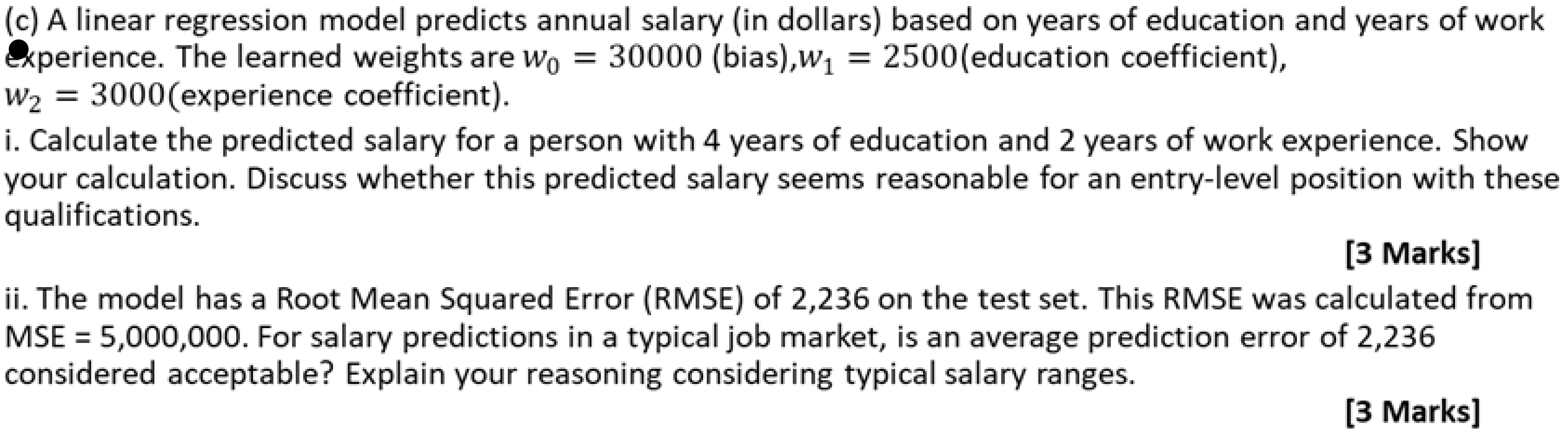
[5 Marks]

Q2 - Linear Regression & Gradient Descent

(b) .. contd…

```
import numpy as np
def sigmoid(z):
    return _____________    # Blank 1
def train_logistic(X, y, batch_size=32, lr=0.01, epochs=100):
    """X: (N x d), y: (N,) binary labels"""
    N, d = X.shape
    weights = np.zeros(d)
    for epoch in range(epochs):
        indices = np.random.permutation(N)
        for i in range(0, N, batch_size):
            idx = indices[i:i+batch_size]
            X_batch, y_batch = X[idx], y[idx]

                    <<<missing lines>>>
X = np.array([[1, 1], [1, 2], [1, 3]])
y = np.array([2, 4, 5])
w = _____________    # Blank 5
```

Q2 - Salary Prediction & Feature Scaling



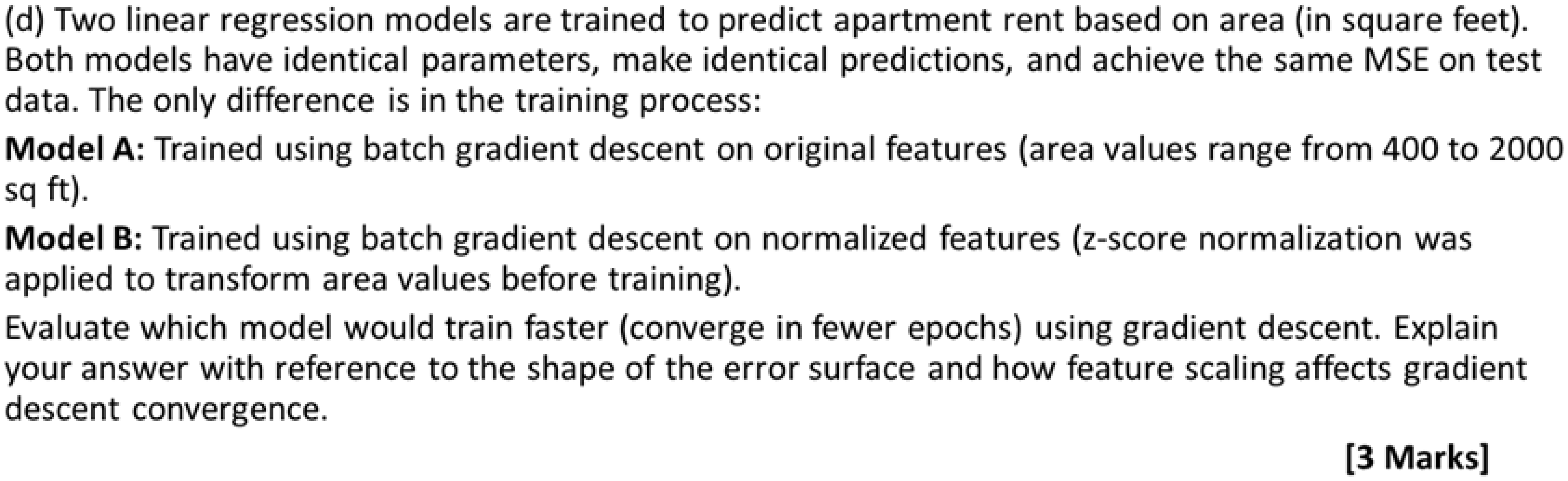
(c) A linear regression model predicts annual salary (in dollars) based on years of education and years of work experience. The learned weights are $w_0 = 30000$ (bias),$w_1 = 2500$(education coefficient), $w_2 = 3000$(experience coefficient).

i. Calculate the predicted salary for a person with 4 years of education and 2 years of work experience. Show your calculation. Discuss whether this predicted salary seems reasonable for an entry-level position with these qualifications.

[3 Marks]

ii. The model has a Root Mean Squared Error (RMSE) of 2,236 on the test set. This RMSE was calculated from MSE = 5,000,000. For salary predictions in a typical job market, is an average prediction error of 2,236 considered acceptable? Explain your reasoning considering typical salary ranges.

[3 Marks]



(d) Two linear regression models are trained to predict apartment rent based on area (in square feet). Both models have identical parameters, make identical predictions, and achieve the same MSE on test data. The only difference is in the training process:

Model A: Trained using batch gradient descent on original features (area values range from 400 to 2000 sq ft).

Model B: Trained using batch gradient descent on normalized features (z-score normalization was applied to transform area values before training).

Evaluate which model would train faster (converge in fewer epochs) using gradient descent. Explain your answer with reference to the shape of the error surface and how feature scaling affects gradient descent convergence.

[3 Marks]

Q3 - Logistic Regression & Evaluation

- (a) A binary classifier for loan approval uses logistic regression with sigmoid activation. The current weights are:

w0 = 0 (bias), w1 = 0.6 (credit score weight), w2 = 0.8 (income weight). The Learning rate η = 0.1.

Process one training example with the following values: credit score = 0.7, income = 0.5, and true label approve = 1

Compute: (i) weighted sum z, (ii) predicted probability using the sigmoid activation function, (iii) The gradient for this single training example where $y = 1$is the true label, (iv) The updated weights using gradient descent

[6 Marks]

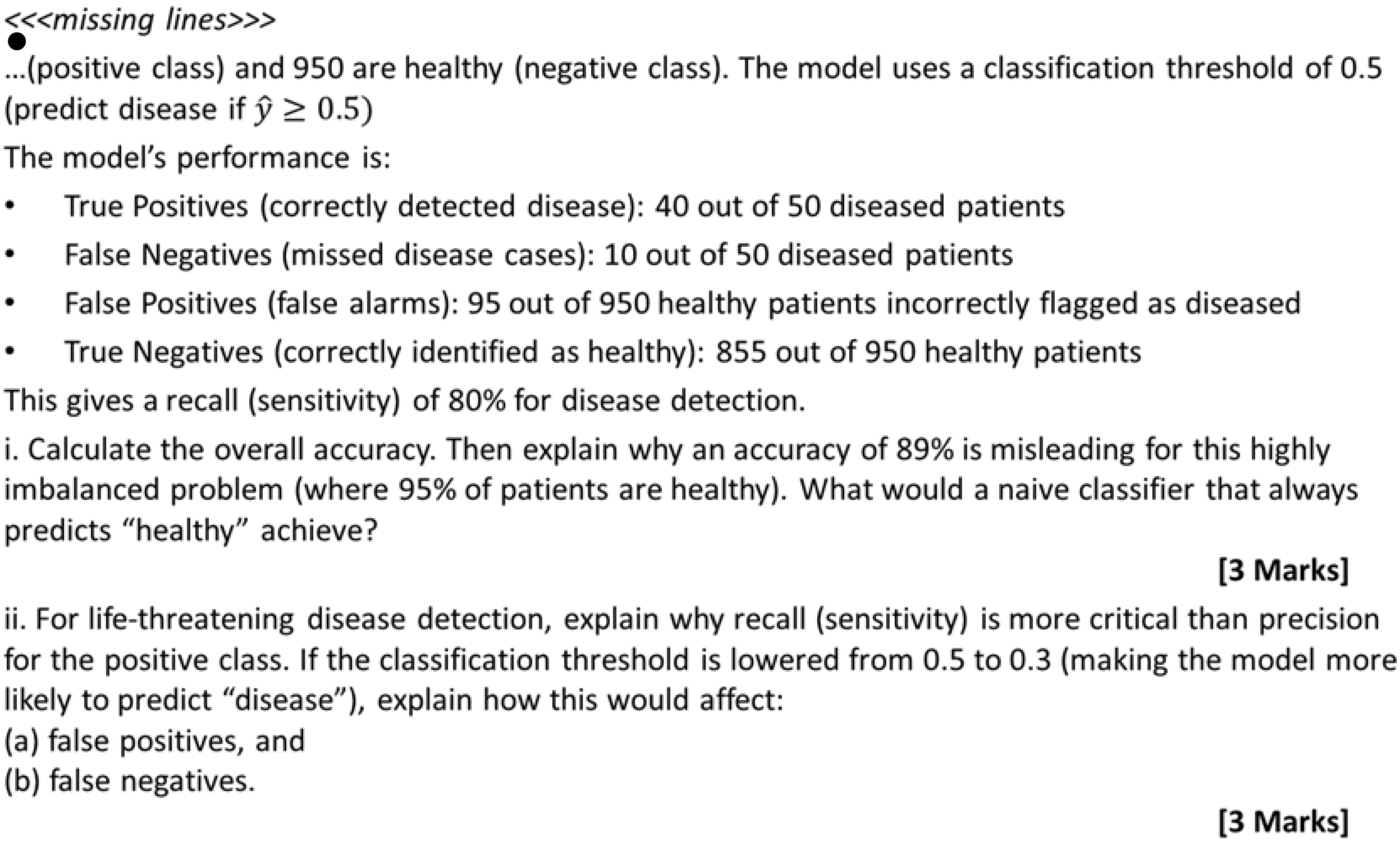
(b) Complete the mini-batch logistic regression code. For the following Python code, write only the code/expression that should replace each blank (Blank 1, Blank 2, etc.). Do not write the entire program.

[5 Marks]

Q3 - Logistic Regression & Evaluation

```
import numpy as np

def sigmoid(z):
    return _____________    # Blank 1

def train_logistic(X, y, batch_size=32, lr=0.01, epochs=100):
    """X: (N x d), y: (N,) binary labels"""
    N, d = X.shape
    weights = np.zeros(d)

    for epoch in range(epochs):
        indices = np.random.permutation(N)
        for i in range(0, N, batch_size):
            idx = indices[i:i+batch_size]
            X_batch, y_batch = X[idx], y[idx]

                <<<missing lines>>>
```

Q3 - Logistic Regression & Evaluation



<<<missing lines>>>

…(positive class) and 950 are healthy (negative class). The model uses a classification threshold of 0.5 (predict disease if $\hat{y} \geq 0.5$)

The model's performance is:

- True Positives (correctly detected disease): 40 out of 50 diseased patients
- False Negatives (missed disease cases): 10 out of 50 diseased patients
- False Positives (false alarms): 95 out of 950 healthy patients incorrectly flagged as diseased
- True Negatives (correctly identified as healthy): 855 out of 950 healthy patients

This gives a recall (sensitivity) of 80% for disease detection.

i. Calculate the overall accuracy. Then explain why an accuracy of 89% is misleading for this highly imbalanced problem (where 95% of patients are healthy). What would a naive classifier that always predicts "healthy" achieve?

[3 Marks]

ii. For life-threatening disease detection, explain why recall (sensitivity) is more critical than precision for the positive class. If the classification threshold is lowered from 0.5 to 0.3 (making the model more likely to predict "disease"), explain how this would affect:
(a) false positives, and
(b) false negatives.

[3 Marks]

Q3 - Logistic Regression & Evaluation

(d) A credit card company is deploying a fraud detection system using logistic regression. Their transaction dataset contains 10,000 transactions: 9,500 are legitimate and 500 are fraudulent. Two models have been trained and tested:

Model A: Uses 3 basic features (transaction amount, location, time of day). Achieves 95% overall accuracy and detects 60% of fraudulent transactions (300 out of 500 frauds caught).

Model B: Uses 20 features (including merchant category, device ID, purchase history patterns, etc.). Achieves 96% overall accuracy and detects 75% of fraudulent transactions (375 out of 500 frauds caught).

The company's cost analysis shows: Each missed fraud costs 100 in losses, and each …

<<<missing lines>>>

Q4 - Softmax, Confusion Matrix & Deep Learning

(a) A 3-class sentiment classifier uses softmax regression to categorize text reviews. The three sentiment classes are: negative (class 0), neutral (class 1) and positive (class 2). For a particular review, the model produces the following logits (pre-activation values): The true sentiment for this review is "neutral" (class 1), represented as the one-hot encoded vector:

$$\mathbf{z} = \begin{bmatrix} 2.0 \\ 1.0 \\ 0.5 \end{bmatrix} \quad \mathbf{y} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

Calculate:

i. The softmax probabilities for all three classes. Show your calculation for at least one class, including the sum in the denominator.

[2 Marks]

ii. The Categorical Cross-Entropy (CCE) loss:

$$L = -\sum_{k=0}^{2} y_k \log(\hat{y}_k)$$

[2 Marks]

iii. Identify which class the model predicts.

[1 Mark]

iv. State whether the model's prediction is correct by comparing the predicted class with the true label.

[1 Mark]

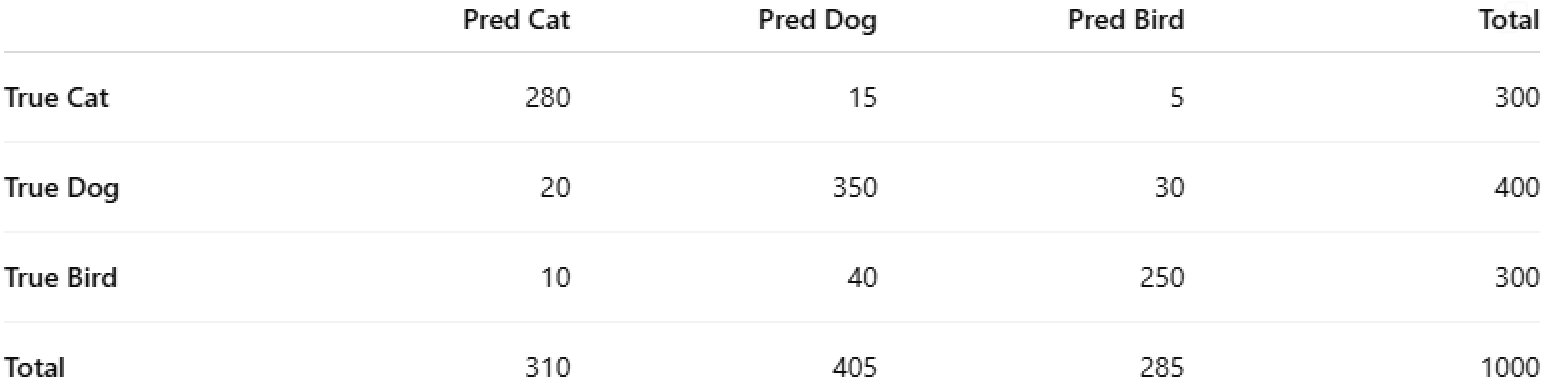
Q4 - Softmax, Confusion Matrix & Deep Learning

(b) Complete the mini-batch logistic regression code. For the following Python code, write only the code/expression that should replace each blank (Blank 1, Blank 2, etc.).
Do not write the entire program.

[5 Marks]

<<<missing code lines>>>

(c) An image classifier for 3 animal types (cat, dog, bird) is trained and evaluated.The confusion matrix on the test set is:



	Pred Cat	Pred Dog	Pred Bird	Total
True Cat	280	15	5	300
True Dog	20	350	30	400
True Bird	10	40	250	300
Total	310	405	285	1000

Q4 - Softmax, Confusion Matrix & Deep Learning

(c)

i. Calculate precision and recall for the "Bird" class. Clearly identify the values of True Positives (TP), False Positives (FP), and False Negatives (FN) from the confusion matrix

[3 Marks]

ii. Most bird misclassifications are predicted as "Dog" (40 cases out of 50 total bird errors). What does this confusion pattern suggest about the features the model has learned? Does the 88% overall accuracy guarantee good performance for each individual class?

[3 Marks]

(d) A hospital is developing a chest X-ray classifier to diagnose 5 conditions: pneumonia (8000 images), tuberculosis (150 images), COVID-19 (200 images), lung cancer (180 images), and healthy (7000 images). The dataset is severely imbalanced. Two trained models are being compared for deployment:

Model A: Achieves 92% overall accuracy across all 5 classes, but only 45% recall for tuberculosis detection (misses more than half of TB cases)

Model B: Achieves 85% overall accuracy, but has 78% recall for tuberculosis detection (catches most TB cases)

Evaluate which model should be deployed in the hospital. In your evaluation, consider: the clinical consequences of missing tuberculosis cases (a serious and contagious disease), whether the 7% drop in overall accuracy is acceptable given the improved TB detection, and...

<<<missing lines>>>

Q5 - Deep FeedForward Neural Network

(a) Perform forward propagation through a 2-layer Deep Feedforward Neural Network for binary classification.

Network Architecture:

- Input layer: 2 features
- Hidden layer: 2 neurons with ReLU activation
- Output layer: 1 neuron with sigmoid activation

Network Parameters: $W^{[1]} = \begin{bmatrix} 1.0 & 0.5 \\ -0.5 & 1.5 \end{bmatrix}, \quad b^{[1]} = \begin{bmatrix} 0.2 \\ -0.3 \end{bmatrix}, \quad W^{[2]} = \begin{bmatrix} 1.2 \\ 0.8 \end{bmatrix}^T, \quad b^{[2]} = 0.1$

Input and Label: Input vector: $\mathbf{x} = \begin{bmatrix} 0.5 \\ 0.8 \end{bmatrix}$, True label: y=1

Calculate:(i) Pre-activation values for hidden layer

(ii) Hidden layer activations after applying ReLU

(iii) Output prediction

(iv) Binary Cross-Entropy loss where y=1

[6 Marks]

Q5 - Deep FeedForward Neural Network

(b) Complete the 2-layer DFNN code. For the following Python code, write only the code/expression that should replace each blank (Blank 1, Blank 2, etc.). Do not write the entire program. **[5 Marks]**

<<<missing code lines>>>

(c) Design a Deep Feedforward Neural Network for sentiment classification with the following specifications: Input features: 1000 (word counts from text)

Output classes: 5 (sentiment categories: very negative, negative, neutral, positive, very positive)

Training data: 50,000 samples

Design and justify your network architecture by addressing the following: **(6 marks)**

(i) Specify the complete architecture: How many neurons in the input layer? How many hidden layers will you use and how many neurons in each hidden layer? How many neurons in the output layer?

(ii) Select appropriate activation functions: What activation function will you use for each layer? Justify your choices based on the classification task.

(iii) Choose the loss function: What loss function is appropriate for this problem? Explain why this loss function is suitable.

(iv) Specify the learning algorithm: Will you use batch gradient descent, stochastic gradient descent (SGD), or mini-batch gradient descent? Justify your choice.

(v) Calculate total parameters: Using your proposed architecture, calculate the total number of trainable parameters. Given 50,000 training samples, does your parameter count seem reasonable to avoid overfitting? Explain.

Q5 - Deep FeedForward Neural Network

(d) A company must select a Deep Feedforward Neural Network architecture for production deployment. The application is image classification on mobile devices with 10 output classes. The system must process 1 million images per day on smartphones with limited computational resources. Two architectures have been trained and evaluated:

Architecture A:

$$[256 \to 128 \to 64 \to 10]$$

(3 hidden layers, gradually decreasing width)

<<<missing lines>>>