Problem 1. *Perceptron by Hand*

Consider a perceptron with the following parameters:

$$\begin{aligned} \textit{Initial weights:} \quad & w_1 = 0.5, \quad w_2 = -0.3 \\ \textit{Bias:} \quad & b = 0.1 \\ \textit{Learning rate } (\eta): \quad & 0.1 \end{aligned}$$

Training data:

$$\begin{aligned} \textit{Data point} \quad & x = (1, 2) \\ \textit{True label} \quad & y = +1 \end{aligned}$$

Question A: Forward Propagation

Compute the weighted sum and the output after activation:

$$\textit{Weighted Sum} = w_1 \cdot x_1 + w_2 \cdot x_2 + b =$$

$$\textit{Prediction} = \frac{1}{1 + e^{-\textit{Weighted Sum}}} =$$

Question B: Calculate the Error

Compute the error between the predicted value and the true label:

$$\textit{Error} = y - \textit{Prediction} =$$

Question C: Backward Propagation

Calculate the gradient of the loss with respect to the weighted sum:

$$\frac{\partial \textit{Loss}}{\partial \textit{Weighted Sum}} =$$

Question D: Update Weights and Bias

Update the weights and bias using the computed gradients and the learning rate:

$$\begin{aligned} \Delta w_1 &= \eta \cdot \frac{\partial \textit{Loss}}{\partial \textit{Weighted Sum}} \cdot x_1 \\ &= \\ &\approx \\ \Delta w_2 &= \eta \cdot \frac{\partial \textit{Loss}}{\partial \textit{Weighted Sum}} \cdot x_2 \\ &= \\ &\approx \\ \Delta b &= \eta \cdot \frac{\partial \textit{Loss}}{\partial \textit{Weighted Sum}} \\ &= \\ &\approx \end{aligned}$$

Update weights and bias:

$$w_1 =$$
$$w_2 =$$
$$b =$$

Question E: Iteration

Repeat Questions A to D one more iteration, adjusting weights and biases to minimize the error and improve model accuracy.

Problem 2. *For $1 \leq i \leq N$, let $x_i \in \mathbb{R}^2$ be a two-dimensional input, and $y_i \in \{-1, +1\}$ be the binary label for x_i.*

Question A: Guaranteed Convergence

Describe, both in words and with a concrete example, when the perceptron algorithm would be guaranteed to converge (find an optimal solution to the training set). For the concrete example, give $N \geq 5$ data points $\{(x_i, y_i)\}$ for which the perceptron algorithm would converge.

Question B: Non-Guaranteed Convergence

Describe, in words and with a concrete example, when the perceptron algorithm would not be guaranteed to converge. For the concrete example, give $N \geq 5$ data points $\{(x_i, y_i)\}$ for which the perceptron algorithm would not converge.

Problem 3. *Classification: Implementation, Experimentation, and discussion*

In the upcoming questions, you will bebuilding, experimenting with, reporting on the performance of, and analyzing a multiclass perceptron classifier. The essential requirements for these questions encompass the following:

1. *Provide any necessary implementations, scripts, and serialized model files required to successfully compile and run your code.*

2. *Deliver a comprehensive written report that addresses the following aspects:*

   - *Your implementations, outlining design decisions made and any encountered difficulties.*
   - *Present evidence and thorough analysis of internal development and experimentation carried out on the perceptron model using the development set.*

   The MNIST dataset is a widely used benchmark dataset in the field of machine learning and computer vision. It comprises a collection of grayscale images, each representing handwritten digits from 0 to 9. With a total of 60,000 training images and 10,000 testing images, MNIST serves as a fundamental resource for developing and evaluating algorithms in tasks such as image classification and pattern recognition. Its simplicity and standardized format have made it a foundational dataset for researchers and practitioners to explore and benchmark various machine learning techniques, making it a cornerstone in the development of image-based models.

The MNIST dataset is a widely used benchmark dataset in machine learning and computer vision. It consists of a collection of grayscale images representing handwritten digits from 0 to 9. The dataset is available through various sources:

(a) ***Official Website****: The official MNIST dataset website provides access to the dataset and additional information:* `http://yann.lecun.com/exdb/mnist/`

(b) ***TensorFlow Datasets****: If you're using TensorFlow, you can load the MNIST dataset using TensorFlow Datasets:* `https://www.tensorflow.org/datasets/catalog/mnist`

(c) ***Scikit-learn****: The scikit-learn library offers an interface to load the MNIST dataset:* `https://scikit-learn.org/stable/datasets/toy_dataset.html#mnist-dataset`

(d) ***Kaggle****: Kaggle provides the MNIST dataset:* `https://www.kaggle.com/c/digit-recognizer/data`

The task is to predict the digit $y \in \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$ *from a* 28×28 *input grayscale images.*

Question A: Base Model

Implement a baseline classifier called the "most frequent" baseline. This baseline strategy identifies the label that appears most commonly in the training data (train) and consistently predicts that label during evaluation, regardless of the input.

Steps:

1. *Implementation: In your report, document the implementation of this baseline classifier. The implementation of this baseline classifier should be fairly straightforward.*
2. *Performance Report: In your report, provide a detailed report on how well this baseline classifier performs on the test dataset. Analyze and discuss the results, considering its accuracy and any insights gained from its performance as a baseline.*

Question B: Multi-layer Perceptron

Train a perceptron on the train dataset and evaluate its performance on the test dataset. In tackling this question, you're allowed to use computation accelerators such as BLAS, NumPy, or MATLAB. However, it's important to note that you should refrain from using any pre-existing perceptron implementations. Your goal is to implement the perceptron algorithm from scratch as part of this question.

Steps:

1. *In your report, briefly discuss your implementation approach and the criteria you used for determining convergence.*

2. *In your report, explain the specific and measurable methods you employed to validate the correctness of your implementation.*

3. *In your report, conduct experiments using at least three distinct configurations. For each configuration, train on train and evaluate on test. Illustrate your internal development process using easily understandable graphs or tables, showcasing how different model setups perform on test. Consider varying factors like learning biases, convergence criteria, or input feature choices when creating these configurations.*