1131 Deep Learning – Homework 2

Due: 11/19/2024, 11:59 pm

Please submit your code and report. (Total: 100%)

1. (20%) Implement the Perceptron algorithm to classify the dataset in "train.mat". Plot the classification results, including the decision boundary created by the perceptron model.

2. (21%) Construct and train three neural networks with 3, 10, and 50 hidden layers, respectively. Each hidden layer should contain five neurons and use the ReLU activation function. Train each network on "train.mat" and evaluate them on the test dataset "test.mat". Report the test error (percentage of misclassified samples) for each neural network.

3. (9%) If you replace the ReLU activation function with the Sigmoid activation function for all neurons, how does this affect the performance of the three neural networks? Report and discuss your observations for each model.

4. (50%) The MNIST dataset consists of images of handwritten digits, with a training set of 60,000 examples and a test set of 10,000 examples. The dataset can be downloaded from http://yann.lecun.com/exdb/mnist/.

   4.1 (15%) Construct a neural network-based classification model for digit recognition and train it on the MNIST training set. Report the prediction accuracy on the test set.

   4.2 (15%) Introduce 10% salt-and-pepper noise to the test images using the sample code below and evaluate them using the model trained on clean images from Question 4.1. Report the prediction accuracy on the noisy test set. Compare these results with those from Question 4.1 and discuss your observations.

   To add salt-and-pepper noise to the images, you can use the following code snippet:

```
import random
import numpy as np

# Assuming X_test is the test set of images with shape (num_samples, 28, 28)
noise_lv = 0.1
img_size = 28 * 28
for i in range(len(X_test)):
    ran_seq = random.sample(range(img_size), int(img_size * noise_lv))
    x = X_test[i].reshape(-1, img_size)
    x[0, ran_seq] = 255  # Add salt noise (white pixels)
```

4.3 (20%) Add 10% salt-and-pepper noise to the training set and re-train the model on the noisy data. Then, evaluate the model on the noisy test images from Question 4.2. Report the prediction accuracy and compare these results with those from Question 4.2. Discuss any differences and insights.