1112 Deep Learning – Homework 1

Due: 10/27, 2024, 11:59 pm

For the following tasks, please upload the source code to Moodle and include a detailed explanation of your results in the report.

1. (**25%**) Please load 'data.mat' into your Python code, where you will find $x, y \in R^{1001}$. Now, do the following procedures.

   1.1. (**5%**) Use the plot function to visualize the data.
   1.2. (**5%**) Compute the least square line $y = \boldsymbol{\theta_0} + x\boldsymbol{\theta_1}$ using the given data and overlay the line on the given data.
   1.3. (**5%**) Fit a least squares second-order polynomial ($y = \boldsymbol{\theta_0} + x\boldsymbol{\theta_1} + x^2\boldsymbol{\theta_2}$) to the data.
   1.4. (**5%**) Fit a least squares quartic curve ($y = \boldsymbol{\theta_0} + x\boldsymbol{\theta_1} + x^2\boldsymbol{\theta_2} + x^3\theta_3 + x^4\theta_4$) to the data.
   1.5. (**5%**) Analyze which model (line, parabola, or quartic curve) is most appropriate for this dataset. Justify your answer by calculating and comparing the mean squared error (MSE) for each fitting model.

2. (**20%**) Following the previous questions, please randomly select 30 data samples and repeat this process 200 times. Plot the resulting 200 linear regression lines ($y = \boldsymbol{\theta_0} + x\boldsymbol{\theta_1}$) and 200 quartic curves ($y = \boldsymbol{\theta_0} + x\boldsymbol{\theta_1} + x^2\boldsymbol{\theta_2} + x^3\theta_3 + x^4\theta_4$**)** in two separate figures: one for lines and one for quartic curves. In your report, explain the visualizations in the context of bias and variance, discussing how the spread and behavior of the curves or lines relate to the concepts of underfitting, overfitting, and model complexity.

3. (**20%**) In 'train.mat,' you will find 2-D points X=[x1, x2] and their corresponding labels Y=y. Please use logistic regression $h(\boldsymbol{\theta}) = \frac{1}{1+e^{-\boldsymbol{\theta}^T x}}$ to find the **decision boundary** (optimal $\boldsymbol{\theta}^*$) based on 'train.mat."

   3.1. Plot the 2-D data points and the decision boundary on the same graph to visually illustrate the model's separation of classes.

   3.2 Evaluate the model on the test dataset ('test.mat') and report the test error, defined as the percentage of misclassified test samples.

4. Download the MNIST dataset using the following example code:

```
################################################
from __future__ import print_function
import keras
from keras.datasets import mnist

# input image dimensions 28×28
img_rows, img_cols = 28, 28

# the data, split between train and test sets
(x_train, y_train), (x_test, y_test) = mnist.load_data()


x_train = x_train.astype('float32')
x_test = x_test.astype('float32')
x_train /= 255
x_test /= 255
################################################
```

Please randomly choose 5,000 handwritten images from either the training or the testing dataset to construct your own dataset, with 500 data samples for each digit.

4.1. (**5%**) Use the following code to show 50 images in your own dataset.

```
################################################
import numpy as np
import matplotlib.pyplot as plt
amount= 50
lines = 5
columns = 10
number = np.zeros(amount)

for i in range(amount):
    number[i] = y_test[i]
    # print(number[0])

fig = plt.figure()

for i in range(amount):
    ax = fig.add_subplot(lines, columns, 1 + i)
    plt.imshow(x_test[i,:,:], cmap='binary')
    plt.sca(ax)
    ax.set_xticks([], [])
    ax.set_yticks([], [])

plt.show()
################################################
```

4.2. (**15%**) Apply PCA (Principal Component Analysis) to reduce the 784-dimensional data to 500, 300, 100, and 50 dimensions. For each reduction, show ten decoded results for each digit and analyze how the data reconstruction changes with decreasing dimensions.

In your report, interpret the results by discussing how the dimensionality reduction affects the quality of the decoded images and explain any observed trade-offs between dimensionality and image clarity.

4.3. (**15%**) Use PCA to project the MNIST dataset down to 2D and sample at least 4 decoded images from different regions across the four quadrants of this 2D projection.