1112 Deep Learning – Homework 1

Due: 10/20, 2023, 11:59pm

For the following questions, please upload the source code to moodle and explain the results in your report. **If you choose to implement the machine learning models by yourself (no built-in APIs), you will get extra 10% bonus for each question.**

1. 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%**) Plot the data using plot function.
   1.2. (**5%**) Compute the least square line $y = \boldsymbol{\theta_0} + x\boldsymbol{\theta_1}$ using the given data and overlay the line over the given data.
   1.3. (**5%**) Compute the least square parabola (i.e. second order polynomial $y = \boldsymbol{\theta_0} + x\boldsymbol{\theta_1} + x^2\boldsymbol{\theta_2}$) to fit the data.
   1.4. (**5%**) Compute the least square quartic curve ($y = \boldsymbol{\theta_0} + x\boldsymbol{\theta_1} + x^2\boldsymbol{\theta_2} + x^3\theta_3 + x^4\theta_4$) to fit the data.
   1.5. (**5%**) Explain which formulation (line, parabola, cubic curve) is more suitable for this dataset and why (please calculate the mean square error for these two fitting equations)?

2. (**25%**) Following the previous two questions, please randomly select 30 data samples for 200 times and plot these 200 lines ($y = \boldsymbol{\theta_0} + x\boldsymbol{\theta_1}$) and 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 the other for quartic curves. Explain these visualizations based on the bias and variance.

3. (**15%**) In 'train.mat,' you can 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." Please report the test error on the test dataset 'test.mat.' (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 28x28
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 different handwritten images from either the training or the testing dataset to construct your own dataset, where each digit has 500 data samples.

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%**) Normalize the data (subtracting the mean from it and then dividing it by the standard deviation) and compute the eigenpairs for the covariance of the data (sorted in a descending order based on eigenvalues).

4.3. (**15%**) Please use PCA to reduce the 784 dimensional data to that with 500, 300, 100, and 50 dimensions, and then show 10 decoding results for each digit, respectively. How do you interpret these results?