1072 Deep Learning – Homework 1

Due: October 24, 2019, 11:55pm

1. (20%) (Maximum Likelihood Estimation) Please set the derivatives of the log likelihood function of a Gaussian Probability Distribution Function $\log p(x; u, \sigma^2) = -\frac{1}{2\sigma^2}\sum_{i=1}^{N}(x_i - u)^2 - \frac{N}{2} log\sigma^2 - \frac{N}{2}\log(2\pi)$ to zero with respect to u and σ and verify the following results below:

$$u_{ML} = \underset{u}{\operatorname{argmin}} - \sum_{i=1}^{N} \log p(x; u, \sigma^2) = \frac{1}{N}\sum_{i=1}^{N} x_i$$

$$\sigma_{ML} = \underset{\sigma}{\operatorname{argmin}} - \sum_{i=1}^{N} \log p(x; u, \sigma^2) = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - u_{ML})^2}$$

2. (20%) Please load 'data.mat' into your Matlab or Python code, where you will find $x, y \in R^{1001}$. Now do the following procedures, **paste your source code and show the results in your report.**
   2.1. Plot the data using plot function.

   >> plot(x, y); grid

   2.2. 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.
   >> hold on; plot(x, $\boldsymbol{\theta_0}$+$\boldsymbol{\theta_1}$*x, '--')

3. (20%) Using the same data from Question 2, 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. (5%) Explain which formulation (line or parabola) is more suitable for this dataset and why? (**paste your source code and show the results in your report**)

4. (40%) 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
################################################
```

For the following questions, please upload the source code to moodle and show the results in your report.

4.1. (5%) Please draw 1,000 different handwritten images from either the training or the testing dataset to form your own dataset, where each digit has 100 data samples.

4.2. (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()
################################################
```

4.3. (5%) Normalize the data (subtracting the mean from it and then dividing it by the standard deviation) and compute the covariance matrix for the data.

4.4. (10%) Compute eigenpairs for the covariance (sorted in a descending order based on eigenvalues).

4.5. (13%) Please use PCA to reduce the 784 dimensional data to that with 500, 300, 100, and 50 dimensions, and then show the decoding results, respectively. How do you interpret the results?