

7CCSMPNN Pattern Recognition, Neural Networks, and Deep Learning

Coursework Assignment 1

This coursework is assessed. You need to submit a type-written report (in pdf format) with your answers to parts 1 to 3. Only include in your report the information specifically requested below: *this is indicated by underlined italicised text*. In addition you need to submit a .h5 file with your solution for part 4. Both files need to be submitted online through KEATS by the deadline specified on the module's KEATS webpage.

Part 0: Creating Datasets

Training Datasets

Some of the exercises in this coursework require you to use the Iris dataset. This dataset contains 150 samples from 3 classes. Each sample is a four-dimensional feature vector. This data can be obtained as follows.

In MATLAB:

Download the file iris_class1_2_3_4D.mat from the module's KEATS webpage. Load this dataset into MATLAB using the following command:

```
load('iris_class1_2_3_4D.mat')
```

This will produce two variables, X and t. Each column of X is a feature vector for one sample. The class label associated with each sample is given by the corresponding element of vector t.

In python:

Use the following commands:

```
from sklearn import datasets  
iris = datasets.load_iris()
```

The structure "iris" contains a field `iris.data` that is an array, each row of which is a feature vector for one sample, and a field `iris.target` with is a column vector defining the class labels associated with the corresponding rows of `iris.data`.

Personalised Test Data

To answer some of the questions in this coursework you will need to create a test dataset that is unique to you. As the answers you get will depend on the test set you use, you must ensure that you generate the test set correctly, otherwise your answers will be wrong!

To create this test set, Xtest, use your KCL student ID (this is the number that appears on your College ID card, it is **NOT** the k-number you use to log-into College computers).

Define s1, s2, s3, s4, s5, s6, s7 to be the 1st to 7th digits of your KCL student ID number, then define Xtest using the following code.

In MATLAB:

```
Stest=[s1, s2, s3, s4, s5, s6, s7; s2, s3, s4, s5, s6, s7, s1; s3, s4,  
s5, s6, s7, s1, s2; s4, s5, s6, s7, s1, s2, s3];  
Stest=Stest./[2.3;4;1.5;4];  
Xtest=Stest+[4;2;1;0]
```

In python:

```
import numpy as np  
Stest=np.array([[s1, s2, s3, s4, s5, s6, s7],[s2, s3, s4, s5, s6, s7, s1],  
[s3, s4, s5, s6, s7, s1, s2],[s4, s5, s6, s7, s1, s2, s3]]);  
Stest=Stest/np.array([2.3,4,1.5,4]).reshape(-1,1);
```

```
Xtest=Stest+np.array([4,2,1,0]).reshape(-1,1)
print(Xtest)
```

Hence, if your KCL student ID was 1234567, Xtest would be:

4.4348	4.8696	5.3043	5.7391	6.1739	6.6087	7.0435
2.5000	2.7500	3.0000	3.2500	3.5000	3.7500	2.2500
3.0000	3.6667	4.3333	5.0000	5.6667	1.6667	2.3333
1.0000	1.2500	1.5000	1.7500	0.2500	0.5000	0.7500

Each column of Xtest is a sample taken from a 4-dimensional feature-space.

In your report give your KCL student ID, and report the values in the array Xtest.

Part 1: k-Nearest-Neighbour Classifier

This exercise requires you to use a k-Nearest Neighbours classifier with the Iris dataset (see Part 0) in order to classify the samples in Xtest (see Part 0). Use the **euclidean distance** between the values as your measure of similarity. You should perform kNN as described in the lectures and tutorials, on the data you have been given, not rescaled or otherwise modified data.

You can do this manually, but it is laborious. It is easier to write some simple code to do this for you. MATLAB provides inbuilt functions for kNN (see “fitcknn” or “knnclassify”), python (as part of the scikit-learn library) provides **KneighborsClassifier**, or you may wish to write your own code .

Determine the class of each of the 7 samples in Xtest using the k-nearest neighbours classifier (kNN), for $k=3$ and $k=7$. *In your report write down the class you determined for each sample in Xtest, for each value of k* [14 marks].

Part 2: Discriminant Functions

Answer tutorial question 14 in the section of the tutorial on Discriminant Functions, except use a **margin vector $b = [s_1, s_2, s_3, s_4, s_5, s_6]^T$** , where $s_1, s_2, s_3, s_4, s_5, s_6$ are the first six digits of your KCL student ID.

This question is about using the Sequential Widrow-Hoff Learning Algorithm to find a linear discriminant function to classify the data given in the table in question 12 of the same section of the tutorial. You can do this manually, but it is laborious and error-prone. It may be easier for you to write some simple code to do this for you.

In your report provide a table showing the results of the calculations performed at each iteration. particularly show the parameters of the discriminant function learnt at each iteration [12 marks].

Note, if you fail to **use your student ID to define margin vector b** , you will receive a mark of zero, even if the method you use is correct.

Part 3: Neural Networks

Answer tutorial question 6 in the section of the tutorial on Neural Networks, except use values of $\theta = -s_3$, $w_1 = -s_4$, and $w_2 = s_5$ as the initial parameter values, where s_3, s_4 , and s_5 , are the **third, fourth and fifth digits** of your KCL student ID. Continue to perform iterations of the algorithm until **convergence to a solution** or until **12 iterations** have been performed, whichever is earliest. For the heaviside function, H , define $H(0)$ as 0.5.

This question is about using the **Sequential Delta Learning Algorithm** to find the **weights of a Linear Threshold Unit** to classify the same data as was used in Part 2. You can do this manually, but it is laborious and error-prone. It will be easier for you to write some simple code to do this for you.

In your report provide a table showing the results of the calculations performed at each iteration, particularly show the parameters of the linear threshold unit learnt at each iteration [6 marks].

Note, if you fail to use your student ID to define the initial parameters, you will receive a mark of zero, even if the method you use is correct.

Use the Sequential Delta Learning Algorithm to learn the weights of a Linear Threshold Unit that should output 1 for class 0 and output 0 for the other two classes of the iris dataset (see Part 0). Apply the Sequential Delta Learning Algorithm for 2 epochs. Use a learning rate of 0.1 and use values of $\theta = -s_3$, $w_1 = -s_4$, $w_2 = s_5$, $w_3 = -s_6$, and $w_4 = s_7$ as the initial parameter values, where s_3, s_4, s_5, s_6 and s_7 are the third to seventh digits of your KCL student ID. For the heaviside function, H , define $H(0)$ as 0.5.

In your report write down the parameters learnt at the end of 2 epochs of training on the iris data [5 marks]. Use these parameters to calculate the output of the Linear Threshold Unit for each sample in the test set, X_{test} , defined as specified in Part 0. In your report write down the output you calculated for each sample in X_{test} [7 marks].

Part 4: Deep Discriminant Neural Networks

This exercise requires you to build and train a deep neural network for classifying the MNIST dataset. The MNIST dataset consists of 60000 28-by-28 pixel images of handwritten digits.

Your network must be built using KERAS. Please see the separate instructions that are available on KEATS for a tutorial on using KERAS. This tutorial describes how to build two networks for classifying the MNIST data: a MLP and a CNN. You can use either of these as a starting point and experiment with making modifications to improve performance. Or you may choose to build your own deep network.

Your neural network will be assessed by testing the accuracy with which it classifies unseen test data ("my testset"). This testing data is distinct from the testing and training data provided as part of the MNIST dataset, but like this data also consists of 28x28 pixel images of hand-written digits. These images have pixel values between 0 and 1, just like MNIST after undergoing the preprocessing described in the tutorial on using KERAS.

Your trained neural network must be saved as a .h5 file and submitted as a separate file to the report with your answers to Parts 1-3 [15 marks]. Note it should be possible to test your neural network to predict class labels on a standard PC running Linux without the need for special hardware, such as a GPU or RAM in excess of 20GB. Note also, that you are required to build and train your own neural network, anyone who is found to have submitted a model that they have not trained themselves (e.g. a pre-trained obtained from the internet) will receive a mark of zero.

Marks will be awarded based on how accurately your neural network performs the classification of my testset. Note that the tutorial on using KERAS explains how to build two networks for classifying the MNIST data: the MLP will classify the standard MNIST test set with an accuracy of about 98% and will classify my testset with an accuracy of about 95%; the CNN will classify the standard MNIST test set with an accuracy of about 99% and will classify my testset with an accuracy of about 94%. Hence, 95% accuracy is considered the baseline and will earn 25% of the available marks. Higher marks will be awarded for accuracies greater than 95%.