University College Dublin
An Coláiste Ollscoile, Baile Átha Cliath

SPRING TRIMESTER EXAMINATION 2021/2022

ACM30100

Mathematics of Machine Learning (Lab exam)

Professor E. Winstanley

Professor E. Cox

Dr Barry Wardell*

Time Allowed: 1.5 hours

Instructions

- Full marks will be awarded for complete and correct answers to **all** questions.
- The exam must be completed individually.
- You may use the following materials:
  1. Lecture notes, assignments and all materials on the Brightspace page for the module.
  2. Any other hand-written notes that you have prepared.
  3. Any textbooks, in either print or electronic form.
  4. You may use a computer or calculator.
  5. You may use Google Colab for running `Python` notebooks.
- Except where otherwise specified, you may choose to use either `Python` or `Mathematica`.
- With the exception of the above, no other sources are allowed. In particular, you may not use web search or other web pages. You also must not discuss the exam with anyone else.
- Once the exam has finished, upload your solutions to Brightspace. Your upload may consist of multiple files, if appropriate.

1. Neural Networks

Consider the code in the file "Neural Network Exam.ipynb". This is the similar to the code we studied previously in the examples repository.

(a) Create and train a neural network on the MNIST dataset in each of the following cases (in all cases use a softmax activation function on the output and a "sparse_categorical_crossentropy" cost function with "adam" optimizer; the input should be a $28 \times 28 \times 1$ array and the output an array of 10 probabilities):

(i) Three fully-connected hidden layers with 35, 30 and 25 neurons, sigmoid activation functions. Train for 2 epochs with batch size 20.

(ii) One hidden layer with 10 neurons, sigmoid activation function. Train for 20 epochs with batch size 10.

(iii) A three-hidden-layer network consisting of: 1. A convolution layer with 8 filters each with kernel of size 4×4, with `ReLU` activation function and with padding to keep the output the same size as the input; 2. A convolution layer with 12 filters each with kernel of size 4×4, with `ReLU` activation function and with padding to keep the output the same size as the input. Train the network for 6 epochs with batch size 64.

Hint: You may need to use Flatten layers in appropriate places.

(b) Briefly comment on which, if any, of your networks are showing signs of being overtrained and why.

(c) Determine which of the digits 0-9 in the test data are misclassified by each of your models:

(i) Most often?

(ii) Least often?

2. Support Vector Machines

Consider the data in the files `svm_plus.csv` and `svm_minus.csv`. This is a pair of three-dimensional datasets which should be linearly separable.

(a) Set up and solve the primal optimisation problem for a support vector machine with this dataset.

(b) Set up and solve the dual optimisation problem for a support vector machine with this dataset.

(c) Plot the data along with the decision hyperplane and margins. Indicate the support vectors on the plot.

(d) Create a function to evaluate the decision rule and use it to categorise the samples in `svm_test.csv` as either "plus" or "minus".

3. Linear Regression and Principal Component Analysis

Consider the data in the file "regression.csv". Each row i in the data file contains three columns of data: x_i, y_i, f_i .

(a) Obtain a least-squares best fit to the data of a model of the form

$$f(x, y) = a + b\,x + c\,y.$$

(b) Compute and plot the errors $e_i = |f(x_i, y_i) - f_i|$ between your model and the data. Compute the squared error $e = \sum_i e_i^2$.

(c) Compute a fit of the same model which minimises the **orthogonal distance** from the data to the best-fit line. Your fit should produce a model of the form

$$\hat{f}(x, y) = \alpha + \beta\,x + \gamma\,y.$$

(d) Compute and plot the errors $\epsilon_i = |\hat{f}(x_i, y_i) - f_i|$ between your orthogonal model and the data. Compute the squared error $\epsilon = \sum_i \epsilon_i^2$.

(e) Produce a plot of the data along with your two models.

—o0o—

Note: A CSV file can be imported into Mathematica using the command:

```
data = Import["file.csv"];
```

or into Python using the commands

```
import numpy as np
data = np.genfromtxt("file.csv", delimiter=",")
```

—o0o—