Machine Learning Summative Assessment

This course is assessed through a take-home exam. The overall aim of this assessment is to examine the students' understanding of modern machine learning techniques at the theoretical, intuitive and practical levels. With regard to practical applications we use Python, as well as related libraries such as Numpy, Scikit and TensorFlow.

Several questions are centred around the MNIST data set of hand-written digits. This dataset consists of grayscale images of hand-written digits, as well as the true labels/digits of each image. The data is split into train and test sets. Many problems relate to the classification task of predicting the true labels given the images. Please see the FAQs below on how to download the data set.

1) **General Knowledge.** [20 points]

   a) [5 points] Describe the bagging algorithm and explain how it improves the test error of a decision tree.

   b) [5 points] How does a random forest differ from bagging? What is the advantage of this difference?

   c) [10 points] In a study you are designing questionnaires to predict student drop out. You gathered data on 600 undergraduates at the University. Every first year student was given a questionnaire consisting of 1,000 questions, and four years later it was recorded whether each student had dropped out. Your advisor wants you to use the answers in the questionnaire to predict drop out. She did a crude analysis of all the data and asked you to develop a model using a specific subset of 20 questions which show a higher correlation with the response.
   You apply Logistic Regression using these 20 predictors. To estimate the test error, you split the students into a training and a validation set; you train the model on the first subset and compute the following confusion matrix on the validation set:

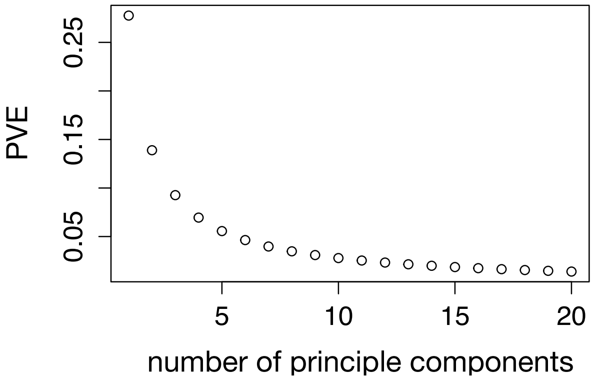
	True Positive	True Negative
Predicted Positive	70	10
Predicted Negative	6	200

   Is this good evidence that the questionnaire can predict college drop out? Explain why or why not.

2) **Unsupervised learning.** [20 points]

   a) **Practical.** [10 points] Perform a principle component analysis on the MNIST data set. Plot the Proportion of Variance Explained (PVE) against the number of principle components. Further, produce various plots in the dimensionally reduced space using the principle components.

b) **Theoretical Exercise.** [10 points] Consider a simulated dataset of 500 observations of 20 predictors $X_1, \ldots, X_{20}$ of are multivariate normally distributed, they are correlated and have variance 1 each. This is the output of a PCA analysis:



Two additional predictors X_{21}, X_{22} are simulated. Both are standard normal and uncorrelated with any other predictor. We now simulate a response using:

$$Y = 0.5 + 0.1 \cdot X_9 + 0.05 \cdot X_{13} + 1.7 \cdot X_{21} + 2.2 \cdot X_{22} + 0.1 \cdot \epsilon$$

with ϵ being standard normal. Consider regressions of Y on $X_1, \ldots, X_{22}$ using (i) a PCA regression on the top two principle components (ii) a Lasso regression using two non-zero coefficients. Which one will have the lower MSE? Explain your answer.

3) **Support vector machines.** [20 points]

a) **Theory.** [10 points] Consider a two-dimensional predictor (X, Y) and a support vector machine (SVM) with polynomial kernel of degree 2, $K(v_1, v_2) = (1 + v_1^T v_2)^2$, where $v_i = (x_i, y_i)$ and $v_1^T v_2 = x_1 x_2 + y_1 y_2$ is the inner product. Show that the decision boundary of the SVM is a conic section, i.e. an equation of the form $ax^2 + by^2 + cxy + dx + ey + f = 0$.

b) **Practical.** [10 points] Build a linear support vector classifier in Python for the MNIST classification task. Next, use various kernels to build a better non-linear classifier.

Explain the advantages and disadvantages of each kernel in comparison of the others.

Also comment on how each hyper-parameter in the kernel relates to model flexibility in the context of the bias-variance-tradeoff.

4) **Neural Networks.** [40 points]

a) **Theory.** [10 points] By applying the chain law of derivatives to the objective function of a neural network, derive the back propagation algorithm.

b) **Coding.** [10 points] Implement from scratch, as a Python class, a feedforward neural network with 1 hidden layer of 100 hidden units merely using basic linear algebra computations in Numpy (matrix/vector multiplications and additions).

Implement the back-propagation algorithm for stochastic or batch gradient descent. Test your implementation on the MNIST classification task and report your test accuracy.

c) **Practical.** [10 points] Use TensorFlow (potentially using a frontend such as Tflearn or Keras) to implement a 2 layer neural network with 1200 and 1200 hidden units. Use RELU activation function instead of sigmoids, and add regularisation methods such as dropout to produce a model which has a test error of below 2%. 10 points will be given for building a working network and 5 points for achieving the benchmark.

d) **Research Understanding Challenge.** [10 points] After having attended the course you should be able to pick up new machine learning research articles, understand the content and use it for your own work. In this last exercise you should read the research paper “Attention is all you need” (https://arxiv.org/pdf/1706.03762.pdf) and describe in your own words the model introduced there, its benefits and how it relates and differs to other models covered in this course, e.g. LSTMs.

The total number of points is 100.

The best projects will typically contain at least some of the following:

- Have clear explanations and mathematically sound derivations
- Have clear code with sufficient comments/documentation for it to be easily understood
- Be well-written and demonstrate a critical understanding of machine learning

Candidates should submit a zip or tar.gz file containing their written report in PDF format and their code in an executable format (e.g., .py Python scripts or .ipynb Jupter notebooks).

Students should submit a written report that should not exceed about 8-pages, and must not exceed 3,500 words as specified in the examination regulations. The report should include equations and figures, describing the solution to each of the above problems. Explanations as well as theoretical derivations should be included in report while code extracts should be referenced from the Python files.

The assessment is due 12 noon GMT of Wednesday of Week 10 of Michaelmas Term (Wednesday 16th December) and should be submitted electronically before this time.

The report should follow the normal OII formatting guidelines, and the submissions should be self-contained: examiners will not access external websites or other linked documents.

FAQs

- Can one of my approaches be in a language other than Python?

  - No. Especially when working with neural networks and TensorFlow, Python has become the de facto industry standard.

- Where can I download the MNIST dataset?

  - The MNIST dataset can be downloaded from http://yann.lecun.com/exdb/mnist/.

- I cannot find an answer to 2.b in my notes. The problem seems difficult.

  - The problem challenges your understanding of the methods involved, namely PCA and Lasso. You will have to derive it from first principles.

- In part 3.c I cannot get my test error below 2%. What can I do?

  - Check whether you are missing an ingredient, or maybe a different optimiser such as Adam will help. Review the research papers cited on http://yann.lecun.com/exdb/mnist/.