

## The Project

### Purpose and Scope

A project is an exploration of an issue or concept or phenomenon we have encountered in our study of ANNs. Crucially, the project is constrained by converting or interpreting the topic into a **specific, testable proposal** (often expressed as a hypothesis). Often the idea is tested in a very simplified or essential form. We are not trying to make functional ML applications for real world problems – projects might explore the mechanism, limitations, possibilities, and exploitable features of ANNs. Alternatively, ANNs might be applied to different problems (i.e. different data sets) to gain some insight into the nature of the data. Projects don't have to solve problems (or even work as intended) but should explore issues and allow some conclusions about them. This is why constructing an interesting, valid, and testable hypothesis is a good assurance that you have a good project.

The other thing you will need is a suitable data set. If you're not using MNIST data, there are many other datasets available (see Kaggle site for example), - in some cases it's possible to make your own data (with your python skills!).

Most student projects fall into being either about the way the ANNs works, or an investigation into what is present in the data (what features, or properties of the data set make it learnable etc.). Some projects are looking for analogies with biological systems, or how ANNs can be used with different classification problems.

### Scope

The project must be based on the standard ANN model that we have been studying. You can compare with other models if you know about them, but you **can't** do a project based within a different programming environment (such as TensorFlow).

Unless you have an exception approved by your tutor, your project must:

- Use the mnist dataset (or some variant of it you implement)

OR

- use the standard ANN (i.e. Tariq's - or some variant of it that you implement)

OR

- both of the above.

You can use advanced python libraries as appropriate, but, fundamentally, you need to understand what the parts of your program are doing (and how): I want to assess (among other things) how well you grasp how an ANN implements learning and how this is achieved in Python code.

Examples of the titles of projects that students have done in the past:

- Can an ANN learn to count?
- What are hidden nodes looking for?
- What can't the ANN learn and why?
- How good is a simple ANN at learning textures?
- Do ANN and t-SNE make the same errors with MNSIT classification?
- Is learning more sensitive to degraded images or degraded networks?
- What, if any, information in the training data is irrelevant for learning?

- Can neural networks accurately identify the number of dots in an image?
- Can the weight matrices for good and poor learners be distinguished visually?
- How adaptable is an ANN in response to a foreign input in the MNIST dataset?
- How would learning MNIST test images be improved by expanding the training data?
- Are critical periods in learning inherent to, or an extension to, the neural network model?
- How important is the degree of randomisation in the training data for learning performance?
- Can simple changes to MNIST training and test data, such as luminance and contrast, affect learning?
- Artificial Neural Networks That Learn Curve Patterns
- The Effect of Training Sequence on Network Performance
- The Inner Depictions of Numbers in an Artificial Neural Net (ANN)
- Compression of MNIST datasets using autoencoder pre- processing
- Image Distortion and its Effect on the Accuracy of Neural Networks
- A novel data pooling method to increase learning speed in the MNIST ANN
- An exploration of regularisation methods with the aim of modelling neurological deficits
- Investigation of the recognition of four geometric shapes via a multilayer neural network
- Corrupting the test: how artificial neural networks respond to increasingly distorted data
- The interactions between model depth (number of hidden layers) and activation function choice
- Capability of Artificial Neural Networks in Distinguishing Between Left and Right Handwritten Digits
- Insights into the nature of digits: An error analysis for MNSIT classifier by a simple three-layers ANN model
- The effect of rotation on Fashion MNIST dataset performance using a vanilla artificial neural network
- Comparing Latin and Arabic handwritten digits: differences in the effect of learning performance
- Investigating the Application of Artificial Neural Networks in Pharmaceutical Research: A Focus on Lipophilicity Prediction in Illicit Drugs
- The effect of randomly introduced translational changes to the MNIST database on non-convolutional neural network accuracy
- Utilising the complementary strengths and weaknesses of neural networks and decision trees in the context of MNIST digit classification

## Word Limit

The report size, of 3,000 words, is a guide to the scope of the report. It may be that your report can be well-described in less than 3000 words, or it is possible that 3000 words isn't enough to provided all of the information that needs to be conveyed. In such cases there is no need to be concerned about exceeding or not meeting the word limit. However, reports that are overly wordy, discursive, repetitive, or contain extraneous information, could attract mark penalties – regardless of the number word used. Similarly, reports that are below the recommended limit and seem to omit information that would be germane to explaining aspects of the study would be penalised.

## Format

Although there is no prescribed format, and different topics may be suited to different formats, there are a number of features a report should have, and, typically, these are arranged in the following way:

The main subject of the report, which is identified in the report's title, needs to be placed in the relevant context. This "Introduction" section starts by explaining the broad context of the question you are examining in your project, and breaks this issue or problem or question down into the

specific question you examine. There should be a logical connection between the broad question and the reason you have chosen to do what you did in your project.

You must describe what you did and how. You should append your code, and data (whatever it is that you created for and with the project), probably as an appendix, but you also need to explain it. You should use markdown cells in the program to explain what the code cells are doing and why (and the usual # comments for specific bits of code – esp. if they are not standard or common).

Results: it's likely that projects will generate quantitative data, so plots and tables should be used. Matplotlib has a huge number of options. Graphs and tables should be accompanied by legends and explanatory text – it's good practice to have figures and text be able to “stand alone” but to both be included in the results section of your report.

Discussion of the results and their implications: this is the most important part – you need to show insight into what you have, and what you haven't, discovered or produced. Don't oversell, but don't be unimaginative in discussing potential application or implications. A poor discussion section is a recapitulation of the results in a slightly more descriptive way. Instead, try and link the findings back to the questions you raised in the introductory part of your report.

I would not expect a lot of reference to primary literature, you are exploring something that is well known in ML and you do not need to explain what a ANN is – all the concepts we have explored in the subject can be assumed knowledge on the part of the reader (me). So, for example, you don't need to explain what ML is and what a big deal it is, what a ANN is, or what a learning rate is, or an activation function, or back-propagation, etc. etc.

### **How will it be assessed?**

While all of the points above are important in terms of determining the mark for your project, a key overall consideration is the degree of insight you evince in respect to the value, validity, and feasibility of the question you pose, the appropriateness (and limitations) of the way you investigated it, and the value of the outcome (in terms of addressing your question and any further implications).

### **Submission**

Please submit the report in docx format (*not* pdf), and the code used (as an .ipynb), and (if it's not the std. MNIST) the database(s) you use (or a link to it if it's huge). Attach these to an email to me (use “NEUR30006 final project” as the email subject). **Due Date: 28th October.**

---//---