# ME4/MSc Machine Learning Coursework Assignment

Autumn 2021

Undergraduate mechanical engineering students have been set the task of designing a catapult device which will hit a target. This project has been run for many students over several years, and one particularly thorough academic has recorded all the design parameters and how successful each design was at hitting the target. The following parameters were recorded:

- Arm length
- Ball weight
- Ball radius
- Temperature
- Elastic constant of spring
- Weight of device

The performance was recorded as either hitting (1) or missing (0) the target. Data has been provided. This is in a CSV format, with a column for each of the parameters above followed by the performance. There are two datasets, since at a point in the past the design specification was changed. You should look at each dataset separately; they do not have the same distribution. Start with the simpler dataset (1), then move on to dataset (2).

Data has been provided. This is in a CSV format, with a column for each of the parameters above followed by the performance. There are two datasets, since at a point in the past the design specification was changed. You should look at each dataset separately; they do not have the same distribution. Start with the simpler dataset (1), then move on to dataset (2).

In Python, with Keras, design a neural network to produce a probability value of whether the ball will hit the target for the different inputs.

You should submit your neural network models in the HDF5 format, as produced by Keras' model.save() function. This should be named 'familyname-givenname-N.h5', where N indicates which of the datasets you are looking to solve for. This should have the 6 inputs ordered as in the parameter list above and produce two outputs of probability for each class – the first output should be probability of missing, and the second should be probability of hitting. If you need to save scaling parameters, these should be put into a single 2 x 6 array – the first row gives the offset (subtracted first) and the second gives the scale – then saved using numpy.savetxt() with a space as a delimiter (the default), as 'familyname-givenname-N.txt'. You practised saving such scaling parameters in tutorial 4.

*Important note:* An interactive notebook (check_coursework.ipynb, for Colab) has been provided which you should run to confirm that your model can be loaded and works as expected. This loads in the model in the format required and runs the data I have provided through it. It is compulsory that all models are run either through the script or through the notebook. If you do not do this and, after submission, I am unable to load the model, or there is something else wrong with it, then we reserve the right to award zero marks for this section.

The marking will be completed as follows:

| | | | | | |
|---|---|---|---|---|---|
| a | Approx weighting 10% | Conciseness of neural network model for dataset 1 | Excessive numbers of layers and nodes per layer, computationally problematic | E D C B A | Very few layers and nodes |
| b | Approx weighting 40% | Performance of model for dataset 1 | Very poor performance, unusable model | E D C B A | Extremely good classification performance |
| c | Approx weighting 10% | Conciseness of neural network model for dataset 2 | Excessive numbers of layers and nodes per layer, computationally problematic | E D C B A | Very few layers and nodes |
| d | Approx weighting 40% | Performance of model for dataset 2 | Very poor performance, unusable model | E D C B A | Extremely good classification performance |

The weighting of marks for this piece of coursework is as given in the table, although these are indicative and will not be directly averaged. A submission with a major flaw, for example, requiring an unnecessarily large network, is unlikely to receive a strong mark overall. The overall grade will be given as a grade on the standard scale A $\geq 70\% >$ B $\geq 60\% >$ C $\geq 50\% >$ D $\geq 40\% >$ E.

Notes:

- You should utilise the tools we have learnt on the course. There are endless 'exotic' options out there and the intention is not to use the most obscure option.
  - Having said this, you may use other tools (e.g. activation functions) if you wish, but these are not necessary.
- You should be able to complete the coursework using knowledge from lectures 1-6 of the course (and corresponding chapters 1-6 of the notes, plus general info from chapter 11).
- There are no set targets for performance. You should just aim to get the accuracy as high as possible for a general dataset of the same form as the training dataset. You should use tools from the course to achieve this.
- *Treat the coursework as a real-life problem where your boss has given you this task and no more information.*

  - Part of the assessment is in whether you are able to think about the task set and put in place the practical skills talked about on the course to deliver a solution. Your boss wants a 'good' solution – it is up to you to judge what that means and deliver this, using understanding from the course. We will not provide more information since spelling this out in detail would undermine the assessment. Please do not ask for more information.
  - We will not give information about what a suitable network size is – this is for you to establish

- The network does require two outputs, while one would be sufficient for this classification problem. The reason is because in general, when we have three or more classes, we will need a probability associated with each of them, so the single output approach fails. We therefore maintain this approach even for two-category output scenarios. (This has been covered on the course.)

- There is a lot of trial and error in the use of neural networks. This is part of the reason why it is good to build experience by doing some coursework like this! In recognition of this, the percentage of marks allocated to this assignment has been reduced significantly from previous years with more weighting on the exam and other coursework elements.

- If your file does not run through the given test script, please do not contact us telling us the script is wrong...
  - This includes if you find you get a very low accuracy. If you see 0.1 or 0.5 coming out then this is a good clue something has gone wrong (in both these cases you should be able to diagnose the issues..).

- Please follow the spec given for the files. This includes
  - Making sure the files are named correctly
  - Get numbers around the right way for classification (I do not want to have to do 1 minus everything to correct this) - using to_categorical() to generate the two outputs needed is the standard approach
  - Do not transpose the scaling/offset matrix
  - Do not use other delimiters (e.g. ,) in the scaling parameters file.

- All questions about the coursework should be directed to the course leader, not the tutors.

- Please do not ask the tutors to assist you with the assignment. They have been instructed not to do this and will report this to me. They are very able at identifying when someone is asking about the coursework but pretending it is 'another project' too!

- The definition of the scaling parameters file is fairly clearly given, but for the avoidance of doubt, the weighting is performed as:

```
Xt = (XtUnscaled-scaleArray[0,:])/scaleArray[1,:]
```

i.e. the offsets are subtracted off the data first, then the data is divided by the scale values. If your model works with the checking script then it means that you are probably using the scaling array properly.