7CS028: **Coursework Assessment Brief**

(*See Canvas for Submission Details*)

This assessment is designed to test your ability to build deep machine learning models using Tensorflow. The tasks concern deep neural networks, convolutional neural networks and recurrent neural networks. Imagine you have been tasked to build an optimum (**you** justify what is optimum!) neural network model for the tasks below given by a client. Hence, you should write a technical report for each task aimed at someone from a similar background as you i.e., a fellow MSc AI student. You can assume they have some knowledge of programming and neural networks but you still need to explain your model, the code you developed and **your** reasoning why you built this specific model.

- **Submission Date:** See Canvas Assignment Task.
- **All work which you submit must be your own. Cheating is a serious offense and will be dealt with under the University disciplinary procedures.**
- You are required to submit .py files and either a single .pdf which contains the reports for each task or three separate .pdf files of the reports for each task.
- You should make comments within your .py files so it is clear how it works to anyone using it. (**Do not go overboard with the comments though!**)
- **When submitting to canvas make sure you submit all your programs and report files (.py and .pdf) within the same submission.**

You will be assessed on the following criteria for each task:

- Model description and implementation.
- Experimentation, discussion and justification.
- Presentation and structure of the report.
- The clarity of the program developed, clear comments within the code explaining the various aspects.
- The outputs given by you program, any visualisations, are they relevent, clear and well labelled?

The module team are here to help in anyway we can, so please ask for help if you are having difficulty.

Task 1: Deep Neural Networks Task

This task is based around the `flower_data.csv` which can be downloaded from the assignment portal in canvas. Your are tasked to:

- Implement a 2-class classification deep neural network using TensorFlow to model the `flower_data.csv`.
- After experimenting with various model parameters decide on:
  - the architecture to use which optimises performance,
  - which activation functions to use,
  - what optimisation methods to use for training the model and,
  - any other parameters required.
- Explain/justify your reasoning behind opting for the particular model.

You are required to write a short report (2 pages) explaining how you constructed the model, including snapshots of your code to help explain your implementation. You must submit a single .pdf file (report) and a single .py file (code) which can be used to verify your model and the results in your report.

[40 Marks]

Task 2: Convolutional Neural Networks (CNNs)

This task is based around the `flower_photos.tgz` images dataset, which contains 3670 images of different flowers. This dataset can be downloaded from the assignment portal in canvas. Due to limited memory on your device you may only be able to load a small batch of images, say 500, for training the network otherwise your system may be "killed" during training. Use the following commands to only load a batch of images:

```
# load the data from directory (info printed to the console)
data = image_dataset_from_directory(data_dir,
    batch_size=500, image_size=(img_height,img_width))

# extract the images and the class labels
for images, labels in data:
    print(images.shape)
    print(labels.shape)
    break # use "break" here to extract a single batch
```

Once you are happy with your model you can then train on a larger batch of the dataset.

You are tasked to:

- Implement a CNN using TensorFlow to predict images from the flower dataset.
- Preprocess the data appropriately for the CNN model, this includes splitting the data into a training and validation subset.
- After experimenting with various model parameters decide on:
  - the architecture to use which optimises performance,
  - which activation functions to use,
  - what optimisation methods to use for training the model and,
  - any other parameters required.
- You are required to save your final model to a `.keras` file using the `model.save()` function in Tensorflow and load your model in a separate .py file in order to make predictions.

You are required to write a short report (2 pages) explaining how you constructed the model, including snapshots of your code to help explain your implementation. You must submit a single .pdf file (report) and **two** .py files (code): one for training and saving the model and one for loading the model to make predictions, which can be used to verify your model and the results in your report.

[30 Marks]

Task 3: Recurrent Neural Networks (RNNs)

This task is based around the `sunspots_data.csv` which can be downloaded from the assignment portal in canvas. Sunspots are temporary phenomena on the Sun's surface that appear as spots darker than the surrounding areas. Their number varies according to the approximately 11-year solar cycle as shown in figure 1

You are tasked to:

- Implement an RNN using TensorFlow to model the Monthly Mean Total Sunspot Number (see figure 1).
- Preprocess the data appropriately for an RNN model, this includes spliting the data into a training and validation subset.
- After experimenting with various model parameters decide on:
  - the architecture to use which optimises performance,
  - which activation functions to use,
  - what optimisation methods to use for training the model and,
  - any other parameters required.
- Visualise the model by plotting the original data with the predictions produced from the model for the training and test set.

You are required to write a short report (2 pages) explaining how you constructed the model, including snapshots of your code to help explain your implementation. You must submit a single .pdf file (report) and a single .py file (code), which can be used to verify your model and the results in your report.

[30 Marks]

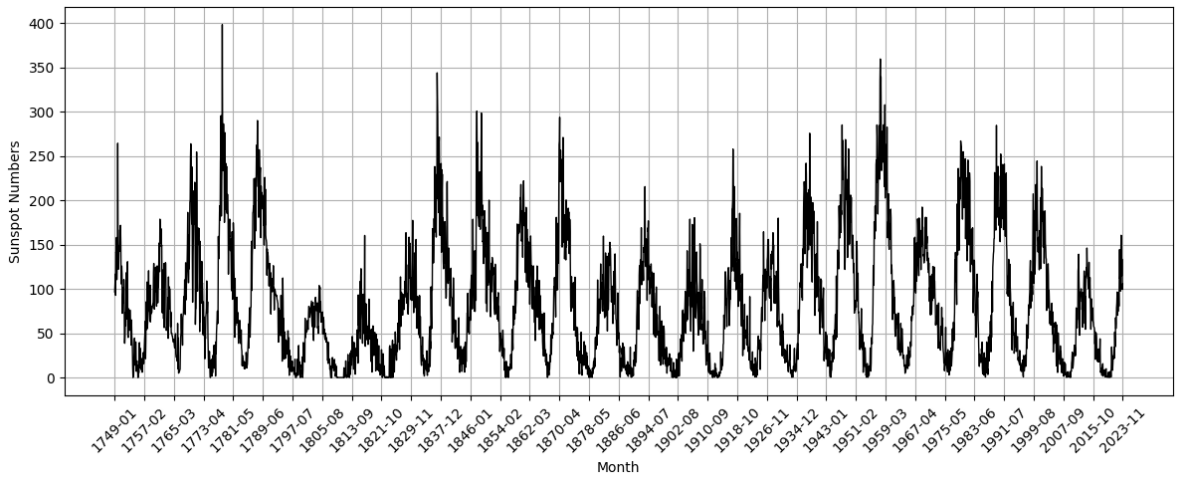

Figure 1: Monthly Mean Total Sunspot Number