Computer and Information Sciences
University of Strathclyde

Week 9 – Neural Networks With Tensorflow (5% of overall grade)

Aim

This week, we are building on the perceptron knowledge that we introduced last week, and introducing neural networks. This is a more open ended lab sheet with no one correct solution. The aim is to produce a reasonable model for both problems. I encourage you to use the forum actively and ask questions! You can also answer each other's questions, because this lab is designed for independent learning. Some guidance is provided in the lecture notes.

NOTES: This will involve installing additional packages such as Tensorflow on Python. If you aren't sure how to do this, look it up or ask in the lab.

When working with extra packages, you are recommended to create a virtual environment to keep your Python install clean. This will stop you from messing your folder up. You should create either a virtual environment (venv) or a conda environment (conda). Again, ask in the lab or look it up.

Part 1: MNIST Network (1%)

Last week, we worked with a single perceptron to identify a single number from a dataset of hand written digits. Now you can build a small neural network to identify digits.

- A file on MyPlace contains the feature extraction and a function for visualisation. You are recommended to do your work in the terminal or an interactive window rather than running files, to minimise the time spent loading files

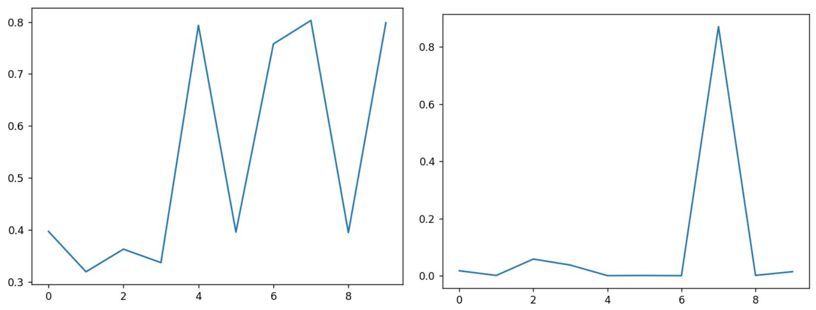
- The week 8 lecture notes introduce model creation, and so you should implement these, make sure you know what is happening, and then try to improve the results. The metric you use should be 'metrics=['accuracy']'

- Your output layer should have 10 neurons, one for each of the numbers, and I don't recommend you use the MSE loss measurement. I have provided you some visualisation plot code that you can work with.

- A poorly designed network will not be able to learn, will not have a clearly identified prediction (see below left), and will have a low accuracy:
  Predicted: [[0.39707124 0.31938416 0.36291438 0.33699358 0.79343987 0.39565927
- 0.75771725 0.8029034 0.39471224 0.79847735]] (class=7)

- A better designed network will generalise well (see below right), and report a clearer estimation:
  Predicted: [[1.7212261e-02 8.0365583e-04 5.8111604e-02 3.7059043e-02 1.9646887e-04
  5.7478528e-04 3.5297933e-05 8.7095451e-01 1.0102795e-03 1.4042095e-02]] (class=7)

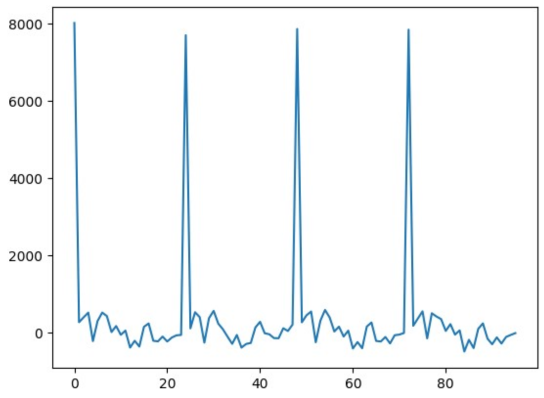
Created March 2024
By Dr Andrew Abel



- Place all your model work in the 'modelling' function, and show it to a lab demonstrator when you are ready

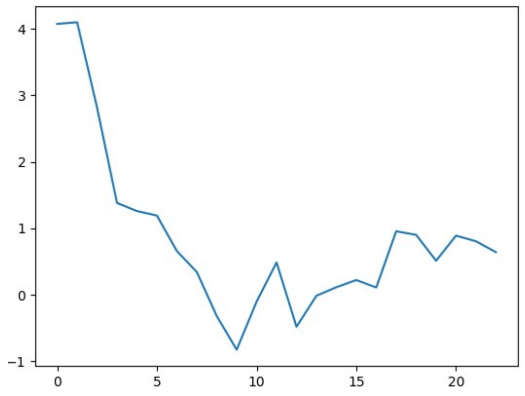
Part 2: Lipreading (2%)

In 2015, I did some work on a lip reading paper (copy is on MyPlace, was discussed in the tutorial), where we tried to estimate audio information using visual information. I.e. given mouth movements, can we estimate the sound?



- Both audio and visual signals are high dimensionality, so we extracted image features (a DCT transform) of the lip region (the lip movements) to create input vectors (see right, top), and we also transformed the audio signal to make a filterbank vector (showing the audio frequency data, right, bottom). To improve results, we combined several frames together into one vector (not recurrent). Part of the dataset is on MyPlace.

- You may need to install extra packages, including Pandas, Scikit-learn to run the initial file



- This is a regression problem, where the aim is to recreate an output vector using continuous data, not to classify, so you will need 23 outputs, one for each component of the vector

- Try to create a model to match the paper. Using the dataset provided, your MSE will likely not be below 2. However, it took me 24 hours to run my models back in 2014. You should be able to do it in a couple of minutes!

- Hint: As it's a large model, try running for a small number of epochs before committing to full scale testing. You should also use the MSE optimiser.
- Hint:Analyse your results. Don't look only at the loss, but look at the output data, see if you can work out what might be happening
- **Hint: Use the forum!** I am happy to answer questions, provide explanations etc.

- We have provided a visualisation for you to look at results in part2.py, and you should place your completed model in the modelling function

- You do not need perfect results. Perfect results are not possible. A reasonable attempt will get credit (loss ~2.5 or better)

- Ideally, you should be able to run this in a terminal, debug window, or interactive window to save time demonstrating

Part 3: Lipreading with a recurrent network (2%)

One thing you may have noticed is that the input data used in the previous frame was simply a concatenation of four frames in one. As you will know from the lecture notes, this ignores the temporal aspect. Convert the data and make it into a recurrent network.

You will find some hints in the lecture notes, but this is independent work for you to figure out yourself.

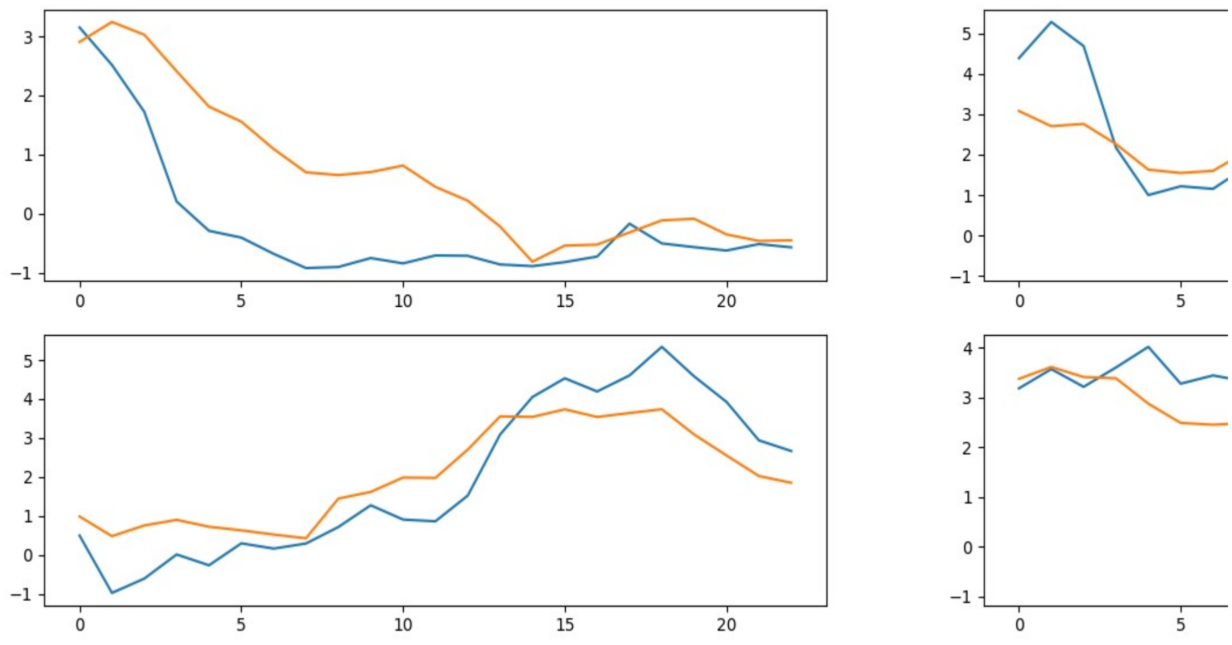
If you have a functioning recurrent network, demonstrate it to the lab demonstrators. Marks will be awarded for understanding your work.

Submission Instructions

When you have working models, show your results to a lab demonstrator. There are 3 parts, and a reasonable attempt will be awarded marks.

Example Outputs

Example of good comparisons between the actual test data (blue), and a well trained network output (orange). Please note that it is not a perfect match!



Created March 2024
By Dr Andrew Abel