EEE4114F ML Projects

Jarryd Son

Introduction

The main focus for these Machine Learning projects is to test your ability to collect and analyse data, choose and implement an appropriate machine learning algorithm for the task and interpret the results using ideas that will be discussed as the course progresses. The actual performance of your system is less of a concern as long as your approach and analysis is reasonable.

These tasks are just to give you basic problems to work towards. You are welcome to adapt them, or come up with your own ideas. If you are unsure if your idea is appropriate, then feel free to contact me via email, or send me a message on MS Teams. Try and keep it simple, you do not have a lot of time/resources to tackle complex tasks. Leave the fancy stuff to final year projects :)

You are expected to investigate, in detail, methods related to solving the problem. There is a design element to the project, and a quantitative evaluation of the performance of the proposed methods must be performed and presented. You are to write up a comprehensive report (of around 10 pages) describing your method and results. You should work in groups of two, although you may work alone if you really want to.

The report should have a clear introduction describing the problem being looked at, and basically what was done. A short literature review should follow, outlining the main aspects of theory used. A description of the work you did should be followed by a comprehensive set of results. These results form the main contribution, and should take the form of one or more experiments that were done in order to interrogate a directed and explicit hypothesis. Results should answer a specific question, not simply provide data on outcome or performance. I find it useful to remember the components required in an experimental write-up: aim, apparatus, method, results, analysis, and conclusion.

Requirements

- Reports should preferably be done in Latex, but Word is fine.
- Please submit as a PDF file
- Jupyter notebooks will not be accepted (last year was a mess)

Task 1: Hand-drawn symbol/shape classification

The idea is to create a system that can take hand-drawn pictures and predict what is being drawn. How you create these pictures and what they depict is up to you, as an example it could be drawn digitally on a smartphone/tablet/PC or you could draw on paper and take photos.

You can simplify the problem (when compared to what quickdraw can do) by:

- Not classifying in real-time. e.g. Ask user to draw and and indicated when the drawing is complete rather than trying to do it while the person is drawing.
- Choosing simple shapes rather than more complex drawings i.e. triangle, circle, square, arrow etc.
- Limiting the number of possible shapes to ease data collection and problem complexity. Five different 'classes' of drawing is probably a reasonable place to start.

If you would like to make things more interesting you could try find an interesting application for this i.e. label hand-drawn electronic schematics or even convert them to Spice schematics for the particularly brave (might even make this a final year project).

Task 2: IMU gesture recognition

For this task the idea is to create a system that can recognise gestures based on movement of an inertial measurement unit (IMU). As with task 1, you can choose how to collect these data and what the gestures are. For example, you can use your smartphone with a sensor recording app such as this to record data, or if you want you can build something with an IMU sensor. Gestures could be a range of movements such as shaking it, rotating it, performing elaborate Harry Potter-esque wand movements etc.

You can keep the problem simple by:

- Not classifying in real-time. e.g. Press to start recording, have a limited time to capture the gesture, then classify. Rather than trying to pick up on random gestures at any time somebody is using the device.
- Choosing short, simple gestures rather than complex movements.
- Limiting the number of possible gestures to ease data collection and problem complexity. For this task I think three different gestures is enough.

As with task 1, you can try look at different ways to use this, don't feel limited by the basic task.

Task 3: μImageNet

For this task the idea is to create your own very small image data set for classification and try to develop a machine learning system to classify the images.

You can find some inspiration from Kaggle image datasets. There is a lot of value in understanding and working with smaller data sets, especially ones that you have made yourselves.

- Keep the data set small and manageable. E.g. choose three to five classes of items and around 50 - 100 images of each. You could take video recordings and pull out individual frames.
- You could compare algorithms e.g. how do complex models perform on small datasets? Are simpler models more effective?
- Analyse the entire process from data collection to model evaluation. What do you need to consider to be confident that your model will work in a real-world situation.

Good luck!