159.372: Intelligent Machines
Assignment 1: Pollen Recognition

Stephen Marsland
s.r.marsland@massey.ac.nz

Semester 2, 2017

This assignment is due on before midnight on Sunday 17th September. No extensions.

1 Introduction

Pollen recognition is a very useful area of research. As any of you that are hay fever sufferers known, pollen can be a complete nuisance, but our gardens would be very boring (although weed-free) without them. Hay fever sufferers are usually only allergic to a few specific pollens, and knowing which they are, so that treatment can be applied only in the exact season, or targeted remedies found is potentially very useful. There are other benefits to being able to identify pollen, such as in *paleobotany*, where core samples are taken from mud and volcanic residue. These include pollen that were captured over time, and give a good indication of how the fauna of the local area have changed. Recognising all of the pollens by hand is a very arduous process, and one that is ripe for machine learning.

There is an active pollen research group at Massey, of which I am a small part. Members of the group capture pollen from the air in season, or from core samples dug into mud or ice, and they would like to be able to reliably detect and classify these pollens. This project includes image processing, feature selection, and machine learning in the form of classification. In this project you are going to work on the last part of this.

You will use a set of features that were extracted from real pollen and will attempt to perform classification of them in two different ways: using a Multi-Layer Perceptron (MLP), and using a Self-Organising Map (SOM). You will try to optimise the parameters of the MLP using a genetic algorithm. You will consider need to consider most of the things that people have to think about for real machine learning tasks, such as choosing network architectures, data normalisation and other preprocessing, and analysis of results. The benefits that you have are that the features have already been selected for you, and that the data is nicely presented so that you don't have to put any effort into extracting it. If you would like to know more about the features that are used, although you don't need to in order to do the assignment, then the paper `PollenPaper.pdf` that is also available on the Stream assignment section describes them.

2 Details

You should use Python for this assignment.

2.1 The Dataset

The dataset is available on the Stream site, from the same place as this sheet. It consists of a zip file (`pollen.zip`) that contains 13 files, each showing a different type of pollen, imaginatively named `pollen1.dat` to `pollen13.dat`. Each file hold data from 50 pollen images, arranged in 50 rows of 43 feature values. Make sure that you can load each file into Python and see the data and plot some of it.

Of the 13 different pollens in this set some are very hard to separate. For instance, those labelled 2, 9, and 13, are all grasses, which are notoriously hard to differentiate.

2.2 Tasks

The marks for each section are given in brackets.

Simple classification with the MLP (30 marks)

- decide on and implement some form of output encoding for the MLP;
- perform any preprocessing of the data;
- separate the data into different groups;
- train a simple MLP (choose a number of hidden nodes that seems reasonable) and see how well you can perform the classification. Use the confusion matrix to output the results;
- test out different sizes of hidden layer to see how many hidden neurons give the best results;

Simple classification with the SOM (20 marks)

- Use the SOM to try to cluster the data and see whether you can identify different pollens in the clusters.
- Both the SOM and the MLP confuse some of the pollens. Are they the same ones in both cases?
- Use a Perceptron to take the activations of the SOM neurons as input and learn the outputs classes. How well does this work compared to the MLP?

Improving the MLP using a genetic algorithm (45 marks)

In the lectures I mentioned that it is possible to use a genetic algorithm to select a good architecture for an MLP for a particular problem. You are going to try this. The way that I would *recommend* that you do this is:

- Design and implement a suitable fitness function;
- Work out a way of encoding the structure of a neural network in a string representation;
- Modify the genetic algorithm code to not perform crossover, and to perform four types of mutation – add a new neuron, delete a neuron, add an extra weight, delete a weight (you will probably want to do this last one by setting the value of a weight to 0 rather than deleting it, which would involve writing your own neural network code, in which case adding an extra weight is giving it a non-zero weight);
- test out your algorithm and then run it on the pollen data;
- compare the results to the ones you got by hand above;

Optional Extras (5 marks)

If you want extra marks there are a few things that you can try:

- test out different subsets of features and see if you can improve the learning. You should automate this process, since doing it by hand would be very, very boring.
- something else of your choice

2.3 Deliverables

The main thing that I want is a *brief* report telling me what you did, together with your (commented) code. The report should include the following things:

- description of the MLP that you used, with the optimal parameters and results you got, plus what preprocessing, etc. you performed
- the parameters that the genetic algorithm selected for the MLP, and the results of that network
- some comments on the difference between the outputs of the two methods
- the results of using the SOM and some comments on the difference in results by using that method
- any extra things that you did

Zip everything together and submit it via Stream.