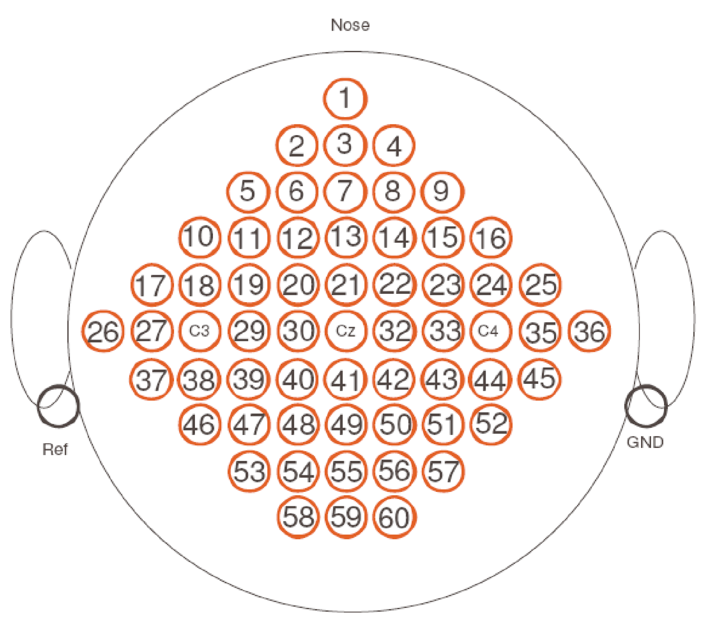
Classifying Multiple Objectives Through Nature-Inspired Techniques

Chris Wirth

Introduction

Brain-Computer Interfaces (BCI) already have a great number of potential applications. If subjects can gain the ability to control multiple independent actions simultaneous, this number of applications can be expanded further, and existing applications can be controlled in more detail and in more dimensions. For example, the ability to control both acceleration and direction would allow a quadriplegic to control a full range of movements for a thought-controlled wheelchair.

The approach taken to this task in this study was to take a single dataset and create multiple classifiers. For each action that could be imagined by the subject, a classifier was evolved to classify 'action X versus NOT action X. Alongside this, a subset of channels to be considered for each action was evolved, choosing from the 60 shown below:



Once each classifier and channel set had been generated, data from the appropriate channels could be passed into each classifier simultaneously, allowing the system to classify any number of the possible actions.

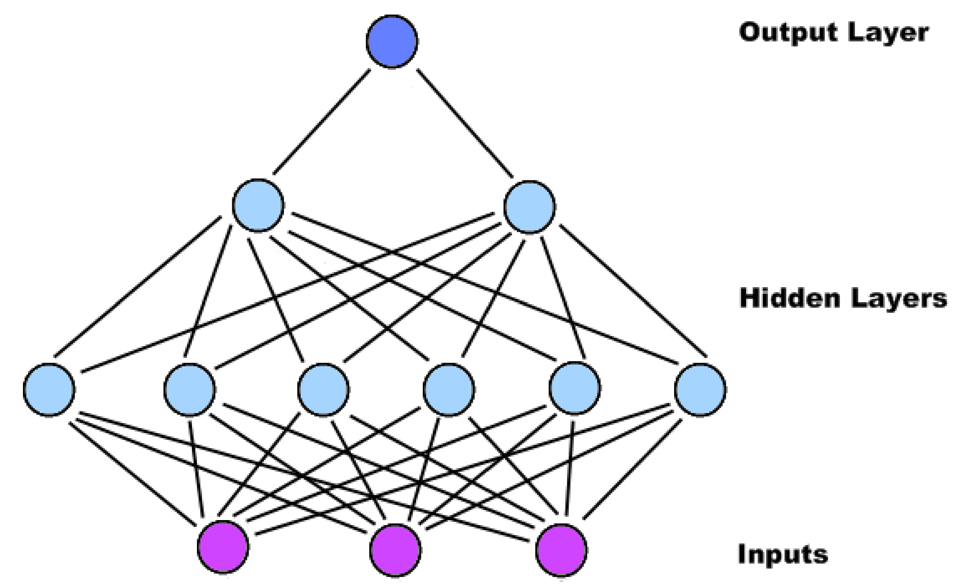
Preprocessing

The data was passed through a Self-Organising Map (SOM) written by a collaborator, Chris Allsopp, before presenting the data to the classifiers. The purpose of the SOM was to convert the data into a discrete form in order to make the task more manageable for the classifiers.

Data Classification

Artificial Neural Network

To classify the data, an artificial neural network (ANN) was constructed. The user could determine the number of layers in the network and the number of nodes in each layer. Connections were made from each node in a layer and each node in the layer above. An example of a network topology is shown in the diagram below:



Particle Swarm Optimisation

With a subset of channels being passed in to the network, the importance of these channels could be weighted. In order to find an optimal set of weights, Particle Swarm Optimisation (PSO) could be employed. Extending this theory, all connection weights for the ANN in this study were set using PSO. At each time step, the velocity of each particle was updated as follows:

$$velocity_t = velocity_{t-1} + \text{distance to best position particle has visited} + \text{distance to best position any particle has visited}$$

The position of each particle was then updated as follows:

$$position_t = position_{t-1} + velocity_t$$

As a consequence of using PSO to set the network connection weights, the optimal solution may not be found on every run. However, as with many natural computation techniques, good solutions can be found relatively quickly.

Channel Selection

The objectives of the channel selection process in this study were twofold:-

- An increase in efficiency: For tasks that require real time processing, computation speed is extremely important. If using a small subset of channels can improve this by a significant margin, it can be considered worthwhile. For some tasks, even if using fewer channels means a slight drop in classification accuracy, it may be worth trading this off somewhat in favour of efficient processing.
- An increase in accuracy: the hope is also that, by selecting a good subset of channels, we can find a set that only uses useful data and discards irrelevant channels that will only be providing noise to the system. if this is the case, classification accuracy can actually be improved by using fewer channels.

A Genetic Algorithm (GA) was used to perform the channel selection procedure. As it is advantageous, from an efficiency standpoint, to have as small a subset of channels as possible, the GA was initialised with 60 individuals in the population, each consisting of a single channel. This allowed a way of testing which channels could be considered most useful.

From then onwards, tournament selection, single-point crossover at a randomly selected point in the string, and bit flip mutation were employed to evolve the population

This was a success, both in terms of computation speed, as shown in the table below:

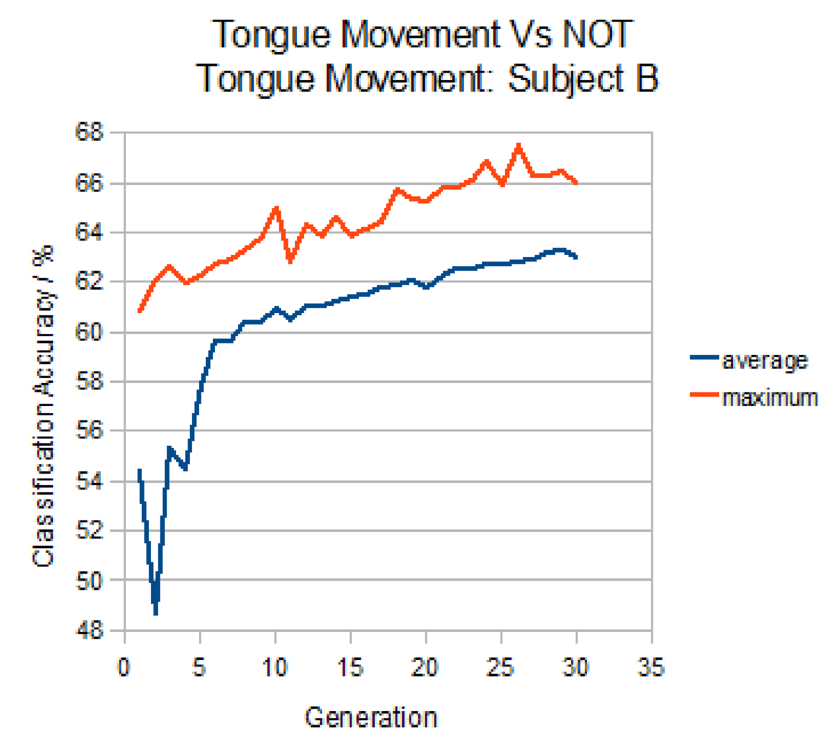
Number of channels	All 60	1
Average time to train a classifier / s	33.73	4.65
Average time to classify a new instance / s	$8\cdot32\times10^{-4}$	$8\cdot32\times10^{-5}$

and in terms of accuracy, as shown below:

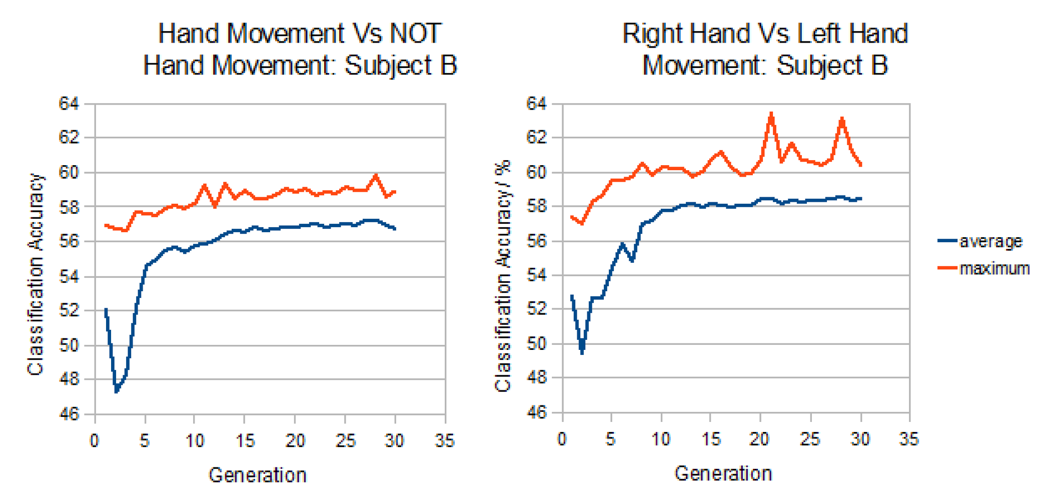
Number of channels	All 60	1	7 (best)
Best classification accuracy / %	57.4	60.9	67.5

Results

The main task in this study was to classify action X versus NOT action X. This was performed reasonably well. The evolution of channel sets can be seen clearly as classification accuracy improves significantly as generations go on. An advantage that we have in the field of BCI is that we can choose to use any brain patterns that the subject is able to generate and suppress at will. As such, we can focus on actions that provide the best results, as these are the ones we would use in practice. An example of these results is shown below:



A secondary task in the study was to classify actions within a subset. The idea behind this is that one could firstly classify, for example, "does the subject want to change direction" and then, if appropriate, "in which direction does the subject want to turn?" - this reduces the number of simultaneous processes that need to be running. An example of this technique is shown below:



With a more detailed dataset, it may also be possible to determine the magnitude of an intended action, allowing, for example, varying levels of acceleration or deceleration to be controlled.