Using Neural Networks to Detect Alcoholism from Simple EEG Features Selected by Genetic Algorithm

Giles Thomson

u4404479@anu.edu.au

Abstract. Electroencephalography (EEG) has been used to detect differences in the brain activity between alcoholics and non-alcoholics. Previous work in designing classifiers have used complex features extraction or deep learning models but here the mean for each EEG channel is used as the only feature with a simple neural network to produce a classifier for alcoholism with an accuracy of 71% and F1 score of 0.77. The effect of varying the threshold on the classification is explored and found to be an effective way of tuning the model for various costs to false classification. The model is further simplified using a genetic algorithm for feature reduction, achieving a model of similar accuracy using only 30% of the original features.

Keywords: Electroencephalography, EEG, neural network, alcoholism, classification, genetic algorithm, feature selection, dimension reduction

1 Introduction

Alcoholism can lead to cognitive inefficiencies and changes to brain activity[1]. Treatment for alcoholism requires there first to be a diagnosis which can be difficult due to misreporting by patients of drinking habits and due the fact that diagnoses can be subjective[2]. This leads to a desire for an objective tool that can be used to screen patients suspected of alcoholism to aid in diagnosis[2].

Differences in brain activity between alcoholics and non-alcoholics can be detected by electroencephalography (EEG)[1]. EEG signals are electrical signals from the brain detected on the scalp in different locations and can be used to investigate brain function[3]. Using these signals might then be used to aid in the diagnosis of alcoholism. To be useful in a clinical setting, an EEG test for alcoholism would have to have high performance at discerning alcoholics from non-alcoholics[4] which can be difficult due to EEG readings often being quite different between different individuals[5]. EEGs produce a large amount of data as they measure several frequencies at tens of sites, hundreds of times a second. Features can be engineered and extracted from these time series EEG data by hand to produce high accuracy[3], or the timeseries can be read through convolutional neural networks (CNN) that find their own features to also get decent accuracy[6].

Engineering features by hand or designing and training deep neural networks may produce good results, but could the simple feature of the mean of each signal be used by a simple neural network to produce results that are good enough, sidestepping the need for high complexity? In this paper, a feed forward neural network with a single hidden layer is used to classify alcoholism using only the mean of each signal from EEG data.

Classifiers may need to take into account the relative costs of false positives and false negatives. In alcoholism screening, it could be that moving a non-alcoholic onto a second stage of screening for alcoholism (a false positive) may be less costly than a true alcoholic missing out of treatment (a false negative). The specific relative costs could vary significantly by setting. Classifiers could be tuned to take these costs into account by simply varying the prediction threshold for classification[7]. Requiring a higher output value from the model to classify a pattern as alcoholic should reduce false positives, and alternatively accepting a lower value should decrease the false negatives. This simple tuning technique is implemented for the alcoholism classifier to see if this variation of threshold can improve the usefulness of the model.

EEG data have high dimensionality and models that interpret them can use genetic algorithm to reduce the number of features used and improve the performance of the classifier[8]. These genetic algorithms build models using a variety of feature sets, and those models that perform the best are used as a basis for the next generation of testing. As the algorithm evolves, a feature set smaller than the maximum may be found that has better performance than when all features are included. A genetic algorithm was used in this paper to further simplify the model created above, by reducing the number of inputs to the model.

2 Methodology

The data was inspected, prepared by reshaping to rectangular data and unneeded features removed, shuffled and split into training and testing data sets by subject ID, normalised, trained and classified by a feedforward neural network, and then finally the model was evaluated. The code was written in *python* and *pytorch* on a Windows environment.

2.1 Data set

The data set was provided as a MATLAB file containing 11057 rows by 5 columns. Column one itself is a vector of the means of a 1 second scan at 256hz of each wave type for each channel of the EEG, with each vector being 192 float values in length. The dataset was flattened out to rectangular dataset, with 11057 rows by 196 columns. The other columns are subject ID (There were 122 subjects), trial number (an integer value between 0 and 119. Each subject had <120 trials in the dataset (mean = 90.65)), a categorical variable that indicated what stimulus was shown to the subject during that scan (an integer value between 1 and 5, approximately evenly distributed), and a binary value indicating whether the subject was an alcoholic or not (64% alcoholic). The subject ID, trial number, and stimulus features were removed before training, as these were not related to the problem of classifying alcoholism by EEG.

The 192 columns of EEG data had a mean value of between 0.6 and 1.5. However, the maximum values were up to ~18 or as low as ~1 (see Fig. 1). Normalising by min-max normalisation would compress most of the signals of the columns with large outliers down to very small numbers but leave relatively large differences in features with smaller outliers, so instead normal standardisation was used on all features, with the training data normalised to a mean of zero and a standard deviation of 1 for each column, with the testing data normalised using the training data's parameters. In this way, most values are between -1 and 1, with the few outliers being brought at least somewhat closer to the central value.

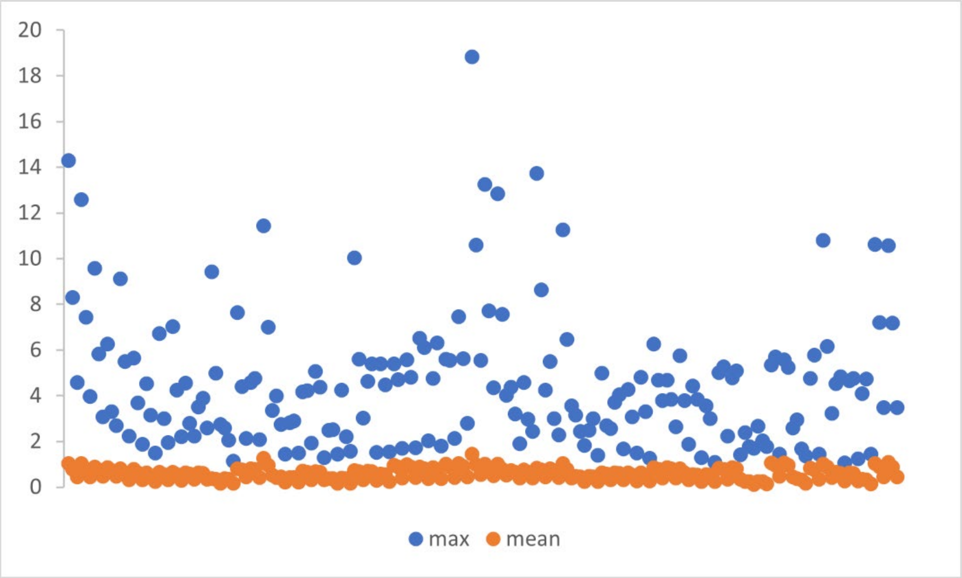

Fig. 1. Mean and maximum values of input features

2.2 Neural Network Design

A feed forward neural network was used for the model. The model had 192 input features and 1 binary output (which was encoded as 2 neurons, one for each class). 10 hidden neurons were used. Adding a few more hidden neurons did not improve performance; adding many more reduced generalisation on testing data and the model performed worse. Reducing the number of hidden neurons by more than a few caused the model to learn to predict all patterns as the majority class (alcoholics).

As the network had only one hidden layer and signal attenuation would not be all that large, a sigmoid activation function was used. The output was also put through a sigmoid function so that the threshold value for prediction could be easily varied from the standard 0.5.

The Adam optimizer was used, and the learning rate was set to $3x10^{-4}$. Stochastic gradient decent was slower and produced a poorer performing classifier than Adam. When stochastic gradient decent was used it required a larger learning rate to work optimally (around 0.01).

The learning was done over 500 epochs on approximately 80% of the data (split randomly by individual). More epochs resulted in little change to performance or eventually in worse generalisation. Fewer epochs resulted in the model not quite converging and so worse performance.

2.3 Training, testing, and evaluation

The order of records of the data was shuffled and then was split 80:20 into training and testing data. The data was split by subject ID so that there were no records of any individual subject in both the training and testing data, producing a cross-subject test for the classifier. As some subjects had more trials in the data than others, this meant that the data was not split exactly 80:20, but within a few percent. A seed of 1 was used in the pseudorandom number generators for *numpy* and *pytorch* before processing the data and building and tuning the model. Once final parameters were chosen, the seed was changed to 2 and the model was trained and evaluated once on this new set of training and testing data. This allowed for the model to be tuned using testing data but not over tuned to it (as mostly different data would be split into the training and testing for the final evaluation) and to conserve the amount of data available to train the final model (i.e. not having to sacrifice more data to a third evaluation set). The performance of the model was evaluated using accuracy, using a threshold 0.5 in classifying the output of the model.

2.4 Prediction threshold variation

The prediction threshold was varied from 0.5 to between 0.2 and 0.8 and evaluated for total patterns correctly identified, false positives count and false negative count, in a similar manner to in Milne et al[7] (see Table 1 and Table 2), in order to find the optimal threshold level.

2.5 Genetic algorithm for feature selection

The model was further simplified through using an genetic algorithm to select features. The features to use were encoding as a Boolean vector of length 192 (total number of original features), with an initial probability of selecting each individual feature set at 10%. A low probability of selecting features was chosen so that there was a possibility of finding a good solution with few features. Fitness was measured by the accuracy each model had in predicting the test data, so performance could easily be compared.

A population size of 15 was used and the number of generations was set to 15. A larger population size would have allowed for a broader search of the solution space, but computation power and time was a constraint, as the model was built on an inexpensive laptop without GPU acceleration and a budget CPU. The number of generations was increased until it became clear that convergence had been reached.

Selection of individuals for the next generation was done by proportional selection, with the relative fitness of each individual determining the probability of it being selected. Once a new generation was selected, uniform crossover was applied at a rate of 80%. This was followed by each individual being mutated by bit inversion, with each bit having a mutation probability of 5%. The high rates of crossover and mutation were chosen to ensure high variation was maintained despite the small population size.

3 Results and Discussion

On evaluation data, the model using the 192 features had an accuracy of 71% with an F1 measure of 0.78. In comparison, deep learning models using CNN for feature extraction and autoencoding for improved training have had accuracies of 76% using the full version data set of the same data (i.e. using the whole time series instead of the mean)[6]. Using different data with an SVM and more carefully engineered and selected features produced accuracies of up to 98%[3].

Although this simple model does not produce state-of-the-art performance, it does have the advantage of being reasonably fast to train and the features being easy to extract. With only a little over a 70% accuracy with an evaluation dataset of 59% positives, there is a small improvement over simply classifying all cases the majority class.

If a classifier were to be used in screening for alcoholism, false positives and false negatives may have different costs (i.e. moving a non-alcoholic onto a second stage of screening for alcoholism may be less costly than a true alcoholic missing out of treatment due to a false negative). For this reason, the threshold for classifying a pattern as alcoholic was varied from the default 0.5, following the technique outlined by Milne et al[7], so that the full range of options could be found. In both training (see Table 1) and evaluation data (see Table 2), the lower end of the threshold is 0.4, where there are no false negatives. Similarly, the upper end is at 0.75 for both data sets, where there are no false positives. Depending on the relative costs of false negatives and false positives, the classifier can be tuned to the right level using these data.

Table 1. The results of changing the prediction threshold on the training data

Threshold	Correct	False +ve	False -ve
0.4	5698	3105	0
0.45	6301	1903	599
0.5	6405	1533	865
0.55	6373	1280	1150
0.6	6276	1054	1473
0.65	6090	797	1916
0.7	5480	494	2829
0.75	3105	0	5698

Table 2. The results of changing the prediction threshold on the evaluation data

Threshold	Correct	False +ve	False -ve
0.4	1334	919	1
0.45	1591	484	179
0.5	1614	379	261
0.55	1609	296	349
0.6	1583	227	444
0.65	1530	165	559
0.7	1358	84	812
0.75	919	0	1335

Measuring the effects of moving the threshold on other measures of performance can be used to evaluate the model. Looking at the relationship of false positive rate and true positive rate at different thresholds leads to an ROC (Receiver-Operating Characteristic) curve which can be used in a similar manner to above to evaluate the model at a range of threshold values, or the area under the curve (AUC) can be used to evaluate and compare models over the whole range[9]. The ROC of the model from this paper (see Figure 2) had and AUC of 0.76, compared with the >0.95 of models that used carefully chosen and engineered features[3].

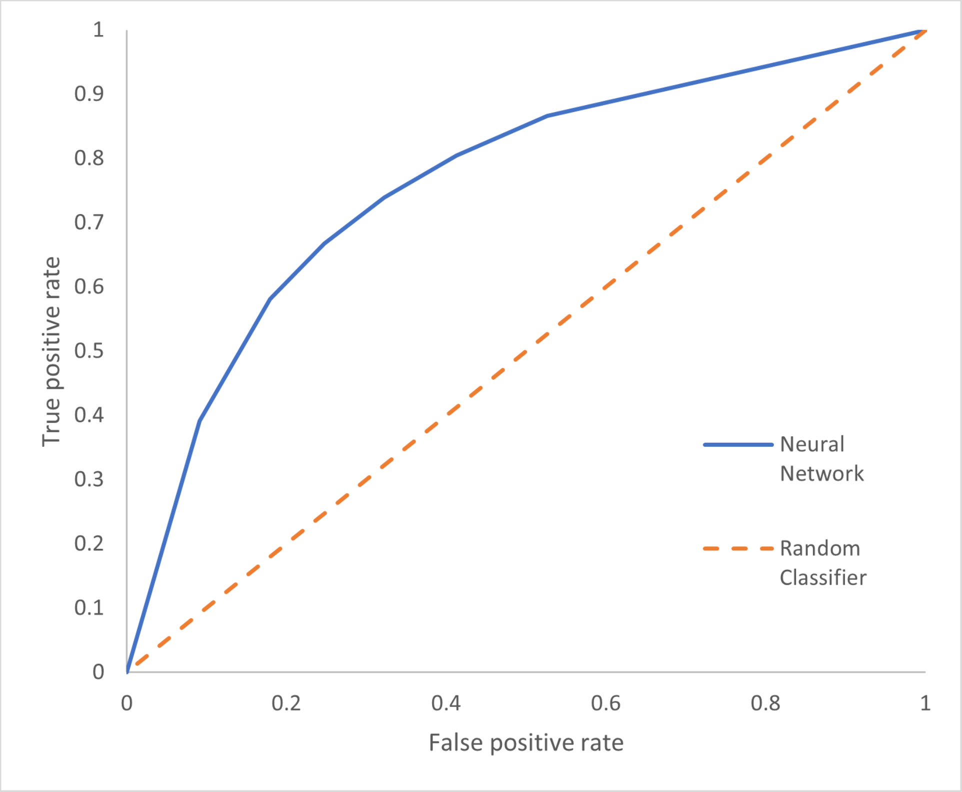

Fig. 2. ROC curve for the neural network classifier. The ROC-AUC was 0.76

It was hypothesised that using a subset of features available could improve the performance, as this has been seen in the past[8]. However, when a few features were removed that had high correlations with another feature, performance did not improve appreciably. When a genetic algorithm was used for feature selection, performance converged at ~71% accuracy again (see Figure 3). However, far fewer features were required to achieve this similar performance, with a model of this accuracy found at generation 6 (51 features, 27% of 192). Previous work has managed to reduce the features down to 0.5% of the original amount[8]. These models, however, were starting from many more features (over 12000), which brings the optimal amount found here to a similar absolute number of features as found previously.

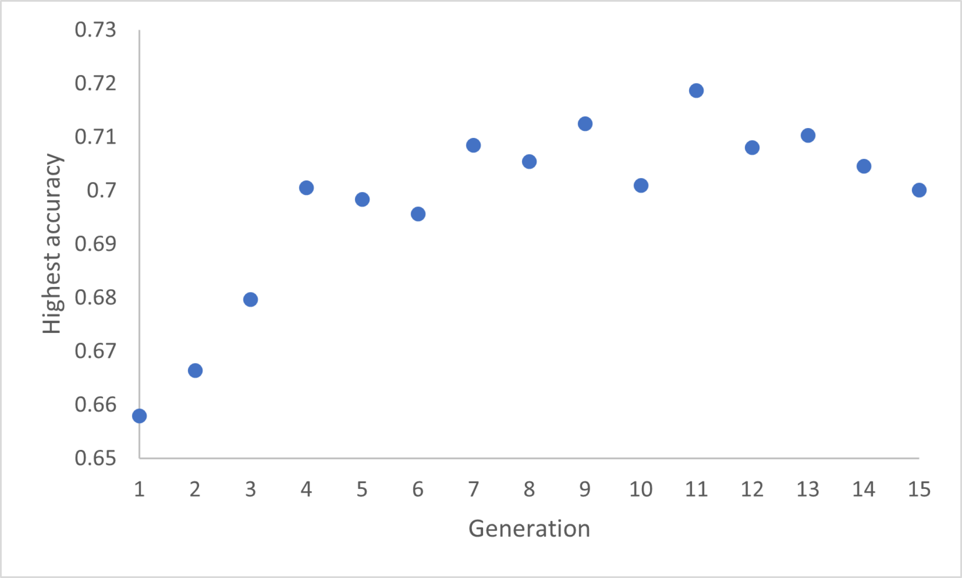

Fig. 3. Accuracy of fittest individual for each generation of the genetic algorithm

The accuracy did not improve much from generation 6, although the algorithm continued to add features until it hit the generation limit (see Figure 4). As there was no penalty for choosing more features, continually crossing over parents to produce offspring would naturally increase the number of features until the population generally had around 50% of the total possible features chosen.

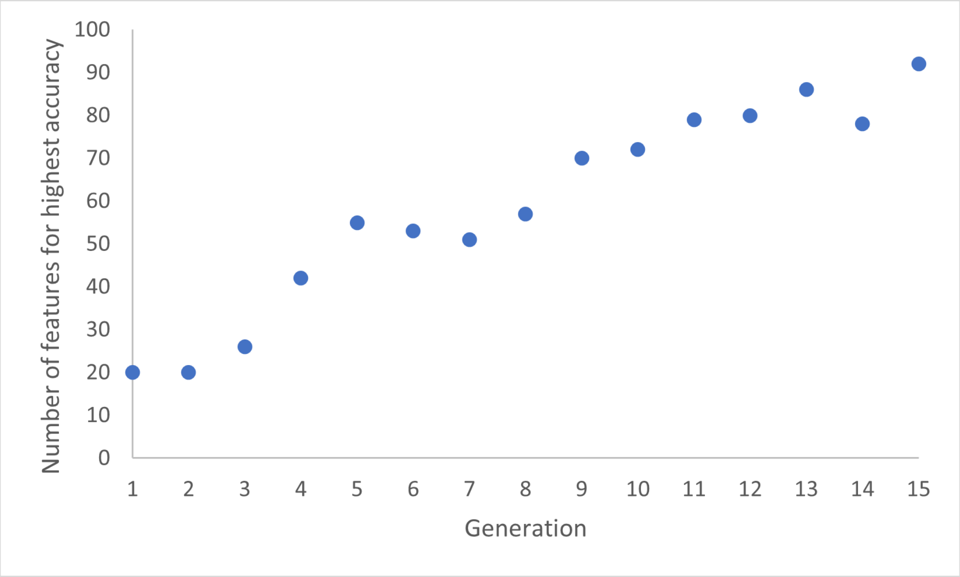

Fig. 4. Number of features used by the fittest individual for each generation of the genetic algorithm

The simple features used in this model (mean of channel over the whole time period) is not enough for state-of-the-art performance. The differences in activity between alcoholic and non-alcoholic brains are more complex than what can be captured by its mean activity. Features previously used with better success[2,3,6] were not available or extractable in this summarised data set. Increasing the complexity of the either the model used (CNN on full time series) or increasing the complexity of the data (more complicated extracted features) would improve the performance of the model over the simplified data used in this model.

The performance of the model is not good enough to be used to screen for alcoholics. The rate of alcoholism in the general population is difficult to estimate but is likely has been estimated to be up to 10%[3]. The dataset used had 64% alcoholics, so a random person from the general population being classified as an alcoholic by this classifier is likely to still not be an alcoholic. Using a more balanced dataset in training may help improve learning, and using a dataset with a similar proportion of alcoholics to the general population in the evaluation may give more realistic results.

4 Conclusion and Future Work

Using only mean values for each frequency band for each EEG channel in a feed forward neural network to classify alcoholism leads to an accuracy of 71% and ROC-AUC of 0.76, below state-of-the-art level. Adjusting the threshold level of the classifier does not improve accuracy but can be used to easily tune the ratio of false positive to false negatives to a desired level, improving the usefulness of the model without having to tweak or retrain the model.

Using an genetic algorithm to select features does not improve accuracy appreciably but does reduce the size of the feature set required to about 30% of the original size (as few as 51 features for 71% accuracy). Training and predicting with fewer features reduces complexity and required computation time. A larger population size could be used in the future to explore more of the solution space, especially in early stages where fewer features are selected. Elitism, where the best individual is kept from one generation to the next was not used and may improve performance on this model if implement. A size penalty could be included in the fitness function to keep the features set from continually growing as the genetic algorithm progresses, forcing smaller models to be explored. The genetic algorithm could be extended to optimise the hyperparameters of the neural network model, including the number of hidden layers and the size of each hidden layer, to improve the structure of neural network and its performance.

If the dataset was modified to be more balanced for training, the model may learn better. For example, some of the patters of the positive class could be removed, or some of the negative class could be oversampled, perhaps with some slight noise added.

This dataset was constructed from a more complete dataset by simply taking the mean of each time series. A dataset with a few more features that describe each time series beyond the mean, such as the standard deviation and the maximum value, may be larger but still simple and could capture more information about the differences in alcoholic and non-alcoholic brain activity, allowing for a higher performing model to be built.

References

1. Parsons, O.A., Nixon, S.J..: Cognitive functioning in sober social drinkers: a review of the research since 1986. J. Stud. Alcohol. 59(2), 180--190 (1998)
2. Mumtaz, W., Vuong, P.L., Malik, A.S., Bin Abd Rashid, R.: A review on EEG-based methods for screening and diagnosing alcohol use disorder. Cogn Neurodyn. 12(2), 141--156 (2018)
3. Ahmadi, N., Pei, Y., Pechenizkiy, M.: Detection of Alcoholism Based on EEG Signals and Functional Brain Network Features Extraction. IEEE 30th International Symposium on Computer-Based Medical Systems (CBMS), pp 179--184. Thessaloniki, Greece (2017)
4. Jurado-Barba, R., Sion, A., Martinez-Maldonado, A., Dominguez-Ceteno, I., Prieto-Montalvo, J., Navarrete, F., Garcia-Gutierrez, M.S., Manzanares, J., Rubio, G.: Neuropsychophysiological Measures of Alcohol Dependence: Can We Use EEG in the Clinical Assessment? Front. Psychiatry. 11, Article 676, 1--17 (2020)
5. Fdez, J., Guttenberg, G., Witkowski, O., Pasquali, A.: Cross-Subject EEG-Based Emotion Recognition Through Neural Networks With Stratified Normalization. Front. Neurosci. 15, Article 626277, 1--14 (2021)
6. Yao, Y., Plested, J., Gedeon, T..: Deep Feature Learning and Visualization for EEG Recording Using Autoencoders. In: 25th International Conference, ICONIP, pp 554--566. Siem Reap, Cambodia (2018)
7. Milne, L.K., Gedeon, T., Skidmore, A.K.: Classifying Dry Sclerophyll Forest from Augmented Satellite Data: Comparing Neural Network, Decision Tree & Maximum Likelihood. Conference: Proceedings Australian Conference on Neural Networks (1995)
8. Yaacoub, C., Mhanna, G., Rihana, S.: A Genetic-Based Feature Selection Approach in the Identification of Left/Right Hand Motor Imagery for a Brain-Computer Interface. Brain Sci. 7(1), Article 12, 1--15 (2017)
9. Kelly, H., Zou, A., O'Malley, J., Mauri, A..: Receiver-Operating Characteristic Analysis for Evaluating Diagnostic Tests and Predictive Models. Circulation., 115(5), 654--657 (2007)