Improve neural network performance in classification problems using fuzzy clustering and neuron pruning

Vera Su

School of Computing, Australian National University
u6379580@anu.edu.au

Abstract. The development of fuzzy set logic has allowed for machine learning classification without absolutes, that is degree of truth rather than a hard, binary classification and therefore can be more suitable for certain, real-world scenarios. One such scenario could be in depression detection. Additionally, network reduction through distinctiveness analysis of hidden neurons have been shown to be an effective technique in the image compression domain. This technique uses vector distance between hidden neurons to determine if they are functionally identical or complementary and removes it from the network. Here we apply fuzzy set logic through c-means clustering and using the resulting membership values to assist a neural network in learning to a classification domain where observers' physiological signals are used as a predictor for an patient's depression level. Additionally, we attempt to improve computational performance through distinctiveness pruning. We achieve 26.63% accuracy in the basic neural network model, 26.90% with c-means and 25.57% after pruning the c-means model. Therefore, we conclude that although c-means is not a suitable method to improve accuracy of a neural network model, distinctiveness pruning shows promise as an effective method of network reduction in classification problems.

Keywords: neural network, pruning, network reduction, fuzzy c-means clustering

1 Introduction

At its essence, depression is a clinical syndrome defined by the presence of several core symptoms. These are: depressed mood, loss of interest or pleasure, decreased energy, and fatigue. In addition to the core symptoms, loss of self-esteem, suicidal ideation and feelings of worthlessness or guilt are also common. Despite well-defined diagnosis criteria, modern psychologoy relies on limited empirical research (Paykel, 2008). Preliminary data has demonstrated physiological responses as a subconscious indicator of emotional state. In particular, depressed patients have shown different eye gaze behaviours, lower Galvanic Skin Reponse (GSR) and reduced Heart Rate Variability (HRV) (Zhu, et al., 2019). These responses are uniform and quantitative, which can be combined with machine learning technology such as neural networks to assist with diagnosis.

1.1 Dataset description

The dataset used for this study was initially collected by Zhu, et al., (2019) for the paper Detecting emotional reactions to videos of depression. Zhu, et al., (2019) was able to successfully measure the same psychological signals of depression in observers to assist with identifying depression in others. That is, they measured the psychological signals of observers (with no prior knowledge or expertise in depression recognition) in response to videos of individuals with depression and used observer signals to classify the individual's depression levels. This dataset contains a total of 192 samples, split across five observers. These observers have generated a total of 86 features. Features relating to pupillary dilation (PD) describe pupillary size, very low pass (VLP), and low pass (LP) signals for both the left and right pupils, as well as the average of each across both eyes. Additionally, summarised data on skin temperature (ST) and GSR have been collected.

For this study, only the data on PD have been included. This is because using only PD data consistently outperformed GSR and ST data, as well as models trained on an aggregate of all three. Additionally, there are number of features to the number of samples, which increases the risk of overtraining.

1.2 Task description

Structured network pruning is a popular approach to optimise and compress a neural network by removing redundancy without sacrificing model accuracy (Tran, et al., 2020). In practice, when a neural network model under-performs, additional hidden neurons are added to improve performance, however this commonly leads to redundant neurons with duplicate functionality and slows the speed of the network. Other use cases of network pruning might be for later rule extraction or for better generalisation.

One method proposed by Gedeon & Harris (1991) of pruning networks is through analysing the *distinctiveness* of each neuron by comparing the angular differences of the activation vector. This activation captures the output activation of each neuron for each input pattern and is a representation of the functionality of the hidden neuron within input space. Hidden neurons which have been determined to be too similar in functionality (redundant), or complementary in functionality are then removed.

This work was later extended upon to determine feasibility in determining underlying neuron function through the activation vector itself, as it is a representation of each hidden neuron in space (Gedeon, 1996). Here, redundant neurons were further defined as having angular separation less than 15° or more than 165° (which are complementary) and are thus removed. This distinctiveness technique was used for image compression.

Another interesting facet of depression is that it is commonly considered to be a spectrum, many subjects with depression do not meet the diagnostic criteria despite benefiting from treatment for depression (Angst & Merikangas, 1997). The original dataset used the self-reported Beck Depression Inventory – II (BDI-II), which rated depression severity on a scale of 0 to 63. However, these values were separated into four broad cateogries; no or minimal deperession, mild depression, moderate depression and severe depression.

Fuzzy logic allows for classification without absolutes where categories may not be precisely defined into exactly zero and one. In these cases, instead of belonging to one or the other, an element belongs to all categories to a certain degrees. This kind of fuzzy set logic can be applied to cluster analysis, as proposed by J.C. Dunn (1973) through the C-means clustering algorithm. Essentially, c-means clustering calculates how much each data point belongs to each category (or cluster) and thus lends itself well to a problem domain such as depression detection where an individual's depression may lie somewhere between multiple categories (Bezdek, 1981).

Therefore, in this study, we aim to improve the performance of depression classification through the use of fuzzy C-means clustering. This will be done through comparison of three models. A basic neural network using the pupillary distance dataset, a second basic neural network model with the class membership values obtained from c-means clustering, and finally a similar neural network model with c-means clustering and neuron pruning.

2 Methodology

2.1 Data preprocessing

There were a total of 85 features and 192 patterns in the raw dataset provided for the study completed by Zhu, et al., (2019). 24 features relate to GSR, 24 to ST and 41 for PD. Given that the best performance achieved by a basic neural network used just the PD data, GSR and ST data have been omitted from our models.

Some common data pre-processing techniques include normalisation or statistical standardisation (Kuźniar & Zając, 2015). However, on inspection of the dataset, all features had already been normalised and standardised. These pre-processing steps included extracting standardisation features (eg. Minimum, maximum, mean, standard deviation etc.) from the gathered signals, and applying a low pass Butterworth filter for normalisation.

Although a majority of of pre-processing had already been completed, some data were to different scales (such as average occurrence) which would increase difficulty of the model. Therefore, all data was once again standardised such that such that the mean of observed values is zero and standard deviation is one. Further, the implementation of cmeans required the dataset to be two-dimensional, thus we used Principle Component Analysis to reduce the dimension of our dataset.

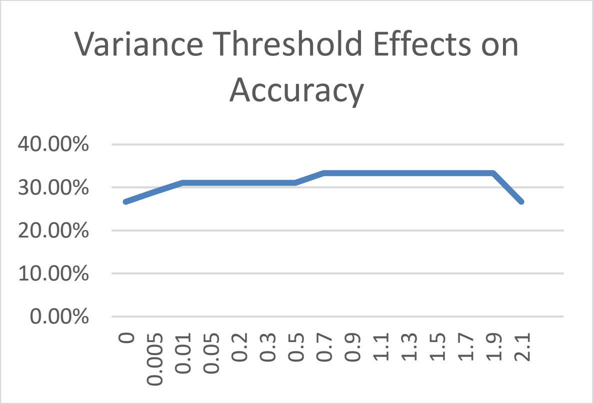

Fig. 1. Results of threshold value on model accuracy on a basic neural network

After initiall attempts with training, we found the model faced significant overfitting that could not be resolved through tuning hyperparameters, thus we attempted to remove features by setting a Variance Threshold where features with variance lower than 0.01 were dropped, leaving 15 features in the dataset. The threshold was chosen after experimentation and results given in Figure 1. However, this also was unable to improve accuracy by any significant amount without removing the added features from c-means clustering and therefore variance threshold was ultimately not used in the final models.

2.2 C-means clustering

The process of c-means clustering can be summarised into the following steps (Bezdek, 1981; Miyamoto & Umayahara, 2002):

1. Choose the number of clusters
2. Assign random membership values for each cluster to each data point
3. Compute the centroid for each cluster
4. Compute the membership values for each data point
5. Repeat steps 3 and 4 until convergence

In this study, we used the implementation found in the sklearn.fuzzy package. Firstly, the optimimal number of clusters were determined by testing various cluster numbers and examining the fuzzy partition coeeficient, which describes the cleanliness of the partitions. Two clusters resulted in best clustering results. FPCs of other cluster numbers are shown in Figure 2. Using this optimal number of clusters, we then obtained the membership values for each datapoint and appended it to the dataset for next steps.

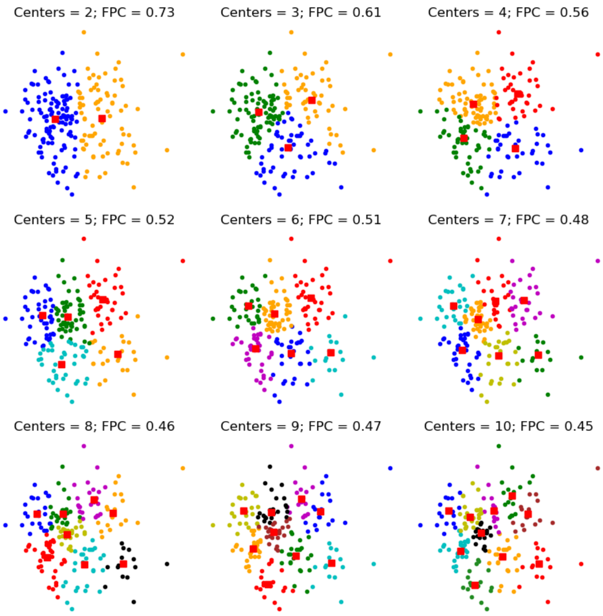

Fig. 2. Fuzzy partition coefficient of various cluster numbers on the depression detection dataset.

2.3 Neural network structure

The basic neural network structure is a classification model with a Variance Threshold based feature selection. All models have a signle hidden layer and an output layer with four neurons, corresponding to the four target classes. These were fully-connected, feed-forward neural networks using the sigmoid activation fuction. Additionally, all neural networks were trained on the Adam optimizer and backpropogation with cross-entropy loss

We then spent some time tunining the hyperparameters of the base neural network model and the model using c-means clustering. The C-means augmented model with pruning will use similar hyperparameters to the base c-means model. These three models will allow us to determine if a c-means augmented model will improve classification accuracy, and whether distinctiveness pruning can help reduce network size without sacrificing accuracy when implemented on a fully trained model (Gedeon, 1996). In addition to the pre-determined and tuned hyperparameters, we experimented with learning rate and training epochs by analysing changes between training progress and final testing accuracy. The results of hidden neuron number are given in Figure 3 and 4.

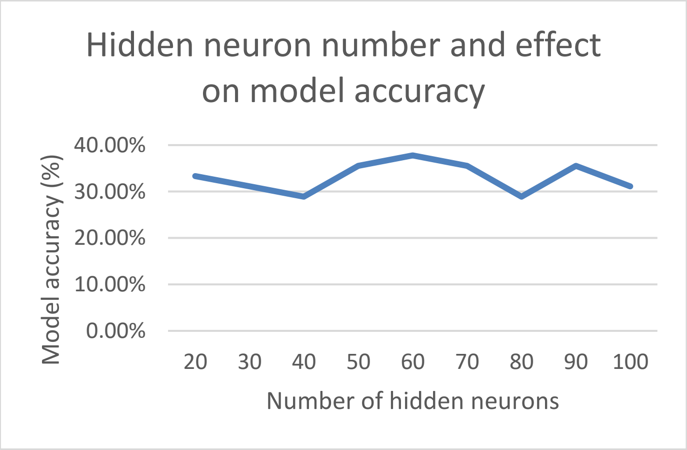

Fig. 3. This figure shows how the test accuracy of the basic neural network model changes depending on the number of hidden neurons

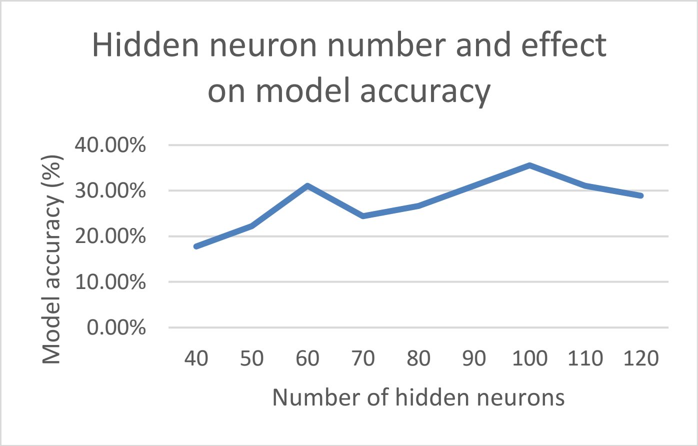

Fig. 4. This figure shows how the test accuracy of the cmeans augmented model changes depending on the number of hidden neurons

As seen in Figure 3 and 4, number of hidden neurons did not seem to change significantly once a certain threshold was reached, however there was a significant spike for both models which was able to bring the accuracy close to 40% for that particular run. Therefore, we chose the number of neurons that gave rise to the peaks.

Best results for the basic neural network model were given by a 40 hidden neurons, 1000 epochs and a learning rate of 0.01. Alternatively, best results for the c-means augmented model were given by 100 hidden nerons over 800 epochs with a learning rate of 0.0005. It should be noted that both models only had marginal improvements from the base values which started with 50 hidden neurons, 500 epochs and a learning rate of 0.001.

2.4 Distinctiveness pruning

The process of distinctiveness pruning can be summarised with the following steps (Gedeon, 1996; Gedeon & Harris, 1991):

1. Compute the activation vector of the hidden layer
2. Normalise the elements of the vector to use the angular range of 0-180˚
3. Calculate vector angle between each pair of hidden neurons
4. If angular distance is greater than 165˚ or less than 15˚, remove one of the two neurons in the pair
5. Add weights of the removed neuron to the remaining neuron

In this study, we implemented distinctiveness pruning slightly differently due to computational restrictions ascaclulating and pruning every redudant neuron is computationally heavy (though results in a more efficient model). Every 50 epochs, the output activation vector was calculated and normalised to lie between 0° and 180° by subtracting 0.5 to the sigmoid activated hidden layer. Then the angular distance between each pair of neurons was calculated using cosine similarily. This formula is given below:

$$angle(i,j) = \frac{\cos^{-1}(i \cdot j)}{\pi}, \text{ where i and j are vectors.}$$

After obtaining the angular similarity, one pair of neurons with similar or complementary angles is randomly selected for pruning. This was implemented by shuffling the calculated angles and iterating through the list until the first pair meeting the prune criteria is found. It should be noted that this method does not require further network training afterwards and can be used as is, thus expensive computations only need to be done once, unless the model is later retrained.

2.5 Evaluation

Depending on how the dataset is split into training and test sets, and the randomly initialised weights, the same model can give varying results from run to run. In order to combat this, we used two methods of evaluation for our models. During the hyperparameter tuning stages, we simply set a random seed to ensure that all aspects of the model remain the same, except for the hyperparameter being changed. Although this does not give the true accuracy of the model, it is unnecessary to determine it at this stage. Instead, we are easily observe the effect of the hyperparameters.

For the final model evaluation to compare the the basic model with the model trained on c-means augmented data, and the c-means augmented model to the pruned model, we ran each model 10 times and used the average to compare. We also used other extracted statistics such as mean and standard deviation across all ten runs to determine if there is a difference between the models, and if so, which model performed better.

Our method differs from the evaluation method used by Zhu, et al., (2019) in the original paper for the depression detection dataset. There they used a leave-one-participant-out cross validation method which guarantees all datapoints for one observer will be kept together, rather than split between training and testing sets. This was not possible for us as participant information had been redacted from the distributed dataset.

3 Results and Discussion

	Average	Std. Dev.	Min.	Max
Basic NN	26.63%	5.06%	20.00%	38.24%
With C-means	26.90%	7.05%	15.38%	38.89%
With C-means and Pruning	25.57%	5.19%	18.75%	34.38%

Table 1. This table shows the final testing results of the basic neural network model, the c-means augmented model, and the c-means augmented with pruning model, comparing average accuracy, standard deviation, minimum accuracy and maximum accuracy over ten runs.

The average test accuracy for all methods are shown in Table 1 and shows no significant difference between the three methods. This result is promising in the case of distinctivness pruning as it demonstrates that the number of neurons can be reduced (thereby also reducing computational costs) without reducing model accuracy. Despite promising results in network reduction, it should be noted that all models performed at a rate of random chance (as there are four target classes) and in fact performed worse in some cases as indicated by the worst accuracy over all runs listed in the minimum column. These results are significantly worse than the 88% accuracy achieved by Zhu, et al., (2019) and therefore further research will need to be conducted to verify the effects of distinctiveness pruning.

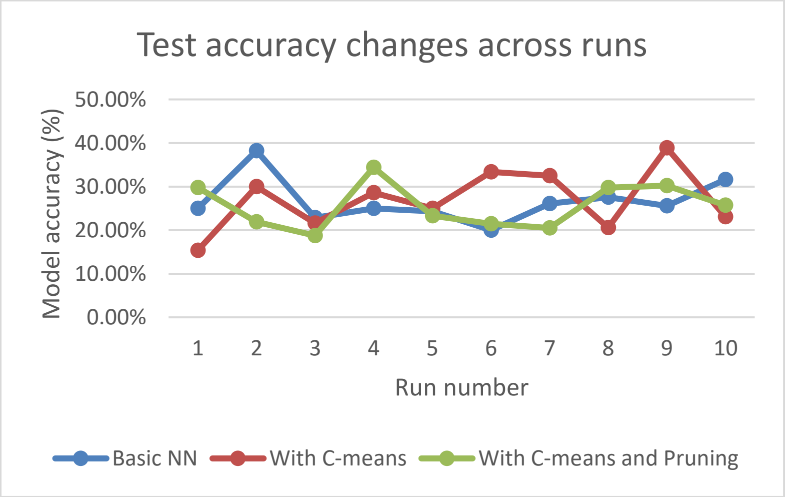

Fig. 5. This figure shows how the test accuracy changes over ten different runs for all three neural network model: the basic neural network, c-means augmented model and cmeans augmented and pruned model.

This disrepency in accuracy could be caused by several reasons, however we believe the most likely reason is the Cross Validation method used. More specificially, Zhu, et al., (2019) noted that random data-partitions were not suitable for datasets such as these; instead the data should be split by groups rather than individual data points. However, this method could not be used for our study as it had been completely redacted from the provided dataset and thus would have significantly affected the ability of the model to learn patterns. Individual responses to videos of patients with depression can vary, therefore our model which lacks the appropriate data partitioning method could have been significantly impaired as a result. This is supported by Figure 1, which shows how accuracy changed over the runs. In particular, we can see that accuracy sits at 25% for all models, but significantly spikes up and down in certain runs. This may be due to random splits separating more or less observer data, leading to different learning.

Although c-means clustering was employed to improve model learning, this was unsuccessful as the model performed almost identically to just the basic neural network. That is to say, both the basic neural network and c-means augmented neural network performed at a rate of random chance. One possible explanation for this could be that the model was simply unable to learn this dataset, therefore any addition of features could not improve the model. To verify this, we would simply have to attempt this method on a different model that was learning, but not to a satisfactory standard. Another explanation could be due the to number of clusters chosen during the clustering phase. Here, we chose the number of clusters based on FPC, not the number of target classes. Given the difference, it is also likely that the clusters were classifying based on features irrelevant to the goal of our neural network models. Further, it can be seen from the clustering results in Figure 2, the data could not be cleanly separated into clusters, therefore indicating that c-means was not suitable to use for performance improvements, if a dataset could be cleanly partitioned by c-means, it may yield more improvements when used in conjunction with a trained neural network.

4 Conclusion and Future Work

This paper explored a method of using fuzzy set logic, specifically c-means clustering to improve performance of an underperforming classification model. This was done by adding the resulting membership values for each cluster as a feature prior to training the neural network model. Additionally, we explored the distinctiveness pruning method to improve computational efficiency without sacrificing model accuracy. Thes methods were applied to a classification which aims to use observer reactions to patients with depression to determine the severity on a scale of 0 (none) to 3 (severe). This application is significantly different to the initial domain to demonstrate the distinctiveness pruning method, which focused on an image compression problem instead.

The poor results of all three models demonstrated an inability to learn the underlying patterns. Despite training accuracy hovering at 95%, the test accuracy averaged at 25 to 26%. This showed significant overtraining problems which could not be alleviated through basic feature selection methods like Variance Threshold nor early stopping, reducing neuron numbers and other similar approaches commonly suggested for overtraining scenarios. Similarly, augmenting c-means results into the featureset did not impact accuracy in any meaningful, yet required additional computational time for data clustering. As this is only an initial study, further studies into this method using different data domains, or different modelling techniques may see more successful results.

Although the model did not train successfully, distinctiveness pruning shows promise as we were able to successfully remove up to 16 neurons from the original 100 neurons without sacrificing model accuracy on any significant level. Therefore, distintinctive pruning seems to be an effective method of network reduction. However, given the poor performance of all our models (which performed approximately at the rate of random chance), these results should be taken with a grain of salt. In particular, this dataset was summarised from observational data, and we used a different data separation technique due to the omission of participant information. These changes could have significantly impacted the mdoel's learning. Further, this study only looked at a basic, feed-forward neural network. Therefore, future research may want to apply the leave-one-participant-out cross validation method used in the original paper of the dataset, or try to determine effects of distinctiveness pruning and c-means augmentation on other model types such as CNN or LSTM.

5 References

Angst, J. & Merikangas, K., 1997. The depressive spectrum: diagnositc classification and course. *Journal of Affective Disorders,* 45(1-2), pp. 39-40.

Bezdek, J. C., 1981. *Pattern Recognition with Fuzzy Objective Function Algorithms.*. s.l.:Springer.

Dunn, J., 1973. A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters. *Journal of Cybernetics,* 3(3), pp. 32-57.

Gedeon, T., 1996. Indicators of Hidden Neuron Functionality: the Weight Matrix versus Neuron Behaviour. *Australasian Journal of Intelligent Information Processing Systems,* 3(2), pp. 1-9.

Gedeon, T. & Harris, D., 1991. Network Reduction Techniques. *Proceedings International Conference on Neural Networks Methodologies and Applications,* Volume 1, pp. 119-126.

Kuźniar, K. & Zając, M., 2015. Some methods of pre-processing input data. *Computer Assisted Methods in Engineering and Science,* Volume 22, pp. 141-151.

Miyamoto, S. & Umayahara, K., 2002. Methods in Hard and Fuzzy Clustering. *Soft Computing and Human-Centered Machines,* pp. 85-129.

Paykel, E., 2008. Basic concepts of depression. *Dialogues in Clinical Neuroscience,* 10(3), pp. 279-289.

Tran, T., Lee, T. & Jong-Suk, K., 2020. Increasing Neurons or Deepening Layers in. *MDPI.*

Zhu, X., Gedeon, T., Caldwell, S. & Jones, R., 2019. Detecting emotional reactions to videos of depression. *2019 IEEE 23rd International Conference on Intelligent Engineering Systems (INES),* pp. 147-152.