Comparison of Deep Bidirectional Neural Network Depths in the Limit Between Neural Network Classification and Class Prototype Cluster Classification Accuracy

Llewelyn Goodall

U6673089@anu.edu.au

Australian National University, Research School of Computer Science, Canberra, Australia

Abstract. Deep neural networks are one of the most powerful modern methods for classification when dealing with large quantities of data. Bidirectional neural networks have the ability to classify in two directions, which can aid training accuracy and be used to generate class prototypes as cluster centroids. Electroencephalograms are one of the most powerful ways to detect brain activity, registering numerous data points for patients who are stressed and those who are calm. Using deep bidirectional neural networks to classify these patients based on the electroencephalogram data, few hidden layers classified with higher accuracy, reaching 99.7% accuracy with three hidden layers using SiLU activation functions. Using the deep bidirectional neural networks to produce class prototypes as cluster centroids for cluster classification, less hidden layers performed better, with 95.8% accuracy with one hidden layer using ReLU activation functions. SiLU-based networks consistently produced better neural network predictions, but the inferior ReLU-based networks produced better class prototypes for clustering.

Keywords. Bidirectional neural networks, deep learning, class prototypes, cluster classification, electroencephalograms.

1 Introduction

1.1 Deep Bidirectional Neural Networks for Classification

Neural networks are a very powerful classification method that are capable of taking in large quantities of seemingly arbitrary data to combine the data points into features used to classify the data [1,2]. Past study with using neural networks for classification of emotions and brain signalling achieved accuracy greater than 99% [1,2].

However, even in neural networks there are numerous options when it comes to creating the model. Neurons in the brain can send signals in two directions [3], and this functionality is best modelled by the bidirectional neural network. Bidirectional neural networks (BDNNs) have the advantage that training can work in both directions, allowing for a range of additional opportunities. The ability to reinforce training by training in both ways helps to ignore noise and reduce overtraining, as the reverse passes ensure that additional noise in the forward pass does not have too extensive an effect, the model is biased towards the expected outputs of the classifications [4].

Neural networks have also seen extension through deep learning, the addition of more layers of hidden neurons for increased feature detection, with each layer finding higher levels of abstraction for features [5,6,7,8]. This can involve the use of different sorts of layers for different tasks and functions, and deep neural networks are now used in modern-day problem solving due to their increased power and functionality [5,6,7,8]. Bidirectional neural networks can also be given more hidden neuron layers, giving them both the capabilities of bidirectionality and deep learning.

1.2 Deep Bidirectional Neural Networks for Cluster Centroid Generation

One of the greater advantages of BDNNs however comes after training, as it allows for twice the functionality of a regular feed-forward network, as classification can work in both directions. This allows for search problems to be greatly quickened, as instead of searching for a solution, the solution can be put through the network in reverse to find the required input [4,9]. It also allows for situations in which data is often passed in both directions to work in a single neural network, such as translation between two languages [9,10]. A less common use is using individual classes to be reverse-passed through the BDNN, producing class prototypes that can be used as cluster centroids [3,11].

Bidirectional neural networks do this by producing what is seen as the most correlated input values for a classification label [9,12]. These exemplary values held by the class prototype can function as a cluster centroid, and can then be used for cluster classification based on similarity [9,12]. The accuracy of these cluster centroids is dependent on the ability of the bidirectional neural network to produce class prototypes the are appropriately positioned. As deep neural networks are typically more powerful than standard neural networks [5,6,7], it is possible that deeper bidirectional neural networks may produce better cluster centroids as they predict more accurately.

1.3 Electroencephalograms

Electroencephalograms (EEGs) monitor brain activity. They do this through the use of multiple, typically 14, electrodes that detect the electrical signals sent between neurons in the brain [13]. EEGs are best used for detecting epilepsy and other forms of seizures, but development in the technology allows them to have enough power to detect even small changes that correspond to changes such as stress [13].

Stress is a physiological response to adversity, characterised by numerous bodily responses [14]. Different causes of stress cause specific biological responses, though through the triggering of the body's autonomic nervous system, many of these responses are shared, and many of result in a change in brain activity [15].

Experiencing ongoing stress is a health risk, capable of causing bodily harm through numerous conditions, as well as contributing to impairment of daily functioning [14,15]. In modern times, stress is becoming more and more commonplace in society, so it is becoming more and more important to be able to recognise it so that it can be rectified. EEGs are capable of detecting the physiological changes in the brain when stress is present, but is unable to classify it [11,16].

1.4 Modelling Stress Classification Using Deep Bidirectional Neural Networks

One issue encountered with cluster classification is that cluster classification on its own is inherently unsupervised [11,17,18]. However, when labels are already present and we wish for distance-based supervised learning, a supervised distance-based algorithm, for example, Nearest Neighbours, can be used, or we must find a different way to generate a supervised cluster centroid [17]. BDNNs have been shown in the past to be capable of using supervised data to create cluster centroids, and this can solve the issue of finding wanted clusters [9,13,15].

The ability to create accurate cluster centroids using class prototypes from deep bidirectional neural networks can be greatly beneficial as cluster classification has a much lower computational cost than a deep neural network. If the accuracy of both are comparable, then cluster classification using these class prototypes can greatly decrease future computation time. It is unlikely that they will be comparable given their relative power and past studies [9,10,11], but the class prototypes as cluster centroids could still be useful for quick preliminary classification.

However, deep neural network classification and cluster classification have fundamental differences. Standard cluster classification associates cluster centroids with data modes [18]. Deep neural networks associates combinations of inputs to create features [8]. These features individually correspond to their own small clusters, but not necessarily to a single large cluster, as this would be more indicative of a single feature. As such, this could impose limits on how the increasing depth of a network improves both the neural network classification and using the class prototype for cluster classification.

As such, the aim is to discover if increasing the depth and accuracy of a bidirectional neural network increases the accuracy of both neural network classification and cluster classification using class prototypes as cluster centroids, or if the two can only be optimised separately.

2 Method

2.1 Data and Preprocessing

The dataset used was a large quantity of already pre-processed data from an EEG brain scan. An EEG measures voltage over time for each electrode [13]. The dataset is a set of measurements of these voltage over time recordings, looking at the mean, minimum, maximum, standard deviation, variance, interquartile range, skewness, root mean square, summation, Hjorth parameter, Hurst exponent, means of the first and second differences of the signals, approximate entropy and fuzzy entropy of each electrode (210 data points total). This gives an overly detailed numerical description of the EEG recordings. Each patient had the process performed on them six times, three times calm and three times stressed, to get a better range of values for the data.

For the classification labels, as calm is only applied as a single class/label, stress ought to be a single class to maintain equality of class size to improve classification accuracy despite the many causes and types of stress [15,19].

The values themselves for the data depended on the electrode, as each electrode would measure voltages of different sizes relative to each other. Furthermore, different numerical descriptors have inherently different magnitude, for example, mean value and standard deviation, even if they didn't have different levels of importance. These mismatched magnitudes and importance are less of an issue for neural networks as their weights allow for them to compensate, but given the intent to use the data for cluster centroids using distance-based comparison, unequal relative magnitudes would give some values much more weight than they ought to have. As such all values would need to be normalized.

Because neural networks use multiplication of values and weights, normalizing between 0 and 1 doesn't work. A reverse pass of 0 means that all values will be 0, meaning proper classification of 0, which classifies for calm, would not be feasible. As such the normalization was over -1 to 1, using MinMax scaling.

Given the large set of potential inputs as we have numerous measurements from each electrode, the computational complexity using all of them would be very high and in danger of overfitting or possibly have unnecessary noise. Feature selection is used to fix both of these issues. For neural networks, past study has shown that genetic algorithms are one of

the most powerful methods of feature selection for neural networks [1]. However, given the goal to use class prototypes as cluster centroids, the issue is that genetic algorithms attempt to search for single feature subspaces, but clusters tend to be comprised of numerous subspaces that overlap [20]. This feature is exacerbated by the use of only two cluster centroids, each cluster has lots of subspaces that overlap. As such K-Best feature selection was used, as this doesn't search as strongly for a single feature subspace [20].

Training and testing data set splits were randomised around a 0.8/0.2 split, as well as shuffling the rows of data before each use of the data to ensure randomisation of data to gain better results.

2.2 Bidirectional Neural Network Design

The neural networks were constructed in Python using the Pytorch library. Each was constructed with the same number of input neurons, and each hidden layer has the same number of neurons in that layer as each other neural network in order to keep the capabilities of each model the same at each layer. Deeper neural networks of course have more layers, with each layer having less neurons than previous layers (unless rounding brings them to be the same) in an attempt to combine features. The models made had one, three, five or eight hidden neuron layers.

Due to the binary classification output, the options for loss functions were greatly cut down, as many loss functions require multiple output neurons to cross-analyse their loss. As such the simple function of mean squared error loss was used. The loss of power here however is not an issue, as given data that follows a Gaussian distribution, the mean squared error is the cross-entropy between the labels and the predictions [21].

The optimiser chosen was the Adam optimiser. Though designed for deep learning, it is still one of the most powerful optimisers available for general neural network use. Adam uses stochastic gradient descent and was designed around being both intuitive and effective [22]. As such, the hyper-parameters were left in their default values so that the optimiser itself could determine them.

In past experiments, using ReLU as the activation function for the bidirectional neural networks showed decent results when using the class prototypes as cluster centroids [9,12]. Another recent development was the advent of the Swish activation function, known as SiLU in Pytorch, which performs better than ReLU for neural network classification [23]. Given the aim is to differentiate between models that are accurate in cluster classification, neural network classification, or both, the different models used ReLU or SiLU to produce these differences in classification.

2.3 Bidirectional Neural Network and Cluster Centroid Classification Testing

The data was split into training and testing sets, with the split having a randomised value centred around 0.8/0.2. All of the bidirectional neural networks were then trained over the same split. The final results were gathered by training them over 500 epochs.

The testing then had the neural networks attempt to classify the testing data. The outputs for the networks were within the range of -1 to 1, but not exactly -1 or 1, which were the labels, so negative outputs were rounded to -1 and positive outputs were rounded to 1. These rounded values were then compared against the true labels to find the accuracy of the networks.

Using a reverse pass on the bidirectional neural networks for both classifications of -1 and ,1 the values for the class prototypes were found and stored.

For each row in the testing set, the absolute difference of each element in the row with its respective element in the -1 classification was summed together, and then repeated with the 1 classification. The testing input was then classified according to which prototype the absolute difference was smaller.

This process was then repeated 500 times on the ReLU BDNNs, with the data shuffled each time and the training-testing split randomised each time, to gather an average result. It was then repeated another 500 times in the exact same manner on the SiLU BDNNs.

3 Classification Results

	One Hidden Layer	Three Hidden Layers	Five Hidden Layers	Eight Hidden Layers
NN ReLU Classification Accuracy	99.5%	99.5%	99.1%	94.6%
NN ReLU Classification Std. Dev.	1.6%	1.8%	2.2%	13.2%
NN SiLU Classification Accuracy	99.6%	99.7%	99.4%	99.1%
NN SiLU Classification Std. Dev.	1.5%	1.5%	1.8%	2.2%

Table 1. Classification results from EEG data using the bidirectional neural networks with the specific numbers of hidden layers and activation functions. Results are averaged from 500 tests.

	One Hidden Layer	Three Hidden Layers	Five Hidden Layers	Eight Hidden Layers
ReLU Cluster Classification Accuracy	95.8%	89.5%	85.4%	67.0%
ReLU Cluster Classification Std. Dev.	5.0%	9.1%	12.8%	15.4%
SiLU Cluster Classification Accuracy	86.7%	85.7%	85.5%	86.1%
SiLU Cluster Classification Std. Dev.	6.2%	7.9%	8.4%	9.5%

Table 2. Classification results from EEG data using cluster classification, in which the cluster centroids were created using bidirectional neural network models with the specified numbers of hidden layers and activation functions. Results are averaged from 500 tests.

For neural network classification, the models using SiLU as their activation function outperformed those using ReLU at all depths. The model with three hidden neuron layers using SiLU had the highest classification accuracy and the lowest standard deviation of all neural network classifiers.

For cluster classification using class prototypes as cluster centroids, the bidirectional neural networks that used ReLU activation functions outperformed those that used SiLU when comparing models with one hidden layer and three hidden layers, were roughly equivalent at five hidden layers, and SiLU performed better at eight hidden layers. The greatest accuracy was achieved by the model with one hidden layer using ReLU at 95.8% accuracy, and also had the lowest standard deviation at 5.0%.

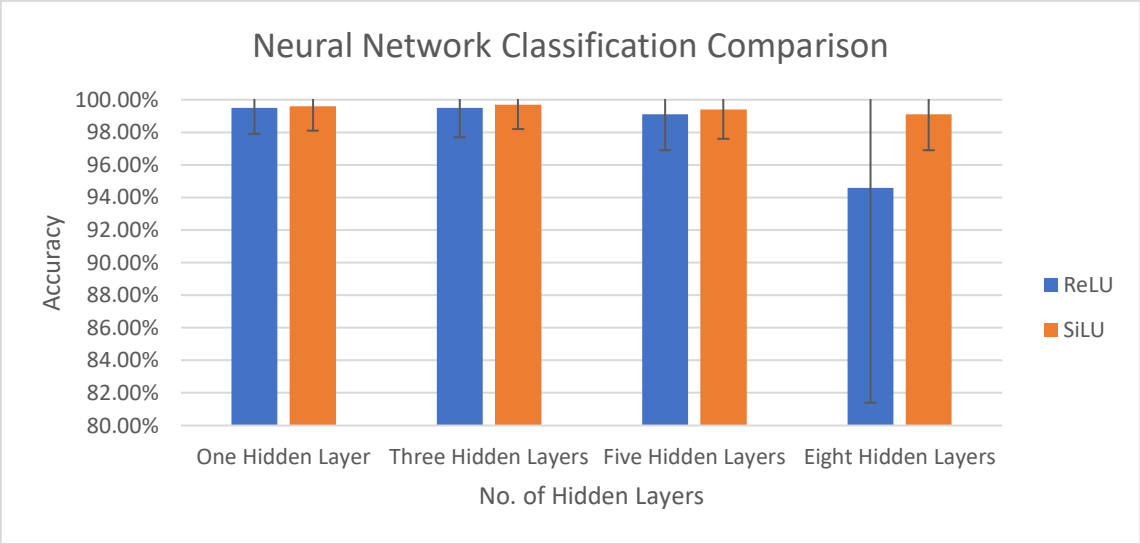

Figure 1. Classification accuracy of the bidirectional neural networks of varying depths, with the number of hidden neuron layers and activation functions as shown. Results were gathered from 500 tests, and the error is one standard deviation to either side.

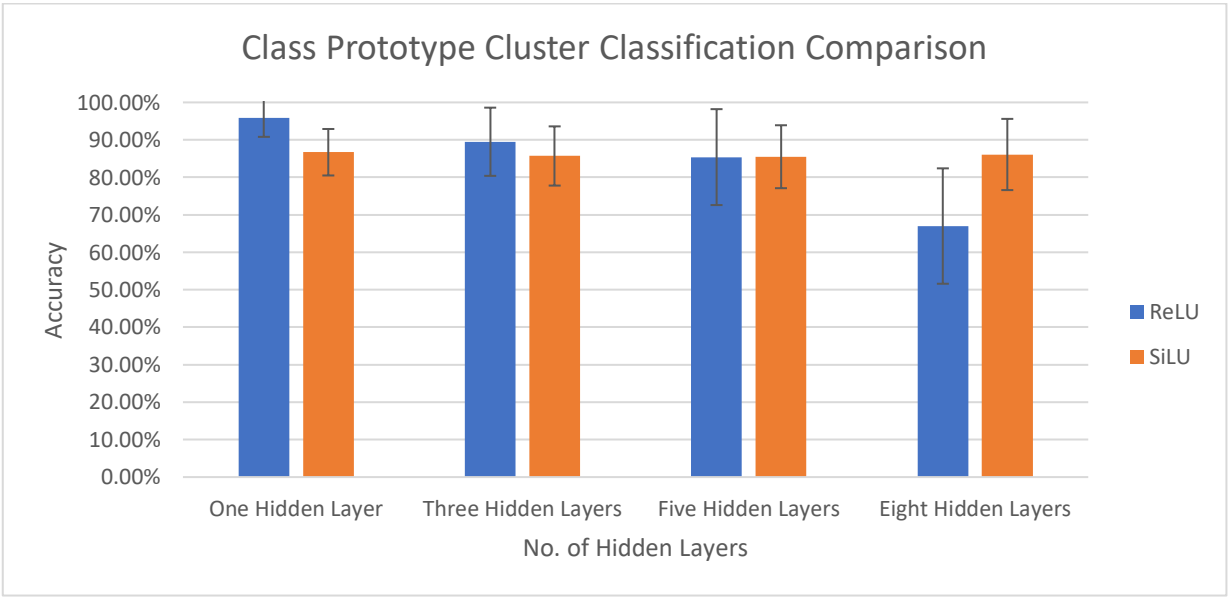

Figure 2. Classification accuracy of cluster classification, using class prototypes generated by deep bidirectional neural networks with depths and activation functions shown as cluster centroids. Results were gathered from 5000 tests, error is one standard deviation.

4 Discussion

4.1 Comparison of Depths in Bidirectional Neural Network Classification\

From Figure 1, we can see that the bidirectional neural networks using the SiLU activation function outperformed those using the ReLU activation function, especially in the deeper networks. The networks using SiLU all achieved classification accuracies greater than 99%, with the mode using three hidden neuron layers achieving the highest classification accuracy of all models. They also had very low standard deviations, showing that the models were consistently accurate.

In the networks using the ReLU activation function, the models with one and three hidden neuron layers performed the best, with 99.5% classification accuracy. This is not dissimilar from those using SiLU, though, they had a higher standard deviation, showing worse consistency. The model with five hidden neuron layers still performed well, but started to show signs that added layers decreased accuracy. The model with eight hidden neuron layers performed relatively poorly.

With both activation functions, performance began to worsen at the point of having five hidden neuron layers, and the worst results were achieved when using eight hidden neuron layers. Already it can be seen that something harmful is being done to the model by adding a large quantity of layers.

The first possibility is that the addition of further hidden layers is not helpful, they do not add to the capabilities of the model. This is similar to the idea of robust and critical layers, in which some layers can be removed without significant effect on the model, robust layers, whereas removal of some layers cause the model to instantly approach the accuracy of random guesses, critical layers [21,24]. Past study has shown that in some deep learning models, many of the later layers tend to be robust and many of the earlier layers tend to be critical [24]. One such explanation for the results for the five hidden layer models shown in Figure 1 is that additional layers are simply not adding enough functionality to show improvement and simply lead to overfitting.

Deeper networks also run a greater risk of overfitting training data, and this overfitting can then lead to poor performance on testing data [5,6,8,21]. This can cause deeper models with more layers to perform worse than models with less hidden layers, as the overfitting of training data harms the performance of the model relative to the shallower models. This is another possibility for the lower classification accuracy of the deep bidirectional neural network with eight hidden layers.

4.2 Comparison of Depths in Bidirectional Neural Networks for Cluster Centroid Generation

From Figure 2, in contrast to using the neural networks themselves for classification, when using the networks to generate class prototypes as cluster centroids for cluster classification, the models that used ReLU, for the most part, outperformed those that used SiLU. Those that used SiLU were consistent around the 86% accuracy mark, whereas those that used ReLU declined sharply declined. The model with one hidden layer using ReLU had the highest cluster classification accuracy at 95.8%. The ReLU model with three hidden layers still outperformed SiLU, having 89.5%, the model with five hidden layers was largely on par, and the model with eight hidden layers performed the worst, with a very low accuracy and high standard deviation.

The common trend is that more layers corresponds to decreased accuracy when using cluster classification. This could be explained by the larger networks simply being less accurate, as the five and eight hidden layer models have already shown to be worse at classification, and if they cannot do that accurately, it is unlikely they will be able to generate accurate cluster centroids.

Another explanation for decreased accuracy is that deep neural networks are looking for high-level features that are built out of low-level features that are built out of input data [8,21]. This leads to multiple smaller clusters being looked for in deeper networks rather than one large cluster [21]. The small clusters belonging to each classification, stressed or calm, can then overlap with each other, and attempting to use a single large cluster centroid causes the two created clusters to encapsulate small, actual clusters that may belong to the opposing label [18,21]. This limits the ability of the deeper networks as they constrict the individual clusters they are searching for too much, they become too small and too numerous, and the class prototype cannot account for all of these individual, small clusters.

4.3 The Inverse Relation of Neural Network Classification and Class Prototype Cluster Classification Accuracy

The important phenomenon present is that the more accurate neural network, the models that use SiLU, create a less accurate cluster centroid via class prototype than the less accurate ReLU neural network. Intuition would have us believe that a more accurate bidirectional model would lead to a more accurate prediction of input values. However, there are limiting factors to this, causing this intuition to be naïve and incorrect.

Use of a single cluster centroid for a single label means that the experiment assumes that, for that label, the input vectors are unimodal [18]. If each input vector is not unimodal, then the clusters produce too much overlap [18]. However, in classifying EEG data with neural networks, there is no guarantee that the data will conform to this assumption. In fact, deep neural networks perform just as well when data does or does not conform to this mono-modal form as they work to identify features, combinations of values of inputs, not the unrelated presence of some few individual inputs [8,21].

So it appears there is a limit to a singular model that aims to maximise both neural network classification and class prototype-based cluster classification. A model which predicts inputs to appear more as one cluster loses its ability to finely differentiate features [8,21], and a model which can finely differentiate between features loses its ability to create clusters without significant overlap [18]. Models which choose to prioritise one must then sacrifice the ability to significantly improve in the other.

This may not appear to be the case looking at the ReLU model with eight hidden layers, as it has the lowest classification accuracies for both neural network classification and cluster classification. However, this is due to it simply being a poor model. The inverse relation most likely holds true for well optimised models.

From Table 1, we can also see that shallower BDNNs provide better cluster classification with only minimal decrease in neural network classification. This is likely due to the decreased feature specificity looked for in shallow neural networks, there are less layers through which the searching of input data becomes more high-level feature-specific [8,18,21\]. The search space of the neural network more closely approximates that of a cluster.

4.4 Further Study

Given that it is the differences in the shapes of the clustering that separates cluster classification and neural network classification, limiting the ability of a single bidirectional neural network to both classify itself and create class prototypes as cluster centroids, changing the data could be the way to enable this capability. Further study could look at data pre-processing that allows for use of bidirectional neural networks this way, or perhaps more feasibly, using bidirectional neural networks and their results to allow for data post-processing. Data processing is a very powerful tool for improving any use of the data itself, and given the power of neural networks, being able to harness this computational power could greatly enhance this useful tool.

5 Conclusion

In this paper we showed that while bidirectional neural networks are capable of both neural network classification and the creation of cluster centroids for cluster classification, the intrinsic workings of cluster classification and neural network classification have differences that limit bidirectional neural networks from improving both methods of classifications simultaneously. All data was gained from using electroencephalogram brain readings of stress. These readings were then normalized and the 30 most relevant features were extracted and used for the findings. Using bidirectional neural networks, models were created that reached very high neural network classification accuracy (up to 99.7%), and different models that reached high cluster classification accuracy (up to 95.8%).

Though limited, due to the power of bidirectional neural networks, this limit is still very high, with the highest scoring cluster centroid producing network also having a neural network classification of 99.5%. It showed that shallower bidirectional neural networks tend to optimise the ratio of cluster to neural network classification accuracy. The shape of the data and the differences in reading this data are what ultimately limit this dual-functioning of bidirectional neural networks.

6 References

1 Rahman, J. S., Gedeon, T., Caldwell, S., Jones, R., Jin, Z. *Towards Effective Music Therapy for Mental Health Care Using Machine Learning Tools: Human Affective Reasoning and Music Genres*. JAISCR 11(1):5-20

2 Anderson, B., Montgomery, D. *A method for noise filtering with feed-forward neural networks: Analysis and comparison with low-pass and optimal filtering*. 1990 IJCNN.

3 Irani, R., Nasrollahi, K., Dhall, A., Moeslund, T. B., Gedeon, T. *Thermal Super-Pixels for Bimodal Stress Recognition*. 2016 Sixth International Conference on Image Processing Theory, Tools and Applications, 2016.

4 Wade, J. J., McDaid, L. J., Harkin, J., Crunelli, V., Scott-Kelso, J. A. *Bidirectional Coupling between Astrocytes and Neurons Mediates Learning and Dynamic Coordination in the Brain: A Multiple Modeling Approach*. PLoS ONE 6(12), 2011.

5 Kussul, N., Laverniuk, M., Skakun, S., Shelestov, A. *Deep Learning Classification of Land Cover and Crop Types Using Remote Sensing Data*. IEEE Geoscience and Remote Sensing, 2017.

6 Chen, Y., Zhao, X., Lin, Z., Gu, Y. *Deep Learning-Based Classification of Hyperspectral Data*. IEEE Journal of Selected Topics in Applied Earth Observation and Remote Sensing, 2014.

7 Qi, C. R., Su, H., Mo, K., Guibas, L. J. *PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation*. CVPR Computer Vision Foundation, 2017.

8 LeCun, Y., Bengio, Y., Hinton, G. *Deep Learning*. Nature, 521:436-444, 2015.

9 Nejad, A. F., Gedeon, T. *Bidirectional Neural Networks and Class Prototypes*. Proceedings of ICNN'95 – International Conference on Neural Networks, 1995.

10 Schuster, M., Paliwal, K. K. *Bidirectional recurrent neural networks*. IEEE Transactions on Signal Processing 45(11), 1997

11 Gan, H., Huang, R., Luo, Z., Xi, X., Gao, Y. *On using supervised clustering analysis to improve classification performance*. Information Sciences 454:216-228, 2018.

12 Goodall, L. *Using Bidirectional Neural Network Class Prototypes as Cluster Centroids for Classification*. 4th ANU Bio-Inspired Computing Conference, 2021.

13 Britton JW, Frey LC, Hopp JL, Louis EK, Frey LC. *Electroencephalography (EEG): An Introductory Text and Atlas of Normal and Abnormal Findings in Adults, Children, and Infants*. Chicago: American Epilepsy Society; 2016.

14 Pearlin, L. I., Menaghan, E. G., Lieberman, M. A., Mullan, J. T. *The Stress Process*. Journal of Health and Social Behaviour 22(4):337-356, 1981.

15 Kemeny, M. E. *The Psychobiology of Stress*. Current Directions in Psychological Science 12(4):124-129, 2003.

16 Jackson, M. *The stress of life: a modern complaint?* Lancet 383(9914):300-301, 2014.

17 Finley, T., Joachims, T. *Supervised Clustering with Support Vector Machines*. Proceedings of the 22nd International Conference on Machine Learning, 2005.

18 Hartigan, J. A. *Statistical Theory in Clustering*. Journal of Classification 2:63-76 (1985).

19 Chen, J. J., Tsai, C. A., Young, J. F., Kodell, R. L. *Classification ensembles for unbalanced class sizes in predictive toxicology*. SAR and QSAR in Environmental Research 16(6):517-529, 2005.

20 Liu, H., Yu, L. *Toward Integrating Feature Selection Algorithms for Classification and Clustering*. Arizona State University Department of Computer Science and Engineering.

21 Goodfellow, I., Bengio, Y., Courville, A. *Deep Learning*. MIT Press, 2016.

22 Kingma, P. D., Ba, J. *Adam: A Method for Stochastic Optimization*. 3rd International Conference for Learning Representations, 2015.

23 Ramachandran, P., Zoph, B., Le, Q. V. *Searching for Activation Functions*. Google Brain Residency, 2017.

24 Ba, L. J., Caruana, R. *Do Deep Nets Really Need to be Deep?*. University of Toronto, 2013.