Pruned Neural Networks: Predicting Visualisation Interfaces by Pupil Data

Jonathan Roberts

Australian National University, CECS, North Road. 108,
2601 ACT, Australia
{Jonathan Roberts} u5372418@anu.edu.au

Abstract. Neural Networks have the potential to predict the visualisation an observer is viewing using eye tracking data. Pruning techniques can be applied to the Neural Network in order to improve the performance of the network. This paper looks at a two-fold problem presented by these ideas. Firstly, to determine if Artificial Neural Networks can correctly classify visual interfaces from eye tracking data. Secondly to determine if a distinctiveness pruning technique impacts the model's classification performance. The paper found that Neural Networks achieved similar results to previous research regarding visualisation classification. It also determined that there is no significant impact on accuracy of the classification when using the distinctiveness pruning technique.

Keywords: Convolutional, Neural Networks, Visualisations, Network Pruning.

1 Introduction

Visualisations can aid observers in learning concepts or understanding information [1]. An observer's level of comprehension can change based on the layout and design of the visualisation [2]. The next challenge comes in measuring the effectiveness of a visualisation by determining how observers view and understand the displayed information. Technologies such as eye tracking have been valuable in determining these concepts, such as the observer's ability to gather information in a visualisation [3].

This technology can be applied to understand observer's visual patterns while they are being asked questions that require observers to perform information searches over the visualisation [4]. Eye tracking has the potential to help researchers understand more about the behaviours of an observer. While there are many different aspects of eye tracking data can be analysed, not every measurement may be applicable when analysing observers and visualisation types [4]. This can be important when trying to predict the visualisation from just eye tracking data. Data such as the observer's number of fixations and response time may provide more value than others when predicting what visualisations observers are viewing [4]. This prediction can be thought of as classifying the visualisations into distinct interfaces using the input eye tracking data.

There are a large variety of Artificial Neural Networks (ANN) that are capable of performing these binary classification tasks [5]. One common ANN used in binary classifications is the Multi-Layer Perceptron (MLP). The goal of a typical MLP in a binary classification is to take an input of values and attempt to correctly classify each set of values into one of the two targets. The MLP can be improved in various ways, such as implementing different and more complex variations of a typical ANN [6].

One example of a more complex ANN is a Convolutional Neural Network (CNN). There are many different uses for CNNs, including language analysis, and image classification [7]. They have also been shown to be beneficial when performing binary classification over time series data [8]. This suggests that a CNN has the potential to improve binary classification if provided with the correct data.

However, an alternative to a new network is to use the same network but implement a pruning technique over the network [9]. One of these pruning techniques is called distinctiveness pruning, which is used to find similar and complementary neurons and prune them down [10]. Distinctiveness calculates the cosine similarity between hidden neuron outputs and either merges or removes them based on their angular separation. If their angular separation is low, they are considered too similar and the weights of the second are added to the first and the first is removed. When the angular separation is very high, they are considered complementary and both are removed. The primary purpose being to prune down unnecessary neurons [9].

The first aspect that this paper seeks to discuss is the usefulness in applying various ANN models to classify visualisations from observer data. Additionally, to explore this idea with deep learning models and applying them to improve binary classification. This is to extend the conclusions of previous work regarding analysis on useful features when classifying visual interfaces [4]. This also attempts to explore the idea of deep learning models and applying them to binary classification. Secondly to investigate whether distinctiveness pruning has an impact on the network's ability to classify the visualisations. This seeks to extend research conducted on using distinctiveness in multiple approaches of network pruning [10].

2 Method

2.1 Prediction Task

The prediction task for this paper was to determine if a Neural Network (NN) could accurately predict a binary classification using data provided by a previous eye tracking research experiment [4]. It contained both the raw pupil data measured by the eye tracking software, and enriched information derived from this raw pupil data. The previous experiment involving two visualisations displaying information to an observer (radial and hierarchical). In the experiment observers were asked questions, where they then had to perform their own information search over the visualisations in order to find answers to the questions. During this time, eye tracking software was used to track the eye movements of the observers and record the raw pupil data. It gathered N = 576 samples from participant's left and right eyes. This was combined and transformed into N = 288 samples for the derived information. However, this paper only used N = 287 samples due to a null sample in the derived information.

The first prediction task performed by the first model used a subset of the features present in the derived data. There were five initial features analysed for the experiment, observers' correct response rate (CR), response time (RT), number of fixations (NoF), saccades duration (SD) and fixation duration (FD). These parameters have a labelled column called 'interface' for either the radial or hierarchical visualisation. This was called the 'full 5 feature' data set in the NN implemented in this paper for model testing.

As the previous research states, only the response time and number of fixations parameters were able to differentiate between the two visualisations [4]. This was called the 'minimal 2 feature' data set in model testing. The first implemented model was tested using the full data set and the reduced minimal data set to compare performance. There was a third data set created from initial testing called the 'partial 3 feature' data set that contained the three features unique to the full data set that the minimal data set did not have. This was used as further comparison.

The second prediction task used the initial 'raw' data to further expand on the first prediction task. This contained the original time series data of the pupil size of the participants left and right eyes. Each of the series of data has a label that specifies the 'interface' the observer was viewing during the entire series. The second neural network model used this data instead of the derived data.

2.2 Simple Multi-Layer Perceptron Design

The first model (the simple MLP model) implemented for the prediction was a Multi-Layer Perceptron (MLP) consisting of two layers (Fig. 1). The MLP contained an input, with a hidden layer and an output activation layer. The hidden layer used a RELU activation function, while the output layer utilized a sigmoid activation function. RELU was used for the hidden layer as it performed slightly better in initial testing. The output layer used a sigmoid activation function because it was attempting to predict a binary classification. Cross Entropy Loss was used as the loss function due to the use of sigmoid in the output.

This model consisted of two versions, the MLP base version and the MLP distinctiveness version. The versions were identical except that the distinctiveness version also implement distinctiveness pruning of the hidden neurons. Neuron distinctiveness was determined by calculating the cosine similarity between the hidden neurons' output vectors. Neurons with a similarity less than 15 degrees or more than 165 degrees were either combined or removed respectively. The same validation techniques for both version of the MLP model were used to measure performance improvements or degradations.

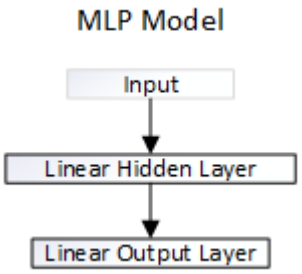

Fig. 1. Representation of the basic structure of the two-layer Multi-Layer Perceptron. The two layers are denoted by black boxes.

2.3 Convolutional Neural Network Design

The second model (the CNN model) implemented was a seven-layer Convolutional Neural Network (CNN) (Fig. 2). The first two layers were one dimensional convolutional layers that used ReLU as their activation functions. The convolutional layers used a kernel of five and a stride of two. The convolutional layers provide the feature extraction component of the model. After each convolutional layer there was a Max Pooling layer. Max pooling was used as it produced more consistent results when validating the initial model. Next was a Dropout layer that was used to increase generalisation of the network as the network was overfitting in training and produced much poorer results in the test phase. The final two layers are fully connected linear layers that provide the final classification. Like the simple MLP model, the first linear layer uses ReLU as its activation function, while the output linear layer uses sigmoid as its activation function. This again is due to the model trying to predict a binary classification. Cross Entropy Loss was again used due to the use of a sigmoid activation function.

Distinctiveness pruning was also implemented for the CNN model to create a second version of the model. The model was pruned after the first fully connected layer. This was done to allow for pruning similar to the simple MLP model as both models were pruned before the final linear layer.

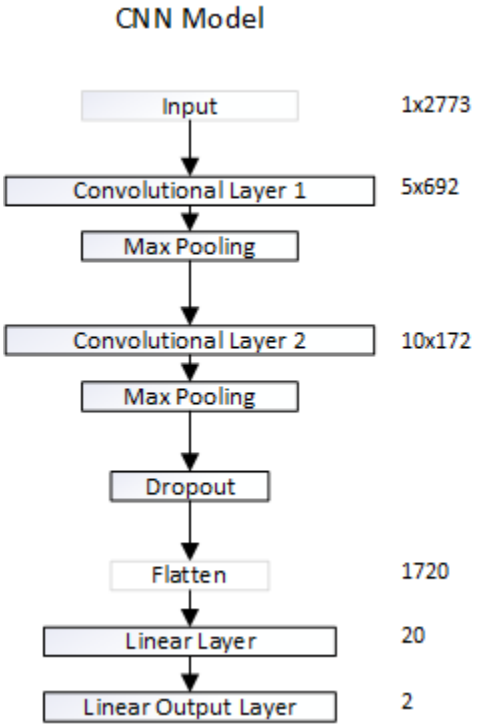

Fig. 2. Representation of the basic flow of a single sample through the Convolutional Neural Network. This describes the seven layers in black boxes with the layer output size next to each transformative layer.

2.4 Data Input and Pre-Processing

The input data for the derived data required certain pre-processing in order to fit into the MLP model. The original dataset was reduced to only contain the five parameters (full 5 feature), and their target interface classification. This was done to align with the initial goal of attempting to predict the interface classification to extend the initial experiment's analysis. The target classifications of radial and hierarchical was converted to a value of 0 and 1 respectively. This made it possible for the MLP to make a prediction of the target class. This data of five parameters and one target class was then further reduced to only contain the two parameters (minimal 2 feature), RT, NoF, and the target class. The remaining three parameters were put into their own dataset (Partial3 feature) to be used for further testing.

The raw pupil input data was also pre-processed in order to fit into the CNN model. The radial and hierarchical classifications were converted into 0 and 1, the same as the derived features data. The data was also transposed by the CNN model for easier use. The data was then populated by zeros where there was null data.

The all datasets were normalised after pre-processing to allow for better results in the models. Normalisation was done using the z-score normalisation formula **(1)**.

$$x = \frac{x - \bar{x}}{\sigma}. \quad (1)$$

Lastly, the data was shuffled for each step in the validation techniques. This was to make the data more stochastic in an attempt to improve the models' learning.

2.5 Model Validation

Performance of the simple MLP model was determined by implementing the Leave-One-Out Cross-Validation (LOOCV) method. The same validation process was performed over both the base and distinctiveness versions. LOOCV was chosen due to the small data samples provided by the initial data set so that the model could train over all the data. When using train/test splitting, it was found that the model had high inconsistency in the measured accuracy. Several different split methods were tested, including an increase or decrease of train/test split, as well as a random or fixed test data set. LOOCV was the validation technique that provided the most consistent result for the same parameters. The consistency was demonstrated in the low variance of the data in the results.

The CNN model was first validated using LOOCV similarly to the simple MLP Model. This was used to determine model hyperparameters and assess if the model was capable at the classification task. The final testing of the model was conducted using an 80% train, 20% test split of the data. This was done in order to determine the generalisation of the model as the CNN often overfitted on the data in training.

3 Results and Discussion

3.1 Simple MLP Model Accuracy

The aim was to classify the data sample as either a radial (0) or hierarchical (1) visualisation. The MLP model was run over ten iterations, for each data set and version. The coefficient of variation of the iterations were all 0.02 to 0.03, meaning that each of the runs had a very small variance with only 3% difference at most. This suggest that the LOOCV method was giving a consistent measurement of the accuracy of the models. This allowed for better comparison of the models and data sets.

Table 1. Results of classification for the simple model versions and input parameters after ten iterations.

Trial ID	Model Version	Data Set	Accuracy (%)	Std. D. (σ)
1	**Base**	Full 5 feature	60.07	1.22
2	**Base**	Minimal 2 feature	61.22	1.77
3	**Distinctiveness**	Full 5 feature	58.36	1.16
4	**Distinctiveness**	Minimal 2 feature	60.27	1.54

When looking at the accuracy of the model, it does not appear to be classifying the visualisations accurately. To compare this, take a thought experiment where a model generates a random value of 0 or 1. Instead of the model using the MLP to predict the radial or hierarchical targets, the random value is applied. The expected accuracy of this model would be approximately 50%, due to there being only two possible correct answers. The MLP models used here only improved on random chance by around 10%. One possible reason is that the MLP models are not very good at classifying the visualisations. Alternatively, the input features may not have a strong enough correlation to the visualisation targets. Both of these ideas were explored in a later section.

Even though the models did not predict the visualisations with high accuracy, the results were still analysed for any significant differences between the data sets and the model.

The first set of results compared was the individual models using each of the data sets. On a one tailed t-test, it showed that there was no significant difference between the two datasets for the base MLP model (trial 1 and 2); $t(10) = 1.61$, $p=0.063$. However, when comparing the results of the distinctiveness model (trial 3 and 4) using a one tailed t-test it did show a significant difference; $t(10) = 2.97$, $p=0.004$. This suggested that while the base model was not much better at classifying between data sets, the distinctiveness model improved significantly when using the minimal data.

The second set of results compared were the differences between the two models, for each of the data sets. A one tailed t-test for the minimal data (trial 2 and 4) showed that there was no significant difference between the two MLP models; $t(10) = 1.21$, $p=0.120$. When comparing the full data (trial 1 and 3) there was a significance in the difference of accuracy; $t(10) = 3.04$, $p=0.003$. This showed that when using the minimal data, the two models did not perform significantly differently, but when using the full data, the base model had significantly improved accuracy. The pruning appeared to have made the prediction worse when using data that had no correlation to the classification.

3.2 CNN Model Accuracy

The purpose of the Convolutional Neural Network (CNN) model was to determine if a more advanced deep learning technique could be used to more accurately predict the visualisation classification. Unlike the previous simple model, the CNN model used the raw data instead. This data was presented in a time series, where the eye tracking software recorded the pupil data over a period of time. The coefficient of variation of the iterations were 0.04 to 0.05, meaning that each of the runs had a small variance with 5% difference, slightly higher than the MLP model. This still demonstrated somewhat consistent results with each iteration.

Table 2. Results of classification for the CNN model and versions after 10 iterations

Trial ID	Model Version	Data Set	Accuracy (%)	Std. D. (σ)
5	**CNN Base**	Raw	80.49	3.22
6	**CNN Distinctiveness**	Raw	78.69	3.82

The results from the CNN were greatly improved over the results of the simple model. It was clearly seen that the accuracy of the CNN was around 20% higher than the accuracy of the simple model. The results were still compared for completeness.

The first comparison looked at was the difference between the simple model and the CNN model. It was shown in a one tailed t-test that the CNN base was significantly improved over the simple model with the minimal data set (trial 2 and 5); $t(10) = 15.73$, $p<0.00001$. Likewise comparing the CNN distinctiveness version with the simple model distinctiveness version (trial 4 and 6), there was a significant difference; $t(10) = 13.41$, $p<0.00001$. This demonstrated that the CNN model had a significant improvement over the simple MLP model.

The CNN was also implemented with a distinctiveness version. It had similar results to the simple MLP model, as there was no significant difference between the CNN's base and distinctiveness versions; $t(10) = 1.08$, $p=0.147$. This showed that distinctiveness pruning did not have a significant impact on accuracy.

3.3 Data Features and Classification

The two full and minimal datasets used here were done in order to compare the results from this research with the results of previous work done on visualisation classifications [4]. For each simple model version, the minimal data set did perform slightly better than the full parameter data set. However, there was only a significant difference in data sets for the distinctiveness model. The base model, while higher, was not significantly different. The distinctiveness model was similar to the previous research that reasoned that only the two features in the minimal data set, Response Time (RT) and No. of Fixations (NoF), had a strong correlation to the visualisation [4].

In contrast to the previous research [4], the full data set still provided greater prediction over none at all. This suggests there was some level of correlation between the original five features. However, as the two minimal features were present in both the full and minimal data set, it is possible that these were still the features that correlated with the visualisations.

This led onto an extended set of trials conducted using only the three features unique to the full data with the models which was called the 'partial 3 feature' data set. Each trial consisted of running the model over ten iterations as conducted previously. The coefficient of variation for accuracy in the iterations were 0.04 and 0.05 for the base and distinctiveness models respectively. This suggests, as previously, that the LOOCV method of validation again produced consistent results for each iteration of the trial.

Table 3. Extended trial for 3 features in simple model after 10 iterations

Trial ID	Model	Data Set	Accuracy (%)	Std. D. (σ)
7	**Base**	Partial 3 feature	54.76	2.02
8	**Distinctiveness**	Partial 3 feature	52.79	2.87

The accuracy of both models using the partial data demonstrates accuracy far below those of the previous two data sets. The accuracy of the partial data set is closer to that of a random choice than the original accuracy from both the full and minimal data sets.

When comparing the base model (trial 2 and 7) in a one tailed t-test, the minimal data set showed a significant improvement over the partial data set; $t(10) = 7.22$, $p<0.00001$. Similarly, the full data set (trail 1 and 7) was significantly improved over the partial data set; $t(10) = 6.76$, $p<0.00001$. When looking at the distinctiveness model, a significant improvement can be seen as well. The one tailed t-test for the minimal data set compared to the partial data set (trial 4 and 8) showed a significance; $t(10) = 6.90$, $p<0.00001$. Likewise did the full data set and the partial data set (trail 3 and 8); $t(10) = 5.41$, $p<0.0001$.

These results gave a clearer picture regarding the correlation between the features and the target visualisations. When the two features that are correlated to the visualisation [4] are removed, the model was much worse at predicting the visualisations. This suggests that the two features RT and NoF were significantly better at predicting the visualisations than the other features.

3.4 MLP Pruning

The distinctiveness versions of the models performed slightly worse than the base models on average. However, there was no significant difference between the models' base and pruned versions. Both the simple MLP model and CNN model demonstrated the same lack of significant difference in accuracy. There was only a significant difference when the full data was used with the simple MLP model. As previously demonstrated with the classification task, the full data set is likely to contain irrelevant data. A possibility is that the distinctiveness pruning method is removing neurons that are considered similar or complementary due to the presence of uncorrelated data.

The advantage of the distinctiveness model is that it has no significant loss in accuracy, while simultaneously having less neurons than the base model. This suggested that the pruning has the potential of creating a more efficient network with only a small loss in accuracy. This aligned with the concepts presented in the previous research regarding distinctiveness pruning [10].

4 Conclusion and Future Work

This paper aligns with some of the previous conclusions regarding the visualisation data set and the distinctiveness pruning technique. Firstly, it demonstrated that, while a simple Multi-Layer Perceptron (MLP) model was not very accurate at predicting the visualisation classifications, it still described the concepts presented by the previous research [4]. The previous research originally looked at five features gathered from eye tracking software and concluded that only Response Time and No. of Fixations where distinguishable between the visualisations. It was shown in this paper that the two features of Response Time and No. of Fixations have significantly more correlation to the target visualisations than the remaining three Correct Response Rate, Fixation Duration and Saccade Duration.

Secondly this paper concludes that distinctiveness pruning does not significantly reduce the accuracy of the model when pruning linear layers. The simple Multi-Layer Perceptron model and the Convolutional Neural Network (CNN) model both demonstrated that there was no significant difference between the accuracy. However, it potentially improves the model as it reduces neurons. This is also similar to what the previous work has shown [10].

Finally, it was shown that when using a more complex deep learning model, specifically a Convolutional Neural Network, that it is possible to predict the visualisation classification with an accuracy much higher than simple models. While the Convolutional Neural Network model did not predict with near 100% accuracy, it was easy to see the improvement the CNN model had over a more simplistic model.

This paper only looks at the pruning technique in a single layer with simple parameters in a specific linear layer. The technique could be analysed with more complex parameters or conducted over multiple and different layers. There is still room to analyse other measured attributes like network speed and efficiency instead of just accuracy. If a much larger data set is used, such as tens of thousands of samples, there could be further insight into the efficiencies of pruning to completement accuracy measurements. Additionally, the technique can be further explored in other Neural Networks than just the two presented in this paper.

Further work can also be conducted on delving deeper into the behaviour of the pruning. The distinctiveness pruning demonstrated a degradation when using the full data. Further analysis could be conducted on why it is that pruning uncorrelated data reduces the accuracy of the model.

References

1. Sedig, K., Rowhani, S., Morey, J., Liang, H.: Application of Information Visualization Techniques to the Design of a Mathematical Mindtool: A Usability Study. Information Visualization. 2, 142-159 (2003).
2. Majooni, A., Masood, M., Akhavan, A.: An eye-tracking study on the effect of infographic structures on viewer's comprehension and cognitive load. Information Visualization. 17, 257-266 (2017).
3. Calitz, A., Pretorius, M., Van Greunen, D.: The Evaluation of Information Visualisation Techniques Using Eye Tracking. International Conference on Computing and ICT Research. pp. 135-151. Fountain Publishers, Kampala (2009).
4. Hossain, M.Z., Gedeon, T., Caldwell, S., Copeland, L., Jones, R. and Chow, C.: Investigating differences in two visualisations from observer's fixations and saccades. Proceedings of the Australasian Computer Science Week Multiconference. 1-4. (2018).
5. Basheer, I., Hajmeer, M.: Artificial neural networks: fundamentals, computing, design, and application. Journal of Microbiological Methods. 43, 3-31 (2000).
6. Jeatrakul, P., Wong, K.: Comparing the Performance of Different Neural Networks for Binary Classification Problems. Eighth International Symposium on Natural Language Processing. pp. 111-115 (2009).
7. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature. 521, 436-444 (2015).

8. Brunel, A., Pasquet, J., Pasquet, J., Rodriguez, N., Comby, F., Fouchez, D., Chaumont, M.: A CNN adapted to time series for the classification of Supernovae. Electronic Imaging. 2019, 90-1-90-9 (2019).
9. Gedeon, T., Harris, D.: Network reduction techniques. Proceedings International Conference on Neural Networks Methodologies and Applications. pp. 119-126 (1991).
10. Gedeon, T.: Indicators of hidden neuron functionality: the weight matrix versus neuron behaviour. Proceedings 1995 Second New Zealand International Two-Stream Conference on Artificial Neural Networks and Expert Systems. pp. 26-29 (1995).