Analysis of Hidden Neuron Pruning for Reading Distractions Classification

Maojun Zhu,

Research School of Computer Science,
Australian National University,
u6170752@anu.edu.au

Abstract. Digital reading has raised its great significance in online learning. The capacity to evaluate the text readability under distractions is deemed to promote the effectiveness of learning. Although the reactions of readers to distractions may vary with the difficulties of readings and different reading contexts, it is still possible to use eye movements while reading to classify the language speaking of readers. This paper may emphasis to construct a basic neural network model to classify English Speakers with their reflections on Reading Distractions. Furthermore, the technique on pruning hidden neurons could be examined to check its performance that reducing model runtime with insignificant effects on the model performance. The evaluation should be compared with the initial neural network model on the accuracy and confusion matrix.

Keywords: Neural Networks, Reading Distraction, Distinctiveness Measure, Hidden Neurons Pruning.

1 Introduction

Many researches agree that the experience of digital reading is different from the experience of reading on printed materials. Digital Reading has been interpreted as distracting since readers could be frequently distracted by dynamic environments with massive advertisements and social media messages. As online learning has raised great importance and popularity, the approaches to evaluate the performance of readers in different demography on digital reading become significant. An experiment is introduced by Leana and Gedeon to collect the reflections of participants on distractions while reading. For their experiment, the fixations of eyes are detected with eye-tracker to record their eye movements under different difficulties of readings. They have interpreted that the differences in reactions to different reading difficulties are insignificant while there is a great difference in performance between the English speakers and other language speakers [1]. Inspired from the paper, we could examine the connections between readers' reactions to digital distractions and their language background.

The dataset collected by Leana and Gedeon includes 4 files of observation records. The reactions of 22 participants are recorded under 3 different distraction background for the first three files and a combination for the “FOR SPSS” file. The dataset enables great ease in understanding that majority of its features are numeric values. There is no data loss of dirty data is recorded to promote accuracy in training. However, there is a great challenge that only 66 samples are collected for the 22 participants. The training model may suffer its accuracy in overfittings because of insufficient training data.

We would like to construct a linear model on neural networks to train on the datasets for Reading Distractions. The neural network model is expected to be simplified in hidden neurons. For most neural networks, the number of hidden neurons is been drafted to in great size which may cause a great delay in the runtime of the neural network model. Pruning on neurons on similar functionalities are expected to have trivial effects in model performance while a great improvement to simplify the model. We may present our evaluations on neural network models with hidden neurons pruning.

2 Main Approach

2.1 Network Architecture

We have adopted the linear model which is trained with the back-propagation method. Back-propagation is the practice of fine-tuning the weights of a neural net based on the loss rate obtained in the previous epoch. The gradient of the error function is calculated with the neural network’s weights to update the weights for the next epoch. Increasing lower error rates are expected to be achieved with the tuning of the weights, and then making the model reliable by increasing its generalization.

However, the time required for the back-propagation method may be relatively high. Especially, many programmers may initialize the number of hidden neurons as a large number to prevent dropout of data in training. While we may find that the minimal number to train the model with similar performance would be much smaller than the initialized number of hidden neurons. These extra neurons are the main cause to reduce the training speed while making no improvements on the performance since they may have similar functionalities as existing neurons.

We have applied a network architecture with one hidden layer and the general expression of the prediction model is:

$$y(x) = Sigmoid\ ((f\ (x, w1), w2)$$

where $w1$ and $w2$ represent the weight and bias of the input layer and hidden layer respectively.

2.2 Data-Preprocessing

We have spilt the datasets into 2 fixed groups, the first 53 rows are considered as the training data and the rest are testing data. The common training dataset for different techniques could promote fairness in comparisons. The "L1/L2" feature is extracted out to be the target values and the rest features are input features. In the neural network, we have initialized the number of outputs as 2 which is either L1 or L2. We have assigned the number of hidden neurons the same number as the inputs which is 18 that we expect to minimize the value throughout the hidden neuron pruning.

We have done several techniques to transfer the datasets which may have promoted better performance in the training model. Firstly, we have dropped the overlapping columns such as condition where condition combine the information as "Text Type" and the second "Condition". We have dropped columns on objective reflections such as "Do you often use social media, email and/or instant message while you are reading course materials or work materials?". These data may introduce the objective bias in which participants are encouraged to mark themselves on the scale. The honesty of the participants could not be ensured. Secondly, we have changed the datatypes for "Text Type", "Condition" and "L1/L2" to numeric and "Time Taken" to seconds. Then, we could adopt normalizations to each of the input columns to arrange their values into the 0-1 range. As such, the initial weights assigned to these values are expected to not have a large score to overweight the rest at the start of training.

2.3 Pruning Technique

There are several assumptions for hidden neuron pruning. Neural network models are assumed to have three layers of neurons that are processed with feed-forward in the network. The connections between neurons are assumed to pass forward stepwise [2]. For every neuron in a layer, it should have a weighted connection to all the neurons in the previous layer.

The basic concept for neuron pruning is to remove the redundant neurons in the hidden layer [3]. In this paper, we would like to apply Distinctiveness analysis which evaluates the functionality of neurons. The neuron output activation vector is evaluated over the presentation pattern space to determine the distinctiveness of hidden neurons. A vector should be constructed for each hidden neuron with the same dimensionality as the number of patterns in the training set. Each component of the vector should represent the output activation of that neuron which collaboratively represents the functionality in input pattern space [4].

2.4 Distinctiveness Measures

We would like to apply the angle analysis between vectors in pattern space to judge the distinctiveness of hidden neurons. The similarity on outputs of hidden neurons is measured to evaluate their functionalities through this approach. Cosine similarity is calculated on the vectors of two hidden neurons to get distinctness angle. Given two vectors vA and vB with the same length, the angle between vA and vB is given by [5]:

$$\phi(v_A, v_B) = \cos^{-1} \frac{v_A * v_B}{\|v_A\| * \|v_B\|}$$

Then this angle is usually compared with a threshold angle to judge pruning decisions. We would like to initialize the threshold angle to be 15 degrees. If the angle between two hidden neurons less than 15 degrees, their functionalities are interpreted as too similar. Any one of the hidden neurons should be deleted and its weight and bias should be added on to the other hidden neuron. If the angle between two hidden neurons is greater than 165 degrees, their functionalities are interpreted as complementary which one's functionality is canceled by the other during the processing. Thus, both hidden neurons should be removed as well as their weights and biases.

2.5 Algorithm

Firstly, the neuron network model should be trained with the initial number of hidden neurons. The loss of data and accuracy should be reduced through each epoch via backpropagation. After all epochs, the training model is expected to adjust the weights and biases of its hidden neurons to promote its best accuracy in training. We would like to copy this final weights and biases of all hidden neurons. The cosine similarity should be taken to evaluate the similarities of the weights of hidden neuron outputs. If the weight vectors of two hidden neurons are evaluated as too similar or complementary, the input weights are adjusted correspondingly. This method should be mapped to every two hidden neurons in the training model. As a result, some of the hidden neurons may have their input weights to be reassigned to zero values. These hidden neurons are identified as redundant in functionalities.

We would like to remove the input weights with zero values and rearrange to new input weights which are named "new weights" as well as "new biases" for the input biases. The number of hidden neurons should be updated to the number of input weights left. Furthermore, a new linear neural network with the updated number of hidden neurons should be generalized to initialize its input weights. This initial input weights then should be overwritten by the "new weights" and "new biases". The same approach is applied to overwrite the output weights of the hidden neurons in the new model. The new linear model is expected to train with the reduced number of hidden neurons and the given parameters of hidden neurons.

At last, the new linear model is expected to generate its final weights and biases of hidden neurons. The same steps are adopted iteratively to minimize the number of hidden neurons and update their weights. The accuracy and loss of data are expected to decrease. When there are no vector pairs of hidden neurons with a distinctness angle beyond the threshold, the iteration is expected to cease on hidden neuron pruning. The left number of hidden neurons is expected to be the optimal solution to the training model.

3 Enhancements on Approaches

3.1 K-Fold Cross-Validation

The K-fold cross-validation is suggested on limited datasets. In this training model, there are only 66 data samples for training on 18 features which inevitably lead to overfitting. The general procedure is to shuffle original datasets randomly and partition them into k sub-datasets in equal length [6]. Each of the sub-datasets is expected to validate the combination of the rest sub-datasets. The reliability of testing accuracy is enhanced by averaging the performance with the k rounds of cross-validation on different training sets and validation sets. In this paper, the parameter k is set to 10 which inadequately leverages the testing accuracy with an acceptable decline in running speed. The split of the k folds is fixed to promote the fairness to comparisons on the neural networks with different parameters. Thus, the randomness of sub-datasets might not be considered to affect the comparison results.

3.2 Evolutionary Algorithms

Evolutionary algorithms are preformed since there are potentially redundant features for prediction in the training model, which could be applied to evaluate the minimum number of features used in training while with insignificant effects on testing accuracy. The remaining could be denoted as the sensitive features in training. This compression not only saves storage and calculation resource but contributes to a concise closed-form expression of the prediction model [7].

In evolutionary algorithms, a gene could be represented as the involvement of a feature which is represented as binary numbers. A chromosome is the combination of the involvements of all features in the training model. Fitness function to evaluate the chromosomes is on the least number of features involved in the training model and the relatively smaller accuracy difference from the training model which involves all features. By removing the non-sensitive features in training, we could minimize the features in neural networks [8]. As such, the limited amount of training data contributes to the fewer number of training features that will greatly reduce the scarcity in training datasets and training inefficiency.

3.3 Network Architecture

In this paper, a genetic model is applied to evaluate all possible combinations of features with a DNA size as the number of features in each chromosome. To evaluate the fitness of each chromosome, the training model is called to generate a new neural network with its number of input neurons reduced to the left number of features which counts 1s in the chromosome list. 10-Fold cross-validation is applied to enhance the reliability of the testing accuracy of each model. The evaluation of the fitness of individuals is based on the weighted sum of the extracted number of features and the testing accuracy to the model generated as a formula “fitness = average testing accuracy + the number of extracted features”. The score of each feature extraction is highly rewarded as 1 to encourage the minimal number of features in evolutions while a higher reward may backfire the testing accuracy of the training model. Further study to balance the weights of testing accuracy and the rewards for each extracted feature, is deserved to evaluate the training model with the optimal set of input features.

The population is initialized randomly which is expected to achieve a high diversity to prompt the optimal solution. The population with higher fitness value has a higher chance to be chosen in selection. Uniform crossover is taken to randomly pick up another parent to produce a child on random cross-over points. The mutation is initialized at a low rate as 0.002 on children which may adequately prevent the model to achieve its local minima rather than the global minima. When the child has a better fitness than its parent, it will be considered to replace its parent in the next generation. As such, the best individual in the final generation is expected to have the highest fitness in evolutions while it is not guaranteed to be the optimal solution.

4 Results and Discussions

The experiments will be taken on the pruning algorithms discussed above. In the set-ups, the initial number of hidden neurons is set to 18 which is expected to be minimized through the pruning process. The parameters of the training model may consist of 18 input features and 2 target outputs according to the datasets. The learning rate is set to 0.01 which is expected to update the model smoothly with backpropagations to converge to the optimal model. The number of epochs is set to be 500 for the training that further backpropagations on more epochs are observed with insignificant values to testing accuracy while collectively decrease the running speed in experiments. The training model is set with Cross-Entropy as the loss function and Adam as the optimizer.

4.1 Neuron Pruning Experiments

In this experiment, the neuron pruning technique is applied once on training models. The subsequent network is set-up with a reduction on the “number of neurons to prune”. The purpose is to observe the changes in testing accuracy with different numbers of hidden neurons in training. “The accuracy and loss” of data is to show the impact of a smaller number of neurons on the training performance. “The number of neutrons to prune” represents the number of the zero values in the output weights of hidden neurons. “The accuracy of testing” is evaluated if the updated hidden weights and biased are put back to the linear model to illustrate the impact of pruning on model performance. This accuracy represents the expectations in performance when the current number of hidden neurons is reduced with the number of neurons to prune.

Table 1.

Number of Hidden Neurons	Accuracy/Loss (Training)	Number of Neurons to Prune	Accuracy after Weights Update (Testing)
18	100%/0.0096	8	61.54%
10	100%/0.0138	1	61.54%
9	100%/0.0174	1	61.54%
8	100%/0.0221	4	69.23%
4	*100%/0.0480*	0	69.23%
2	*96.23%/0.2056*	0	61.54%
1	*92.45%/0.3720*	0	84.62%

From the observations above, we may find out that the accuracy of training may not be reduced when the number of hidden neurons is pruned to 4 while a significant drop in accuracy if the number is set to be smaller. The loss of data is increasing slowly before the number is reduced to 4 while increasing drastically for a smaller number of hidden neurons. We may interpret that the greater number of hidden neurons may lead to less loss of data. We may also conclude that the number of hidden neurons as 4 may promote the best balance on tradeoffs between training accuracy and loss of data.

While the testing accuracy after weights update may vary strangely. The initial testing accuracy with 18 number of hidden neurons keeps constant as 61.54%. When the number of hidden neurons is modified to 8 which points to the expected performance if pruned to 4. The testing accuracy is increasing to 69.54% which is expected to be reduced. We may interpret that the initial number of hidden neurons is too large to overfit the training model which may have decreased its performance in testing. The testing accuracy as 69.54% may be its optimal performance in this model. When the number of hidden neurons is modified to 4, the number of neurons to prune gets to zero which may be interpreted that the minimal number of hidden neurons should be 4. As the number of hidden neurons is modified to be smaller, the testing accuracy has dropped to 61.54%. This is expected as decreasing the number of hidden neurons over the minimal may decrease the performance significantly.

As the number of hidden neurons is modified to 1, the testing accuracy is greatly increased to 84.62%. We may interpret that this resulted may be directed from the loss of data. The loss of data for correct predictions (TP, TN predictions in confusion matrix) may be overweighed by the loss of data in false predictions (FN, FP predictions in confusion matrix). As such, testing accuracy has increased.

4.2 The Minimal Number of Hidden Neurons Experiment

In this experiment, we attempt to examine the minimal number of hidden neurons for the training model. The hidden-pruning technique is repeatedly called to the newest neural network until all distinctiveness vector angles between each pair of hidden neurons are less than the threshold. The left number of hidden neurons in the last neural network is expected to be the optimal solution. To find a reliable solution, this process is looped for 50 rounds to find out the mode value in each trial. Average Accuracy on the last neuron network for all loops is estimated on the effects of pruning on the testing accuracy.

Table 2.

Trial	The Minimal Number of Hidden Neurons	Counts	Average Testing Accuracy for Each Trial
1	4	48	62.615%
2	3	39	64.000%
3	3	49	63.076%
4	3	29	62.000%
5	4	32	64.615%
6	4	43	63.231%

From the table2 above, we may observe that the minimal number of hidden neurons exists in training for the Reading Distractions datasets with even a slight increase in the testing accuracy. As the interpretations in experiment 4.1, the initial neural network with 18 hidden neurons may prompt overfitting which leads to a smaller testing accuracy. However, the minimal number of hidden neurons are varying between 3 and 4 in the 6 trials which are expected to be constant through all trials. We may interpret that whenever a new neural network is created, its initial weights on hidden neurons are randomly generalized which may cause the great differences in its final hidden weights after training with other neural networks initialized on the same set of parameters. As such, hidden pruning on these different neural networks may prompt to the different minimal number of hidden neurons and their related testing accuracy.

4.3 Evolutionary Algorithms on Feature Extraction

In this experiment, the process of evolutions in the pre-defined genetic model is observed to examine the minimal set of features in training. For each generation, the chromosome with the largest fitness is chosen to assess its average testing accuracy in training models. The population size is initialized to 100 which is expected to take fast convergence while it may increase greatly in demands of calculation resources. The number of generations is reset to 50 which may significantly reduce the running time for the experiment. It could be adequate to evaluate the performance of feature extractions. The cross-over rate is set to 0.8 and the mutation rate is 0.002. The generations are picked up for every 10 generations for the first 40 generations. All the last 5 generations are selected to evaluate the convergence of the minimal set of features.

Table 3.

Generations	The Most Fitted DNA	Fitness	Average Testing Accuracy
1	[0 0 0 0 0 1 0 0 0 0 1 0 1 0 1 1 1 0]	92.476	80.476%
10	[0 0 0 1 0 1 0 0 0 0 1 1 0 1 0 1 0 0]	93.905	81.905%
20	[0 0 0 1 0 1 1 0 0 0 1 0 0 1 0 1 0 0]	94.143	82.143%
30	[0 0 0 0 0 1 0 1 0 0 1 0 0 1 0 1 0 1]	93.667	81.667%
40	[0 0 1 0 0 1 0 0 0 0 1 0 1 1 0 1 1 1]	93.905	83.905%
46	[0 0 0 0 0 1 0 1 0 0 1 0 0 0 0 1 0 1]	93.000	80.000%
47	[0 0 0 0 0 1 0 1 0 0 1 0 0 1 0 1 0 1]	92.238	80.238%
48	[0 0 0 0 0 1 1 0 0 0 1 0 0 1 0 1 0 1]	92.714	80.714%
49	[0 0 0 0 0 1 0 1 0 0 1 0 0 1 0 1 0 1]	92.238	80.238%
50	[0 0 0 0 0 1 0 1 0 0 1 0 0 1 0 1 0 1]	92.571	80.571%

From the table3 above, we may observe that the minimal set of features exists in training for the Reading Distractions datasets. It could be summarized that some of the features are redundant and 6 of the input features are adequate in training. However, the changes in fitness and the average testing accuracy is not expected that should be constantly increased through generations. As the interpretations in experiment 4.2, once a chromosome is evaluated on its fitness, a new neural network is generalized with random initialization on hidden neuron weights. 10-fold cross-validation has been applied to leverage this variance, while the constant increment on fitness and the testing accuracy is still not feasible on different neural networks. Since parent chromosomes are only replaced with children with higher fitness, we could still interpret that the most fitted DNA in generation 20 has worse performance in training than the most fitted DNA in generation 30 even its fitness is higher. Besides, the last 5 generations are expected to align their most fitted DNA during convergence, while the generation 46 and 48 have different sets of the most fitted DNA. We may interpret that the generation size as 50 may not be enough for convergence that a higher generation size is worthy to evaluate with the expected alignment on the most fitted DNA. Another interpretation is the variance in the initializations of hidden weights in neural networks. This variance may cause the same chromosome to behave differently in different neural networks. As such, initializing equally on the hidden weights for each hidden neuron in all neural networks is expected to provide a more reliable solution.

5 Conclusion and Future Work

5.1 Conclusion

In conclusion, the connection between readers' reactions to digital distractions and their language background is significant in the Reading Distraction Dataset. The classifications on the language background of readers have achieved an average 62% accuracy in testing. The neural network established is deemed to be suitable for hidden neurons pruning. The testing accuracy could be enhanced with a minimal number of hidden neurons which may relieve the overfitting in training. Feature extraction with evolutionary algorithms could be applied to enhance data-efficiency in training that leverages the scarcity of datasets to overcome overfitting. With the pruning on redundant hidden neurons, we may conclude that the simplicity of neural networks could be achieved with insignificant effects on their testing accuracy.

5.2 Future Work

The distinctiveness angle reflects the performance of hidden neurons by evaluating the similarity between each hidden neuron on their functionalities. However, the setting of the threshold angle as an optimal value could not be validated in this paper. In further work, we may hypothesize that a threshold angle may promote the optimal performance in hidden neuron pruning for a specific dataset. The evaluations on the distinctiveness angle may have potential values to promote efficiency in hidden neurons pruning for applications in other datasets.

The neural network should be further modified to initialize equally on the hidden weights for each hidden neuron. The randomness of hidden weights may not be considered as distractions where the correlations of the minimal input features and the minimal number of hidden neurons could be more reliable to analyze with testing accuracy. We may think that the insufficiency in training data may cause the model to be overfitting. Feature extraction and k-fold cross validation may promote the data-efficiency in training while enlargement of the dataset is expected to directly solve the overfitting that may increase the testing accuracy significantly.

References

1. Leana.Copeland, Gedeon, Tamás D. "Visual Distractions Effects on Reading in Digital Environments: A Comparison of First and Second English Language Readers." (2015)
2. D. S. Yeung and Xiao-Qin Zeng, "Hidden neuron pruning for multilayer perceptrons using a sensitivity measure," Proceedings. International Conference on Machine Learning and Cybernetics, Beijing, China, 2002, pp. 1751-1757 vol.4. (2002)
3. Gedeon, T. D., and D. Harris. "Progressive image compression." IJCNN International Joint Conference on Neural Networks. Vol. 4. IEEE. (1992)
4. Gedeon, Tamás D. "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. IEEE. (1995)
5. Steinbach, Michael, George Karypis, and Vipin Kumar. "A comparison of document clustering techniques." KDD workshop on text mining. Vol. 400. No. 1. (2000)
6. J. G. Moreno-Torres, J. A. Saez and F. Herrera, "Study on the Impact of Partition-Induced Dataset Shift on k-Fold Cross-Validation," in IEEE Transactions on Neural Networks and Learning Systems, vol. 23, no. 8, pp. 1304-1312. (2012)
7. S Liu, J Plested. "Structure simplification of neural network for smile classification." Proceedings 2019 Neural Information Processing - 26th International Conference. (2019)
8. Y. Wang, Q. Chen, C. Kang, Q. Xia and M. Luo, "Sparse and Redundant Representation-Based Smart Meter Data Compression and Pattern Extraction," in IEEE Transactions on Power Systems, vol. 32, no. 3, pp. 2142-2151. (2017)