Predicting Image Manipulation by Human Eye Gaze Dataset Using 3 layer MLP with Network Pruning based on Distinctiveness

Sukai Huang

The Australian National University,
Research School of Computer Science
Acton, ACT, Australia
u6492211@anu.edu.au

Abstract. Human's conscious and subconscious visual cognition often reveals much more information than people ever imagined. As such, in order to understand and analyse human cognition signals, we are going to evaluate the performance of the network model which is trained by genetic algorithm. A series of experimental comparison were performed to analyse the performance of various genetic algorithm settings under different choices of generation gap methods, crossover methods, selection methods and mutation methods. In addition to this, we also evaluate the performance of the distinctiveness pruning technique. We tested on the performance of various cases including swapping the order of pruning and setting different threshold value etc.
Our research showed that the predictor trained by genetic algorithm could achieve an testing accuracy of 70.76%, which is higher than the human recognition accuracy (56.0% stated by the experiment run by Sabrina Caldwell (1)). Moreover, the accuracy of the model trained by genetic algorithm is also 7.25% higher than the model trained by back-propagation technique before applying any pruning techniques. The resulting neural network predictor yield not only higher performance than human prediction, but it also demonstrates a huge potential of utilising human physiological signal in area of prediction task.

Keywords: Visual Cognition Signal · Multilayer Perceptron · Experimental Comparison · Genetic Algorithm · Distinctiveness Pruning.

1 Introduction

1.1 Background

Today people encounter more and more manipulated images on the online social media. Compared to the traditional analogue photography, manipulation of digital photography requires less skill. People can easily modify digital photography by using computer apps like Photoshop or Instagram. Rather than checking the metadata of the image, people are often required to use their eyes to spot fake images.

1.2 Motivation

From the experiment run by Sabrina Caldwell, human can only achieve 56.0% accuracy of identifying image manipulation. Accuracy was even lower if participants were not familiar with image manipulation techniques. (2) Human's conscious and subconscious visual cognition signals often reveal much more information than we think. Therefore, one would be wondering whether we could input the eye gaze data directly into a machine learning model and predict the image manipulation with higher accuracy than human's judgement. There are some similar projects that used human physiological signals to predict the target. One of ANU colleague Lu Chen used human's pupillary response to predict whether the participants is looking at genuine or fake angry face. (3) A similar project that was conducted by another colleague Hossian used perceiver's physiological signals to classify genuine smiles (10). However, those projects were mainly focusing on predicting emotional information and few projects were using physiological signal to predict a more general target.

In the previous experiment the model is trained by back-propagation technique. However, one of the major drawback of back propagation was that, the gradient might get stuck at local minima and thereby failing to improve the performance even further. (13). In contrast, a genetic algorithm is able to approximate the global optimal weights when training the MLP network. (13)

1.3 Project Scope

A 3 layer Multilayer Perceptron neural network, trained by genetic algorithm, was constructed to recognise image manipulation and a progressive experimental comparison study is conducted so as to obtain optimised configuration for the genetic algorithm model.

Furthermore, we implemented distinctiveness pruning technique and explored whether the pruning could enhance the model performance.

2 Method

2.1 Data Profiling and pre-processing

The dataset used for this experiment is collected from experiment run by Sabrina Caldwell (1). The experiment recorded the eye gaze data when participants were recognising whether the images that were presented to them were manipulated. The eye gaze data in the dataset consists of the number and duration of visual fixation of the participants as well as the number and duration of fixation by the participants when looking at the target manipulation region. The dataset also stored the code No. of the image the participants looking at. The dataset contained two types of output data. The first one is the binary data that indicates whether the participants are viewing a manipulated version of the image or not. The second one are participants' subjective opinions about if the image presented was manipulated. In our project we only received a part of data from the experiment. The data we have only recorded the observations of 5 images.

Table 1: Number of images groupby manipulation and choice of image

Image manipulated	Image	No.
0	10	39
	11	37
	12	38
	13	38
	14	37
1	10	37
	11	38
	12	38
	13	36
	14	34

Data profiling is necessary when we want to explore the characteristics of the data and also prove the investigations claimed by Sabrina Caldwell (1). (e.g. checking if the data is balanced.(See Table. 1))

The discovery from Sabrina Caldwell (1) (2) and the Table. 2 showed that the accuracy of the participants varies significantly for different images. It indicates that the images would also be a significant feature that helps to predict the image manipulation. Thus the image number can be used as input feature of the neural network.

Table 2: Accuracy of identify image manipulation for different image

Image No.	Accuracy for manipulated image	Accuracy for unmanipulated image
10	76.92%	24.32%
11	81.08%	55.26%
12	44.74%	68.42%
13	78.95%	52.78%
14	75.68%	35.29%

There are concerns about if using image number as input could cause over-fitting problem, given the fact that only 5 different images were provided in the dataset. Nevertheless, it is worth mentioning that based on the observation that different images resulted in significantly different value of eye gaze data, the model that uses only the eye gaze dataset would still implicitly over-fits those 5 image as the eye gaze value is highly dependent on image we chose.

It is necessary to provide a justification for considering image number as input of the model. According to the research, human focus on salient features of an image to discern its meaning (17) (11). Meanwhile, Sabrina Caldwell also found out that human's recognition of image manipulation is affected by how features of an image are presented. The more prominent features of an image would prevent human from identifying less obvious manipulations. (1). It indicates that having different eye gaze and different accuracy for different images is because those 5 images have different salient features. Therefore, the image number can be used as category number that represents different image class categories based on how their salient features.

We also applied z-score standardisation to the input features. Standardisation allows to re-scale the solution space and thereby increasing the efficiency of the training process. (9).

The cross validation technique for this experiment is that the dataset is separated into three sets, called the training set, validation set and the testing set, with the ratio of 6:2:2. The model is trained using training set only and then we evaluate the generalisation performance of the model using the validation dataset. One problem with this method is that the model will end up being over-fitting to the training dataset. Nevertheless, Fig. 2b in the later section 3.1 shows that there is a turning point (around 400^{th} generation) where the training accuracy of the best individual in the population of the genetic algorithm continue to increase while the validation accuracy started to drop. Therefore, we implement Early stopping techniques to achieve a better generalisation performance. The value of model's training and validation accuracy is recorded for every certain number of epochs. A heuristic that is inspired by fisher's linear discriminant is considered:

"Maximise the sum of both the training accuracy and validation accuracy but at the same time minimise the difference of two accuracy"

In that case, we are going to choose the weight and bias parameters that maximise the criterion:

$$criterion = \frac{accuracy_{train} + accuracy_{validation}}{max(1, (accuracy_{train} - accuracy_{validation})^2)} \tag{1}$$

2.2 MLP network model and configuration

A 3 layered MLP with Softmax Function on the output layer is used for this experiment. 3 layers is adequate for the classification problem because a 3 layered Neural Network is able to form arbitrarily complex decision regions. (12). Based on the previous experiment, the configuration of the MLP model is shown in Table. 3.

Presets	Value
size of 1^{st} hidden layer	40
size of 2^{nd} hidden layer	20
activation function	LeakyReLU
adding dropout layer	False

Table 3: Optimised Hyper-parameters

The number of neurons in the hidden layers was deliberately set to be large so that later we are able to apply the pruning technique on the redundant neurons. In fact, the optimal number of hidden neurons is dependent on different tasks. Often we are required to use brutal force to search for the optimal number. As such, Gedeon and Harris suggested to set the number of hidden neurons large when training the model and then remove the redundant neurons at the later stage. (6)

A standard ReLU function is computationally efficient and does not have vanishing gradient problem. However, when inputs approaches to zero or become negative, the model will not learn because the gradient for the non-positive value are 0. As such, we use Leaky ReLU that helps to prevent dying ReLU problem by adding gradient to the non-positive values. (18) However, for this experiment the vanishing gradient problem does not really matter because we used genetic algorithm to train the model. ReLU function is chosen simply because it is more computationally efficient than Sigmoid function.

Dropout layer is not required when the model is trained by genetic algorithm because Dropout technique only makes sense when the model is trained using back-propagation. (15)

2.3 Genetic Algorithm Configurations and Implementation

In this experiment we use genetic algorithm instead of back-propagation algorithm to train the model. The main idea of genetic algorithm is that individual candidates with higher fitness value have more chance of reproducing their offspring and pass their good traits to the next generation. With crossover and mutation mechanism, the model is able to explore more solution space. In this sense, the genetic algorithm is a stochastic search for an optimal solution to a given problem. (4) The detailed procedure of genetic algorithm is described in Fig. 1

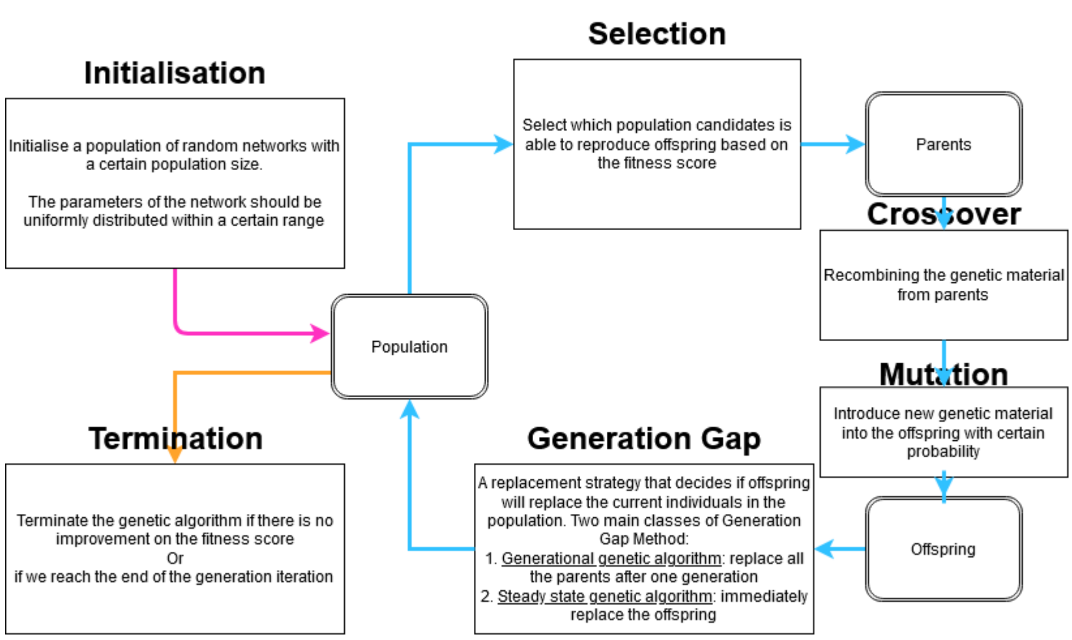


Fig. 1: Genetic Algorithm Flow Chart

The suggested hyper-parameters for the genetic algorithm is shown in Table. 4. The settings is constructed based on simple testing and heuristic. For instance, the parameter boundary is obtained from observing the parameters value range of the back-propagation model trained in the previous experiment. The value range of the parameters of the previous model is within -3 to +3. Thus, it is safe to set the boundary by extending the original range a bit. Similarly, population size is set based on the computation strength of the researchers' device. Generally, a smaller size of the population will limit the model to explore more solution space while a larger size of population will take longer time to train the model. In addition, we set the number of generation for training larger enough (i.e. 2000) so that we can monitor the trend of loss and accuracy of the model nicely. In this experiment the negated value of cross entropy loss is used to be the fitness value of the candidates so that the candidate networks which have lower cross entropy loss has higher chance to survive at the end of the optimisation process.

Presets	Value
Population Size	100
Crossover Rate	0.8
Mutation Rate	0.002
Number of Generation	2000
Parent Size	Sexual
Parameter Boundary	(-5,5)
Fitness	-cross entropy

Table 4: Suggested GA Hyper-parameters

Different generation gap method, selection method, crossover method and mutation method are tested. Those were:

1. Generation Gap Method
   (a) Generational genetic Algorithm: Replace all parents after one generation.
   (b) Replace worst steady state genetic algorithm: immediately swap the offspring with the worst individuals in the current population.
2. Selection Method
   (a) Tournament selection: Sample a certain number of individuals from the current population as tournament group and then select the best individual from this group.
   (b) Proportional selection: A probability distribution proportional to the fitness score of the population is generated and the parents are sampled by the distribution.
3. Crossover Method
   (a) Arithmetic crossover: recombine the parent's genes with a random ratio.
   (b) Linear combination: generate 3 children in which the genes are $x_1^{'} = 0.5 \times x_1 + 0.5 \times x_2$, $x_2^{'} = 1.5 \times x_1 - 0.5 \times x_2$, $x_3^{'} = -0.5 \times x_1 + 1.5 \times x_2$ respectively. After that select best two based on fitness score as offspring.
4. Mutation Method
   (a) Static mutation: based on the given mutation rate, each parameter of the candidate network has the equal chance of being mutated to a random number that is within the limited range.
   (b) Dynamic mutation: decrease the mutation rate as the iteration of generation increases.

The model is trained using different combination of those methods. The method candidates are selected based on the knowledge of whether these methods can fit to the neural network model. For example, we select Arithmetic crossover and Linear combination method because these two methods are suitable for neural network training. First of all, parameters in neural network is represented by floating number, thus we need to use crossover methods that deal with floating number representation. Secondly, researchers claim that using typical crossover method such as one point crossover or uniform crossover in training neural network will disrupt the functionality of neural network (16). More discussion will be in "Results and Discussion" Section.

2.4 Distinctiveness Pruning

Neural networks models often require a significant amount of computing, memory and power as the number of hidden layers increases. In modern deep learning models, an accurate ResNet152 model will require 1.5 weeks to train. (14) This hence limits the productivity of the researchers. Rather than pruning based on the neurons contribution, Gedeon and Harris proposed a simple and computationally cheap distinctiveness pruning method. (6) Distinctiveness pruning is to remove neurons that is similar or complementary to others. The distinctiveness of two selected neurons is measured by the cosine similarity of the neuron's output activation vector over the batch input dataset. In this experiment we investigated the effect of distinctiveness pruning technique on the performance of the model. Below is the detailed steps in obtaining the redundant hidden neurons:

1. Obtain the activation matrix of the hidden layer by inputting the batch training dataset. Each vectors in the activation matrix represented the functionality of the hidden unit in the pattern space.
2. Normalise the activation matrix so that different vectors can be compared under the same domain.
3. Calculate the cosine similarity for each neuron pair. After that we obtain the angle value between each activation vector.
4. Compared the angle with the angle threshold. If the angle between the two vectors was smaller than threshold (e.g. 15 °), then these two hidden units were said to have similar behaviour, one of them was removed.
5. Practically, we need to decide the order of pruning if we have more than one hidden layer.

In this section the optimal model obtained from the previous section is selected for the pruning experiment. We applied the distinctiveness pruning to both of the hidden layers and progressively removed the redundant neurons of the hidden layer until all neurons in the layer were distinctive from each other (degree larger than 15°).

In the original experiment run by Gedeon and Harris (5), the tested network is a Auto-associative network, which only had one single hidden layer. However, in this experiment, the neural network for recognising image manipulation task contained two hidden layers and thus, the pruning order matters. In this experiment we pruned the layer one after the other. We have two choices of pruning order: pruning the first hidden layer first and then the second hidden layer versus the other way round. We tested on both cases and investigated how the order of pruning would impact on the performance of the network.

Increasing the degree threshold allows to compress the network furthermore. If the threshold value is set very high, we might remove neurons that has distinctive functionality. In this experiment we also tested 30°pruning threshold. Also in order to avoid under-fitting, we further trained the model using back-propagation algorithm for a certain epoch number after the removal of a distinctive neuron. Genetic algorithm cannot be implemented here because the population data is not stored at this stage.

3 Results and Discussion

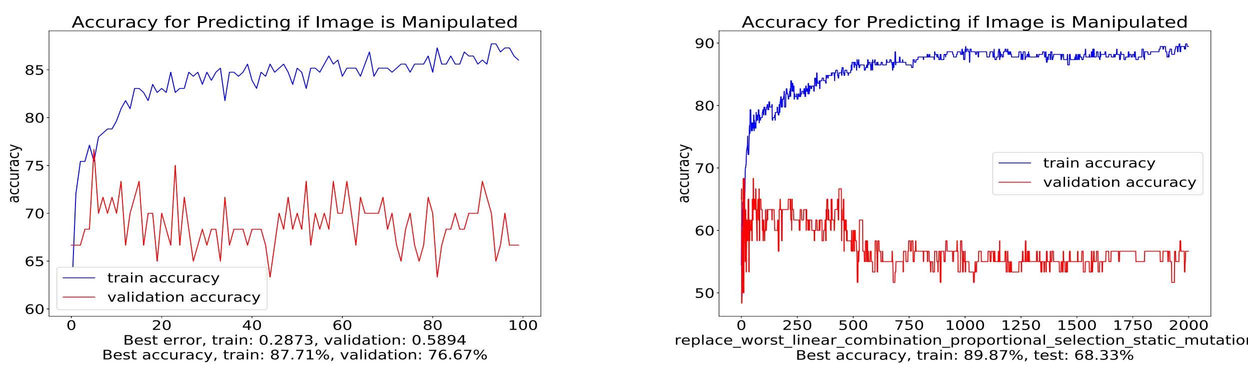
3.1 Performance compared to model trained by SGD

Figure. 2 shows the training and validation accuracy comparison between the optimal model trained by back-propagation algorithm in the previous experiment and the optimal model trained by genetic algorithm at this time. By using the early stopping technique and heuristic selection, the best model for both back-propagation method and genetic algorithm method is shown in the Table. 5

Table 5: Performance Comparison Betweeb Backprop and GA

Type	Training Accuracy	Validation Accuracy	Testing Accuracy
Backpropagation	75.42	76.67	63.51
Genetic algorithm	67.51	68.33	70.67

According to the result, both models have the similar training and validation accuracy trend during training while the model that uses genetic algorithm optimising technique is able to obtain a higher testing accuracy than the back-propagation model(i.e. 70.67 % vs 63.51 %). It shows that the model which is trained by genetic algorithm could obtain a higher generalisation ability. In other words, the result shows that the genetic algorithm model was not strongly affect by the overfitting problem. In contrast, although backpropagation model is able to achieve higher training and validation accuracy in the training process, the model is strongly overfited to the training data and thereby generating a relatively low generalisation performance. This were probably because compared to the back-propagation, the genetic algorithm is a global search technique. Thus, the probability of finding a global optimum greatly increases. (7) Meanwhile, back-propagation uses gradient descent technique, which could get stuck in the local minima. (8)



(a) Accuracy Graph of optimal model trained by back-propagation

(b) Accuracy Graph of optimal model trained by genetic algorithm

Fig. 2: performance comparison between optimal model trained by different optimising technique

3.2 Performance comparison of different genetic algorithm settings

As shown in Table. 6, all of the genetic algorithm settings result in accuracy higher than human recognition accuracy (56.0%). The best three settings is coloured in the table. From the result it is shown that "Replace Worst" method generally generates higher accuracy than the "Generational Genetic Algorithm". The reason might be because first of all, based on the coding rule, we should avoid adding unnecessary traits into the model. In our case, the training process of the task model has no relation to the idea of generational evolution (i.e. there is no generation concept for this task). Thus generational genetic algorithm is unrepresentative for this task. Secondly, it might be because generational genetic algorithm generate a more diverse population than the replace worst algorithm, thus it would take longer time to find the optimal solution.

Dynamic mutation method is meant to have a larger mutation rate in the earlier stage, so that the model is able to explore more solution space. Meanwhile in the later stage, the mutation rate will be very small so that the model is able to focus on exploiting the optimal solution under the searched space. This is also called "Exploration and Exploitation trade off". However, in our experiment we see that the models that use dynamic mutation method generally perform worse than the models using static mutation method. This were probably because in our experiment the dynamic mutation rate is set to be decreasing exponentially over time. Thus, it is highly possible that the exploration ability of the model become inactive in a early stage, thereby preventing the model from reaching the global optimal result.

There is no significant performance difference between the arithmetic crossover and linear combination method. Generally, arithmetic crossover generates more diverse offspring than linear combination because it recombines the parents parameters by a random ratio and thus there is no guarantee that the offspring will inherit the good traits from its parents. In fact, researchers claim that using typical crossover method such as one point crossover or uniform crossover in training neural network will disrupt the functionality of neural network (16). Based on the knowledge of the neural network, neurons in successive layers are strongly interconnected. For instance, in transfer learning, cutting the the neural network in the middle and connect to another pre-trained model without further training will decrease the performance. Thus, crossover method will definitely break the interconnection between different neurons. In addition, there is no univocal genotypic representation for the neural network solutions, for example, we can swap the neurons in the same layer of the neural network without changing the functionality at all. The permutation problem is likely to make crossover inefficient, and disruptive on nearly optimal solutions.

The result also shows that this task has no significant bias on either tournament selection method or proportional selection method. We can see that the average accuracy of using each method are very close to each other. (i.e.72.09% and 73.36% respectively)

Table 6: Performance of Network trained by GA with different settings

No.	Generation Gap		Selection Method		Crossover Method		Mutation Method		Accuracy	
	Generational Gentic Alg	Replace Worst	Tournament	Proportional	Arithmetic Crossover	Linear Combination	Static	Dynamic	train	test
1	x		x		x		x		69.02%	70.67%
2	x		x		x			x	69.02%	70.67%
3	x		x			x	x		71.04%	72.00%
4	x		x			x		x	69.70%	70.67%
5	x			x	x		x		68.69%	70.67%
6	x			x	x			x	67.34%	68.00%
7	x			x		x	x		71.72%	73.33%
8	x			x		x		x	72.39%	72.00%
9		x	x		x		x		76.09%	74.67%
10		x	x		x			x	75.08%	73.33%
11		x	x			x	x		69.36%	69.33%
12		x	x			x		x	77.44%	78.67%
13		x		x	x		x		77.44%	76.00%
14		x		x	x			x	75.42%	76.00%
15		x		x		x	x		79.12%	78.67%
16		x		x		x		x	74.75%	74.67%

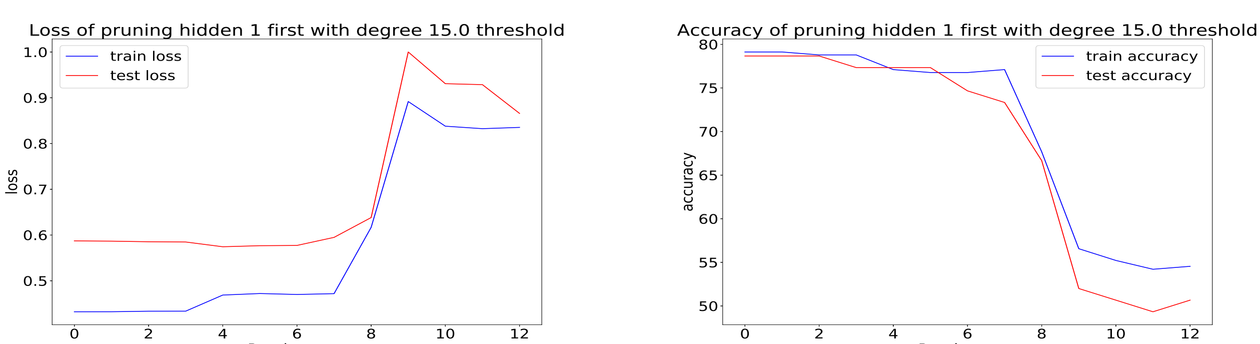
3.3 Analysis of Distinctiveness Pruning Effect

We applied pruning technique to the model. The setting for the first try is shown in Table. 7. Since the first try only considered neurons which the smallest angle are within 15°, we do not further train the model. Nevertheless, back-propagation technique is used for further training instead of genetic algorithm. It is because genetic algorithm is a continuous optimising process. At this stage the population database of the genetic algorithm was not stored and thus further training on genetic algorithm cannot be implemented.

Hyper-parameter	Value
degree threshold	15.0°
Which first to be pruned	1^{st} hidden layer
No. of epochs for further training	50

Table 7: First try for pruning

As shown in Figure. 3, the loss value is almost unchanged for the first 7 neurons. As we further removed 8^{th} neuron and so on, the loss value significantly increased. There is no significant improvement for the generalisation ability of the network. As shown in the graph, the testing accuracy of the pruned model did not increase. This observation is different compared to the model that is trained using back-propagation. In our previous experiment, the model trained by back-propagation could increase its testing accuracy after distinctiveness pruning. One of the reason could be that since genetic algorithm is a global search technique, the parameters of the model have already reached their global minima. Therefore, there is no more space for the model to increases its generalisation performance. In fact, the testing accuracy of the genetic algorithm model has already reached quite high even before the pruning process. Overall, the distinctiveness pruning helps to remove redundant hidden units while maintaining the prediction performance of the model.



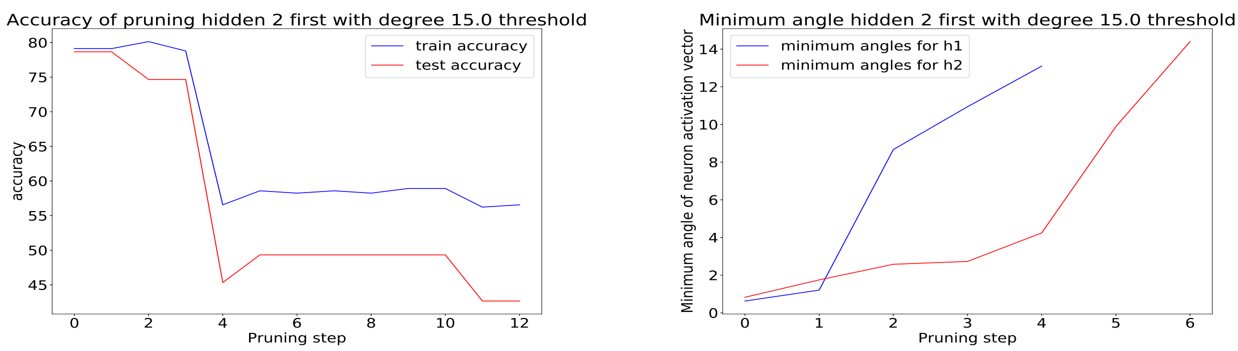
(a) Loss trend as the no. of removal increase

(b) Accuracy trend as the no. of removal increases

Fig. 3: Loss and Accuracy Graph for the first try of the pruning

3.4 Order of Pruning

Figure.4 shows the performance when we let the second hidden layer to be pruned first. As we can see, the total number of neurons that are removed for the second hidden layer is 7 while the accuracy significantly decrease when we removed the 4^{th} neuron in the second hidden layer. This means that the 4^{th} neuron is supposed to be redundant but at the same time removing it causes a huge drop on the performance. This results from either inconsistency of representing the distinctiveness of the neuron's functionality by cosine similarity (which means the Godeon and Harris' distinctiveness pruning technique has serious fault) or from the fact that neurons between two layers are interconnected strongly such that removing the the neurons that seems redundant in the second layer would disable the functionality of the significant neurons that are connected in the previous layer.



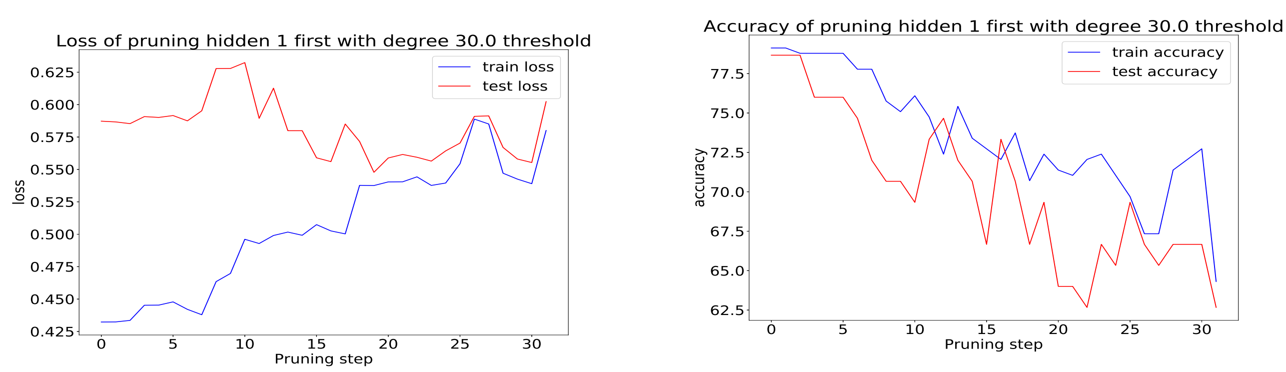
(a) Accuracy when second hidden layer is pruned first

(b) Smallest angle of the removed neuron

Fig. 4: Accuracy and Angle graph for pruning experiments that swap the order of pruning

3.5 Threshold for Pruning

When removing neurons which the smallest degree is larger than 15°, we need to further train the model so as to avoid underfitting. In the experiment we change the threshold degree value up to 30°and then set the number of epochs for further training to be 50. As mentioned above, genetic algorithm cannot be used here due to the lack of the population data of the original genetic algorithm. As shown in Figure. 5, the model is able to maintain its performance after removing 15 numbers of neuron.

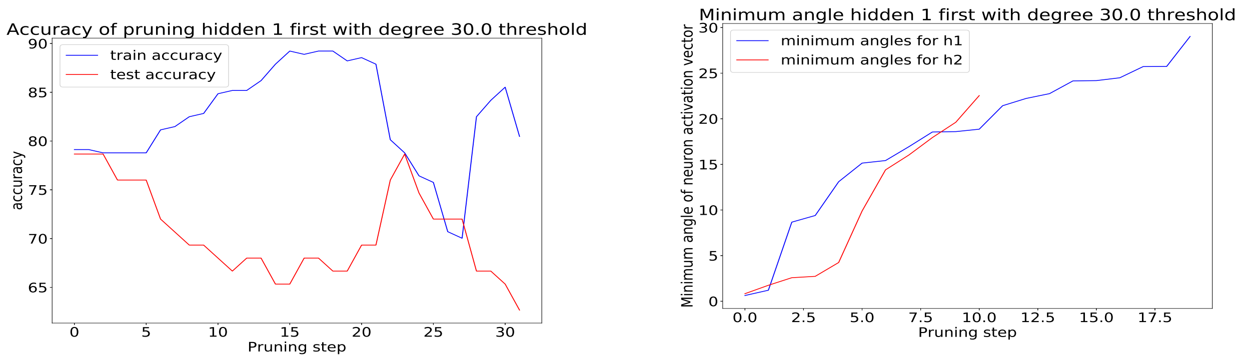


(a) Loss when threshold changes to 30, with further training

(b) Accuracy when threshold changes to 30, with further training

Fig. 5: Loss and Accuracy graph for pruning experiments that increase the threshold

Nonetheless, there is descending trend as the number of removed neurons increases. Increasing the number of training epochs would produce a very interesting phenomenon, as shown in Figure. 6. The result shows the training accuracy for this test could achieve above 90%, which is a very incredible performance and can never be obtained from the original non-pruned network. We can also see that the training accuracy increases and testing accuracy decreases when we remove about first 20 neurons. This is because a very large training epoch for back-propagation algorithm would definitely cause the overfitting of the model. However, from the removal of 21st neuron to the 23rd neuron, the training accuracy dropped and the testing accuracy restored. This were probably because after removing 20 neurons, all the first layer neurons is gone and network now become a single hidden layer network. As such the network will re-construct all its internal features and thereby restoring the generalisation ability.



(a) Accuracy when threshold changes to 30, with epoch 2000

(b) Smallest angles of removed neurons when threshold changes to 30, with epoch 2000

Fig. 6: Accuracy and Angle of removed neurons graph for pruning experiments that has a high epoch number

4 Conclusion

In this experiment we successfully demonstrate that a 3 Layer Multilayer Perceptron model that is trained by genetic algorithm is capable of predicting image manipulation. The result also shows that the model that is trained by genetic algorithm could obtain a higher generalisation performance than the model that is trained by back-propagation, though this advantage is gone when we applied pruning technique. Our experiment shows that the pruning technique is useful to shrink the model size without losing too much prediction performance. However, compared to the model that is trained by back-propagation, the generalisation performance of the genetic algorithm model did not improve during the pruning process. The final model's could achieve a high accuracy: 70.76%, which is much more accurate than the human's recognition (56%) as well as the model using back-propagation technique.

5 Future Work

5.1 Rationale behind the result

There is lack of the rationale behind the success of our model. More diverse dataset are needed to further discover the reason behind it. For example, we might want to see whether we can use the eye gaze data when people are not informed to identify image manipulation to predict the result.

5.2 Salient feature of image as input

In this experiment we treat image number as a representation of the salient feature of the image and successfully increase the performance. In the future work we could record the real salient feature of image as input and test whether this is indeed helpful in predicting the image manipulation. More images from different categories may be required so as to avoid overfitting.

5.3 Further investigation of the impact of pruning order

In this experiment we found out that pruning the later hidden layers first could achieve a better result. But the rationale behind it is still lacking. Also, in this experiment we only pruned the hidden layer one after each. In the future we might want to see the performance of other combination of the pruning order. For example, we can remove one neuron for each hidden layer each time instead of pruning the whole hidden layer before moving to the next one.

Bibliography

[1] Caldwell, S., Gedeon, T., Jones, R., Copeland, L.: Imperfect Understandings : A Grounded Theory And Eye Gaze Investigation Of Human Perceptions Of Manipulated And Unmanipulated Digital Images (308), 1–9 (2015)

[2] Caldwell, S.B., Gedeon, T.D., Jones, R.L., Henschke, M.: Comparing eye gaze tracking to reported perceptions of manipulated and unmanipulated digital images Accuracy in identifying manipulated images unaided **14**(3), 26–36

[3] Chen, L.: Human Classification and Physiological Signals for Angry Faces (2017)

[4] Engelbrecht, A.P.: Computational intelligence: an introduction. John Wiley & Sons (2007)

[5] Gedeon, T.D.: Indicators of hidden neuron functionality: The weight matrix versus neuron behaviour. Proceedings - 1995 2nd New Zealand International Two-Stream Conference on Artificial Neural Networks and Expert Systems, ANNES 1995 pp. 26–29 (1995). https://doi.org/10.1109/ANNES.1995.499431

[6] Gedeon, T.D., Harris, D.: Network Reduction Techniques pp. 1–7 (1988)

[7] Gen, M., Lin, L.: Genetic algorithms. Wiley Encyclopedia of Computer Science and Engineering pp. 1–15 (2007)

[8] Gori, M., Tesi, A.: On the problem of local minima in backpropagation. IEEE Trans. Pattern Anal. Mach. Intell. **14**(1), 76–86 (Jan 1992). https://doi.org/10.1109/34.107014, https://doi.org/10.1109/34.107014

[9] Han, J., Kamber, M., Pei, J.: Data Mining: Concepts and Techniques (2012). https://doi.org/10.1016/C2009-0-61819-5

[10] Hossain, M.Z., Gedeon, T.: Classifying posed and real smiles from observers' peripheral physiology. In: ACM International Conference Proceeding Series (2017). https://doi.org/10.1145/3154862.3154893

[11] Itti, L., Koch, C.: A saliency-based search mechanism for overt and covert shifts of visual attention. In: Vision Research (2000). https://doi.org/10.1016/S0042-6989(99)00163-7

[12] Lippmann, R.P.: Pattern Classification Using Neural Networks. IEEE Communications Magazine (1989). https://doi.org/10.1109/35.41401

[13] Montana, D.J., Davis, L.: Training Feedforward Neural Networks Using Genetic Algorithms. Proceedings of the 11th International Joint Conference on Artificial intelligence - Volume 1 (1989)

[14] Singh, R.: Pruning deep neural networks (Aug 2019), https://towardsdatascience.com/pruning-deep-neural-network-56cae1ec5505

[15] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: A simple way to prevent neural networks from overfitting. Journal of Machine Learning Research (2014)

[16] Thierens, D., Suykens, J., Vandewalle, J., De Moor, B.: Genetic weight optimization of a feedforward neural network controller. In: Artificial Neural Nets and Genetic Algorithms. pp. 658–663. Springer (1993)

[17] Treisman, A.M., Gelade, G.: A feature-integration theory of attention. Cognitive Psychology (1980). https://doi.org/10.1016/0010-0285(80)90005-5

[18] Xu, B., Wang, N., Chen, T., Li, M.: Empirical evaluation of rectified activations in convolutional network. arXiv preprint arXiv:1505.00853 (2015)