Investigate With Casper Neural Network: A Novel Approach Exploring Potential Correlations Between Human Eye Gaze And Image Manipulation

Ziyang Liu

The Australian National University, Research School of Computer Science
Acton, ACT, Australia
U6210090@anu.edu.au

Abstract. To investigate whether human have the ability to detect and recognise manipulated images naturally, we proposed to apply a generative neural network – Casper Neural Network to explore the correlations between human eye gaze response and image manipulation. Eye gaze tracking was used to investigate on 80 volunteers viewing a combination of manipulated and unmanipulated images together with pre-familiarisation on image manipulation techniques. Based on the proposed Casper neural network, we also tried to improv the Casper network's performance by redesigning the artchitecture. However, we still found its performance is lower than the baseline fully connected neural network.

Keywords: Eye gaze; manipulated images; casper neural network.

1 Introduction

Image manipulation is no longer the skill that can be owned by professional photographers solely. With simpler image editing tools appearing in the market, normal users got more chances and less difficulties on editing images and upload them to the social media platform. For instance, Facebook users share more than three million images each day on the platform. Those images could be taken directly from cameras or being manipulated through software.

Therefore, the significance of this research is to explore whether human can detect manipulated images through his visual system. There could be unrecognised visual system response to the unnatural images, so we apply an artificial neural network to explore the hidden correlation among them.

1.1 Eye gaze dataset

The eye gaze dataset is collected through a series of experiments conducted on 80 participants [2]. Each participant will view the text provided along the image to have a basic context, then vote for whether the image is manipulated or not in each cohort. The dataset also includes the ground truth of image manipulation and the supporting data that is relevant to human eye gaze. The image manipulation for the experiment only involves three actions: copy, move or splicing.

In the dataset, the number of fixation and the total time of duration for each participant's eye gaze in the experiment is collected. The displayed images and participants are also numbered sequentially. The total amount of time that participant spent looking within the target area is recorded in seconds as well.

Participants' eye gaze was recorded using two Facelab 5.0.2 infra-red cameras and a single IR light emitter pod centrally located below the monitor displaying the images. To record video evidence of each experiment, Eyeworks v3.8 was taken for usage and also applied for experiment design, delivery, recording and analysis.

2 Method

2.1 Casper Algorithm

The Casper algorithm [3] is a very powerful and dynamic method for training artificial neural networks. It is a generative method that dynamically adjusts the network architecture by adding new neurons when the network is considered in a 'freezing' stage.

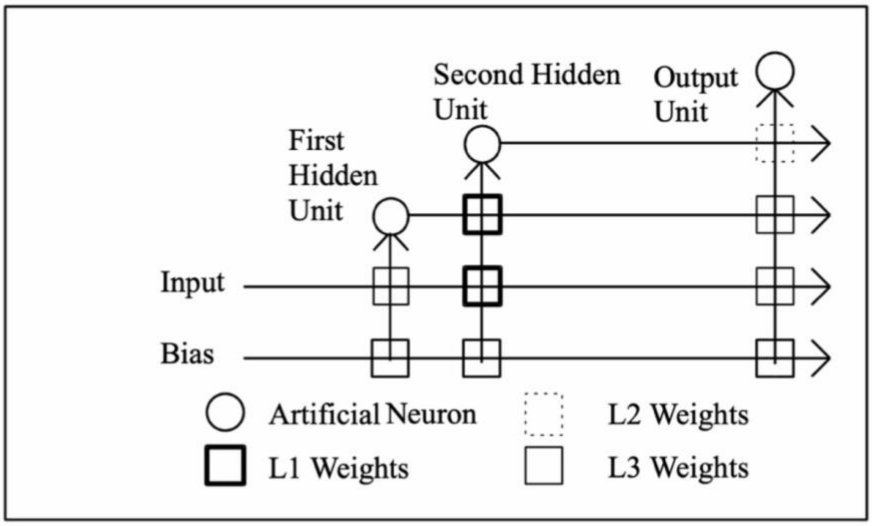


Fig. 1. A conceptual graph showing different neurons of the Casper model

Compared to the original Cascade Correlation algorithm [4], Casper algorithm provides better flexibility during the training phase. It limits the weight decay to a much smaller range for old generation weights (L2) but gives a larger weight decay to new weights (L1). These different weight decays assure the latest neuron adjusts the gradient quickly and the loss value could drop into another local minima. This mechanism also reduces the interference to other weights while the new neuron is reducing the network error.

To avoid the network being too 'deep' for training, the Casper algorithm also introduces a TIME_PERIOD threshold to check whether the network is suitable to install new neuron. The TIME_PERIOD is calculated with formula below. P is a user determined constant and N is the number of hidden neurons installed at the network. After each TIME_PERIOD of epochs, the reduction on loss will be checked to determine whether to install new neuron. The threshold for the loss reduction is usually set to 1%, if the loss of network falls larger than 1% then a new neuron can be added to the Casper network.

$$\text{TIME\_PERIOD} = P \times N + 15 \quad (1)$$

The Casper algorithm also improves the generalisation ability by introducing Simulated Annealing (SA) term into the weight decay. The large initial value of the weight decay will be reduced by SA term which is often used in the SARPROP algorithm [5].

$$\frac{\partial E}{\partial W_{ij}} = \frac{\partial E}{\partial W_{ij}} - K \times sign(W_{ij}) \times W_{ij}^2 \times 2^{-0.01 \times HEPOCH} \quad (2)$$

As formula 2 described, by combining Simulated Annealing and RPROP algorithm, the error gradient can be calculated. In the formula, HEPOCH is the number of the epochs since the latest neuron installed on the network. K is a user determined constant as it is problem dependent. In this scenario, K is set to 1 to achieve the best convergence.

2.2 Pytorch Framework

Pytorch [6] is an open sourced deep learning framework that helps researchers and developers build, test and evaluate their neural network models easily without spending unnecessary time on bottom level mechanisms.

Pytorch is chosen to build the Casper network as it provides comprehensive documentations and APIs for deep learning, and the popularity of the framework makes the source code easier to be reviewed by peer researchers.

2.3 Data Pre-processing

As the original data are all numeric input data, the input can be directly fed into the network. However, as shown in Table 1, the distribution of the data is imbalanced. A normalisation is therefore applied to the input data to receive a better training result.

Table 1. Distribution of different columns of input data.

Column Name	Min	Max	Type
participant	1	80	Category Integer
num_fix	6	215	Integer
fix_dur	0.967	63.459	Float
num_man_fixs	0	115	Integer
image	10	14	Category Integer
man_fixs_dur	0	32.305	Float
image manipulated	0	1	Binary Integer

To produce a persuasive result, training and testing dataset is split in a 7:3 fashion and. However, for the ease of debugging, the seed is fixed during the developing phase. Furthermore, by utilising the provided Dataset API, the data is fed to the model in the batch size of 10.

2.4 Debug methods

While developing the model, it is found that the parameters of the model didn't generate gradients as expected. To check the completeness of the model, Tensorboard is introduced to the debugging as a visualisation technique assisting the observation on neural network.

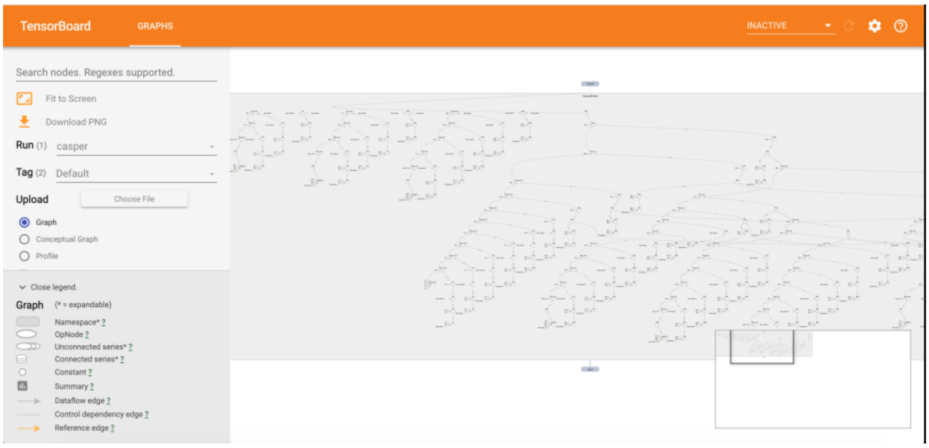
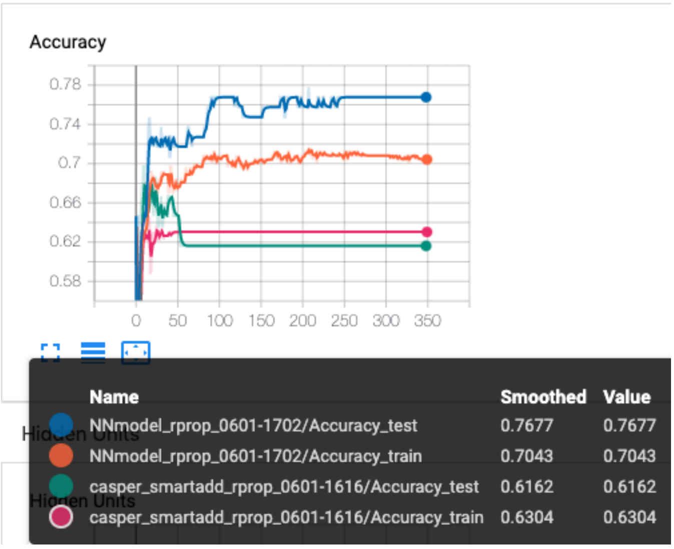


Fig. 2. A conceptual graph showing connections between different nodes in the Casper model

Apart from the Tensorboard, ipdb is also applied for debug. With the ipdb tool, variable status is much easier to track and follow. When debugging the optimiser, the parameters can be seen by using ipdb to measure the performance of the algorithm.

3 Result and Discussion

3.1 Result Comparison



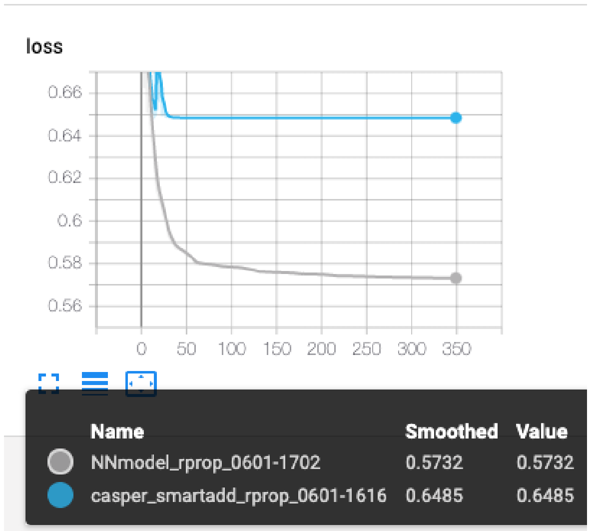
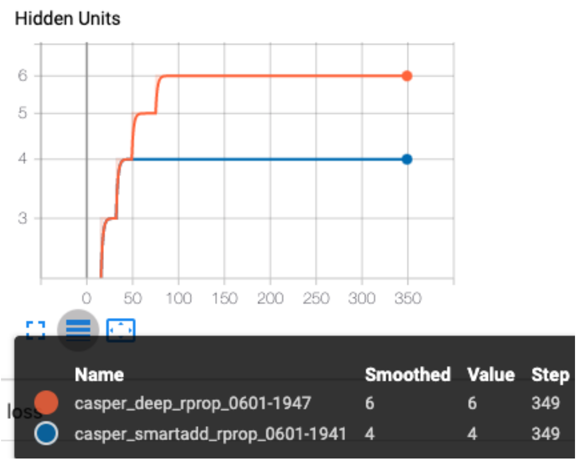


Fig. 3. The training and testing dataset accuracy of Casper model and fully connected neural network.
Fig. 4. The convergence of Casper model and fully connected neural network's loss during training.

The epoch number is set to 350 to ensure both networks are fully converged. As the model finished training, there are four hidden neurons inserted to the Casper network. On average of five runs with same settings, the final accuracy of train and test dataset for Casper model is 63.04% and 61.62% respectively. Compared to the baseline model – a three layer fully connected neuron network which brought 70.43% and 76.77% for train and test dataset, the Casper model is not performing well as expected.

3.2 Deeper model

With the experiment result showing above, we believe the model might have too little parameters to fit to the dataset. Therefore, the install threshold is adjusted from 1% to 1%% to achieve a deeper model. In this training, the model installed 6 hidden neurons but still converged in the same loss value.



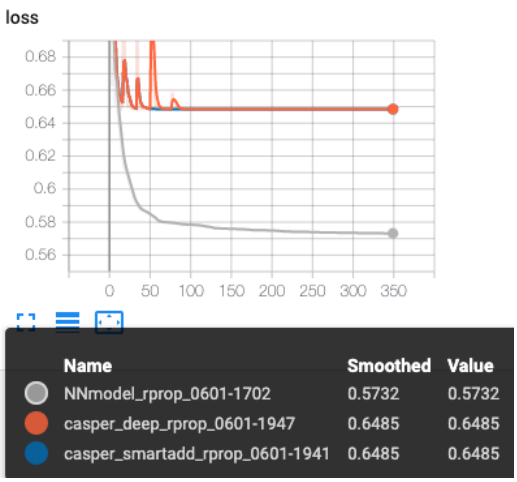
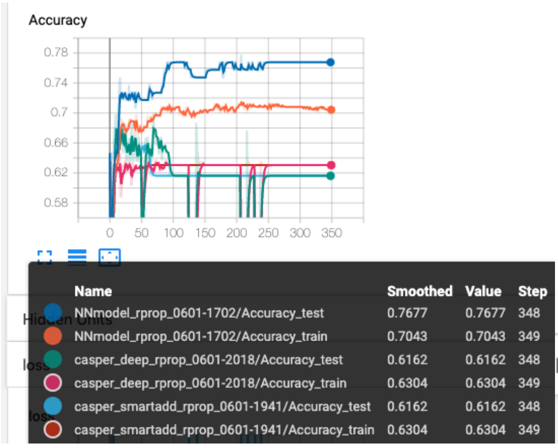


Fig. 5. A graph describing the installation of neurons on Casper while converging.
Fig. 6. The loss trend of deeper Casper model.

As figure 5 and 6 displayed, the model with deeper hidden layers still converges to the same point and remains little changes on the loss value. Based on this result, we proposed a more aggressive design on the Casper model by removing the loss change rate threshold. The model will install new neurons only based on the TIME_PERIOD variable. In this experiment, the model was trained in the same epochs (350) but have 19 hidden units.



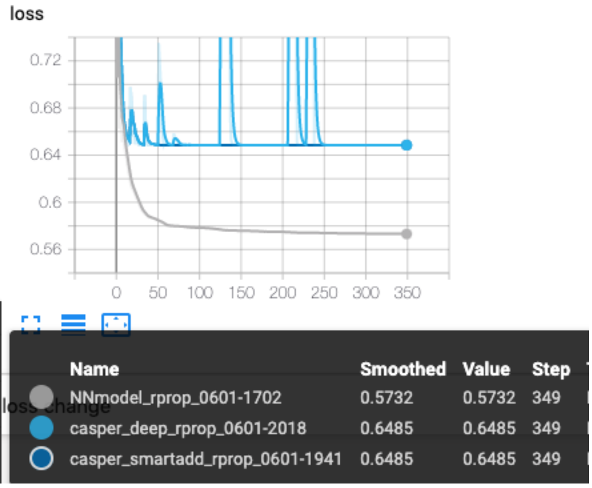


Fig. 7. A graph describing the installation of neurons on the 19-layer-Casper while converging, compared to a 4-layer-Casper and a fully connected neural network (baseline model). The 19-layer-Casper is named as 'casper_deep_rprop' and 4-layer-Casper is named as 'casper_samrtadd_rprop'

Fig. 8. The loss trend of deeper Casper model, comparing to the normal 4-layer-Casper and the baseline fully connected neural network.

From the figure 7 and 8 we can see the deeper Casper model's performance bounces down several time after a new neuron is installed but converged back to the same point in the later stage. This result pointed out the real problem of Casper model – pure linear calculation. As the linear calculation can be divided into several sub linear calculations, the newly installed neuron is not providing extra exploration ability for the network [7]. Instead, it is only substituting one linear calculation with several other linear formula.

3.3 Conclusion

The Casper model provides basic ability in this classification task, showing there exists correlation between human eye gaze response and image manipulation. However, the Casper model is not performing well compared to the baseline neural network. In this classification task, adding extra hidden units will not improve the performance of the model. To improve the Casper model performance, non-linear neurons need to be added into the model to provide exploration ability for the model.

4 Future Work

According to the conclusion and experiment result, we expect to have non-linear activation neurons added to the Casper network. However, the condition of adding it and the performance improvement it may bring still need further research.

In addition to the neuron types, the way of network connection may also affect the performance of Casper model. Therefore, some connections between neurons can be set as dropouts. These methods also need further research.

Acknowledgments. Unless specified in the paper or codes, I acknowledge all the content and codes are created by me, Ziyang Liu.

References

1. Ericsson, & Qualcomm. A Focus on Efficiency. White Paper, Facebook, (2013)
2. Calwell, 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. In: World Congress on Electronical Engineering and Computer System and Science, paper No.308, Barcelona (2015)
3. Treadgold, N.K., Gedeon, T.D.: A Cascade Network Algorithm Employing Progressive RPROP (2005)
4. Fahlman, S.E., Lebiere, C.: The cascade-correlatoin learning architecture. In: Advances in Neural Information Processing II, pp.524-532, Morgan Kauffman, San Mateo (1988)
5. Treadgold, N.K., Gedeon, T.D.: A Simulated Annealing Enhancement to Resilient Backpropogation. In: Processing International Confluence Soft and Intelligent Computnig, pp. 293-298, Budapest (1996)
6. Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., Lerer, A.: Automatic differentiation in Pytorch. In: 31st Conference on Neural Information Processing Systems, Long Beach (2017)
7. Huang, D., Ma, S.: Linear and Nonlinear Feedforward Neural Network Classifiers: A Comprehensive Understanding. In: Journal of Intelligent Systems, vol. 9: Issue 1 (1999)