Cascade Correlation on Binary prediction with Dropout (CasCorBD): Propose Neural Network Architecture on Image Manipulation Prediction

Kai Yin Chan,

The Australian National Univeristy, Research School of Computer Science
Acton, ACT, Australia
U6405500@anu.edu.au

Abstract.

The rise in popularity of neural networks was supported by a wave of significant hardware improvements in the past decade. Yet, for a consumer grade lightweight hardware, such as an AR/VR headset, supposedly useful image manipulation predictions remained challenging. Learning from a convoluted eye gazing dataset will require deep learning that can be cumbersome. To effectively make a simultaneous image manipulation prediction, Cascade Correlation with One dimensional pass on Binary prediction (CasCorOB) and CasCorOB with dropout (CasCorBD) is proposed. Adapted from Cascade Correlation (CasCor), CasCorBD (and CasCorOB) is a binary prediction architecture that is advantaged on better testing accuracy on eye gaze dataset when compared against a baseline 2 output CasCor and 3 layer fully connected network, at 27.40% and 23.89% increased testing accuracy on average. CasCorBD is strictly better in accuracy and training speed against CasCorOB, whereas CasCorOB is strictly better than CasCor. CasCorBD is 97.45% faster than CasCor. While human participants predicted with 55.7% accuracy on image manipulation, CasCorBD was able to detect with 67.81% accuracy on average, based on the second-hand information – eye gaze habits of the participants. Future works can be made to this architecture by employing a varied dropout rate for different connections in the network. Testing against the changes in Pearson's correlation and zero averaged variables may also show the performance benefit distributions between CasCorOB and CasCor. Nevertheless, with further research, the use of this architecture in wearable electronics may better comprehend user intentions for better personalized user experience.

Keywords: CasCorBD, CasCorOB, deep neural network architecture, binary output prediction, one dimensional value pass, dropout, PyTorch, image manipulation.

1 Introduction

Traditionally, neural networks required commercial grade machines that possess higher magnitude of processing powers, to compare with a personal computer. Upon the transfer of neural network processing platform to the more efficient graphics processing units (GPU), in additional to the introduction of GPU-based dedicated machine-learning hardware, consumer grade computers are becoming more accessible to training big datasets at a reasonable time scale [1].

The popularity and expectations for AR/VR (or "XR" for extended reality) headsets are in the rise as of late. With aim to bring automation of trivial tasks in real life, XR is the bridge to connect machines to reality. Complex functionalities, such as real-time image manipulation detection, is more efficient with XR that mirrors the vision of the user, when compared to the traditional computer method that requires a scanner. To review the image live and adjust the predictions adaptive to the individual user profile eye gaze behaviours, XR devices must be able to simultaneously train the algorithm. However, XR standalone headsets has a lightweight processing performance and gathers convoluted unprocessed sensory data. In additional to the rise of photo editing software quality, better image manipulation detection software is increasingly essential to combat the increasing pattern recognition, putting pressure to the further popularization of XR as a productive utility. Further to the growing expectations for XR, it seeks stricter neural network architectures that is more accurate (i.e. more hidden neuron layers) while requiring it to be efficient (i.e. fewer epochs) [2]. To improve accuracy and speed of neural network predictions, many architectures and filtering processes had been proposed dating back in the 1950's, extending to the growth in popularity of the recent decade.

1.1 Cascade Correlation (CasCor)

In 1990, Fahlman E proposed a self-automating and lighter processing, data-relationship driven neural network architecture termed “Cascade Correlation” (CasCor) [3]. A CasCor neural network is trained in the way that hidden layers will cascade at each step, with one hidden neuron at each layer. The connection weight values from the input to the hidden neurons will remain unchanged, termed “frozen”, to retain the hidden neuron as a permanent feature set. This allowed neural networks to be light-weight and broadened automatically.

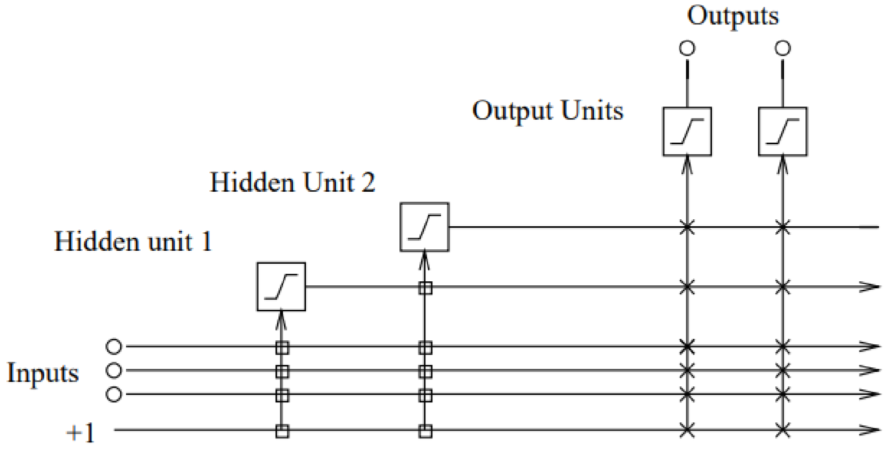

Fig. 1. A conceptual graph depicting frozen and live weights between neurons. Each interjection between two lines is the weight shared for the corresponding connections. The icons on the interjection represents a different configuration of the weights. A box represents the weight is “frozen”; a cross represents the weight is “live”.

A CasCor network model begins with a fully connected network containing only input and output neurons. They are connected by weights that can be modified in back propagation, called “live” weights. The model is trained ordinarily until improvement becomes stagnant. A pool of candidate neurons with distinct incoming weights will be trained in the model with two conditions: 1) the new hidden neuron will only be connected to the previously trained hidden neurons and the input (no connections to the output), 2) the loss function, denoted as S, is calculated by the correlation between the activation of the candidate node V and the residual output error E. This residual error is the difference between the sum of the output of the network and the target value. We define S as:

$$S = \sum_{o} \left| \sum_{p} (V_p - \overline{V})(E_{p,o} - \overline{E_o}) \right| \tag{1}$$

A backward pass on S completes the epoch for the candidate model. Until there is no improvement in S of each candidate nodes, a permanent hidden neuron will be injected to the original network, using the best candidate node in the pool. All connections to this hidden neuron will be frozen, and then the training process will repeat. As a result, CasCor network is able to significantly improve the speed and efficiency for convergence in deep learning, by reducing training redundancy in frozen connections [3].

1.2 Dropout Technique

Srivastava proposed the dropout technique in 2013, where zeroing neurons (and its connections) with random probabilities can address overfitting and slow networks [4]. When a unit is randomly dropped from the network, it prevents neurons forming an excessive reliance over neighbouring units.

During a training epoch, hidden neurons (or sometimes input neurons) will be randomly removed temporarily from the network, termed a “thinned” neural network. The thinned neural network takes a similar approach to the theory outlining the role of sex in evolution. Sexual reproduction creates offspring that takes the genes of their parents. Rather than a random pairing of partner, offspring with the gene copy of a successful parent may exploit the survivability for natural selection. Another method to increase survivability, the mixability of genes, where multiple genes work together in collaboration can make genes robust together, however, they are flawed on its own. Hence, by occasionally thinning some genes in the group, no genes must rely on their exact partner to be present at all times and must learn to mitigate the shortcomings. Similarly, the dropout technique ensures that each neuron is robust on its own and own an important role even without the neighbouring units, to regularize each unit to make it generalizable.

The resulted problem may be noisy, however more adaptive, and more efficient. Removing units may induce noise, however, it strengthens the individual neurons to become independent. It trains their flexibility to adapt to unexpected situations, while training less neurons at each epoch.

1.3 Binary Prediction Neural Networks

Further from architectures and techniques, using a more optimised argument to be passed through the network can reduce the passing redundancy while improving weight update relevancy. With modification of CasCor, a binary prediction oriented neural network is hypothesized to 1) improve efficiency while maintaining the same accuracy, and 2) have a higher testing accuracy on average, when compared to a baseline CasCor network and a baseline fully connected network.

To limit the needs to calculate poly-dimensional data, only a representative single value passing from each instance should be used. It is computationally more efficient than a multi-value passing for each instance, for example passing a 1 instead of 2 dimensional tensor.

The binary prediction oriented architecture should speed up the convergence, meanwhile improving the accuracy on average. In comparison with training multiple neurons at once, training one hidden neuron at a time benefits from a focused loss reduction process, where loss can be reduced by receiving more focused input signals generated from fewer other hidden neurons.

1.4 Eye Gaze Dataset

The eye gaze data is collected over 80 participants to investigate the ability for humans to perceive manipulated images [5]. For all images, each participant is provided context for image manipulation techniques, and then presented with images. A series of questions were asked to evaluate their thought processes, and then the participants would vote whether the image was manipulated or not. Eye gazing data is tracked throughout the image viewing processes and recorded.

The dataset contains 8 features for image manipulation prediction from eye gaze. Ground truth of the image manipulation, in addition to supporting data related to eye gaze is included in the dataset. Each participant and image sets contain a unique ID for identification. Eye gaze fixation count and duration on the image and at the target area were recorded.

Eye gaze data was captured using two Facelab 5.0.2 infra-red cameras and a single IR light emitter pod, that is centrally located below the image displaying monitors. Video evidence of each experiment was recorded using Eyeworks v3.8, including for the use to design, deliver, and analysis the experiment.

2 Method

2.1 Cascade Correlation with One dimensional pass on Binary prediction (CasCorOB)

Cascade Correlation with One dimensional pass on Binary prediction (CasCorOB) architecture is proposed in this paper to make binary predictions. In additional to the aforementioned CasCor architecture, CasCorOB model is 1) passing a one-dimensional value for each neuron output, 2) adopting hyperbolic tangent activation functions, and 3) utilizing the Pearson's correlation in replace with S from formula (1) as loss function.

Output classes are represented by the range of a single value rather than a SoftMax matrix. It should reduce processing performance overhead by passing simpler values. The binary class is represented by a value range, where values are averaged to 0, meaning small values account to closer to average and large values are further from average. In image manipulation, a negative value is false and positive value is truth.

A hyperbolic tangent activation function is used in parallel with the value range for weight update relevancy. In comparison with a sigmoid activation function, it mitigates exploding or diminishing weights by positioning the average at 0, meanwhile retaining the binary property of representing 2 prediction classes [6].

Correlation is adapted in place of the suggested "correlation" loss function that was mentioned in the Fahlman's paper. In fact, the correlation mentioned in the paper is a covariant, which simply evaluates the positive and negative trends. This is however not useful in determining the numerical difference from the true value trend, i.e. how close the predictions are to the actual trend line. Pearson's correlation achieves this instead.

2.2 CasCor on Binary prediction with Dropout (CasCorBD)

CasCorOB with Dropout technique (CasCorBD) architecture is another proposal in this paper to make further training efficiency for binary predictions. CasCorBD builds upon the CasCorOB architecture, such that network training speed is further improved and can be generalized to lower the hidden neurons required.

Dropout takes place in the general training network, which is separate from the candidate training process. While the partial network is frozen, some inputs from the input and hidden neurons can be removed of the network to further reduce processing power required on each epoch. Meanwhile, the loss of certain neurons will induce more losses and create noise. Due to the more frequent loss from any part of the original the network, the output neurons are strengthened to be more robust and less lenient from other neurons. The resulting network would be regularized and require fewer hidden neurons. CasCorBD will require more evaluations and k-fold validations as a result of the increased noise.

To demonstrate the performance of a binary prediction oriented network on an image manipulation dataset, a three layer fully connected network and a two output CasCor network will be used as a baseline. It will aim to measure the accuracy, epochs counts, elapsed time, and binary prediction oriented techniques of each neural network, for predicting image manipulation from an eye gazing behaviour.

2.2 Data Pre-processing

The collected data was entirely numeric, including the truth value for the image manipulation in each instance. However, the data is distributed vastly between each feature. A z-score normalization is used to correct the imbalance and confines the outliers. While retaining the scaling between the data, it also shifts the average to zero for better distinction. As a result, it initializes the higher than average values to be positive, while lower than average values to be negative, which helps long-term scaling of the values in the long run in a neural network.

The features set contains 7 independent variables in total – 4 useful variables and 3 inessential variables. The adopted features are "num_fixs" for number of eye gaze fixations of image fixations, "fix_dur" for eye gaze fixation durations in seconds of image fixations, "num_man_fixs" for number of manipulated area fixations, "man_fixs_dur" for fixation durations in seconds of manipulated area fixations. The discarded variables are "vote" for participant's vote, "participant" for participant ID, and "image" for test image ID, which were discarded because they are keepsake datasets and do no provide indications for image manipulation prediction.

The dependent variable "image_manipulated" is an integer to indicate whether the corresponding independent variables are a response to an original image ("0") or a manipulated image ("1").

Table 1. Data distribution of its corresponding features. Excluding feature vote, participant, and image.

Feature Name	Minimum	Maximum	Average
num_fixs	6	215	81.8
fix_dur	0.97	63.5	18.9
num_man_fixs	0	115	16.1
man_fixs_dur	0	32.3	12
image_manipulated	0	1	0.5

The overall dataset is split in an 8:2 ratio for training data and testing data. A mix of 5 and 10-fold validation is used to evaluate the accuracy.

2.3 PyTorch Framework

PyTorch is an optimized, open sourced tensor library for deep learning. It provides developers and researchers a higher level accessibility to construct, assess, and evaluate neural network models [7]. It helps maintaining code readability especially in a neural network with a convoluted structure. Furthermore, a comprehensive documentation and PyTorch forums are easily accessible online, hence it is chosen to build the CasCorOB network.

2.4 Machine Specifications

Each neural network is trained and tested on a machine with a 4-core 8-thread 4.5Ghz CPU and a 16GB 2133MHz memory. AI processor with 272-Tensor Cores is used in the RTX3080 GPU in the machine.

3 Results

The eye gaze data is trained in the proposed CasCorBD and CasCorOB, with the baseline two outputs CasCor and three layer fully connected network ("FullyConnect").

Table 2. Hyperparameters used for a CasCorOB, CasCorBD, CasCor, and FullyConnect

	Epochs (max)	Hidden Neurons	Learning Rate	Candidate training learning rate	Input Neurons	Output Neurons	Activation Function
CasCorBD	100	5	0.01	0.1	4	1	tanh
CasCorOB	100	7	0.01	0.1	4	1	tanh
CasCor	100	6	0.01	0.1	4	2	leaky ReLU
FullyConnect	100	7	0.01	-	4	1	tanh

* tanh = hyperbolic tangent

All networks are initialized with 4 input neurons and Adam optimizer is used. The networks are capped at a more than required number of epochs, 100, for hard limiting. Stagnant loss improvement is determined by a within 1% change and network is pre-emptively terminated before maximum epochs when it is stagnant.

In CasCor, CasCorOB, and CasCorBD, each candidate pool contains 15 candidates, and the learning rate for the candidate training is higher, at 0.1 for exploration.

Hyperparameters for the (maximum) number of hidden neurons of all networks, and the dropout rate for CasCorBD are measured through statistical testing to find each of their best results. The number of hidden neurons is tested against 10 values: [1, 2, ...,10] and dropout rate is tested against 6 values: [0.5, 0.6, ...,1.0]. For CasCorOB, CasCor, and FullyConnect, each value of the two hyperparameters are tested against 5 evaluations, each with a 5-fold validation; for CasCorBD, it is 10 evaluations, each with a 10-fold validation instead for stricter measurements to combat higher noise.

Table 3. Averaged Testing Accuracy

Hidden Neurons	CasCorBD, %	CasCorOB, %	CasCor, %	FullyConnect, %
1	62.98	60.27	47.45	49.84
2	64.80	62.98	48.26	50.27
3	66.25	63.28	51.07	52.74
4	66.30	61.11	48.77	53.83
5	**67.81**	60.11	50.12	54.58
6	65.40	60.30	**53.23**	52.76
7	64.86	**63.62**	50.37	**54.74**
8	64.12	57.89	48.60	52.64
9	63.09	61.45	51.64	52.65
10	63.93	60.87	47.58	53.02
Avg	64.95	61.19	49.71	52.71

After statistical testing, 5, 7, 6, 7 (maximum) hidden neurons were chosen for CasCorBD, CasCorOB, CasCor, and FullyConnect respectively for the best testing results.

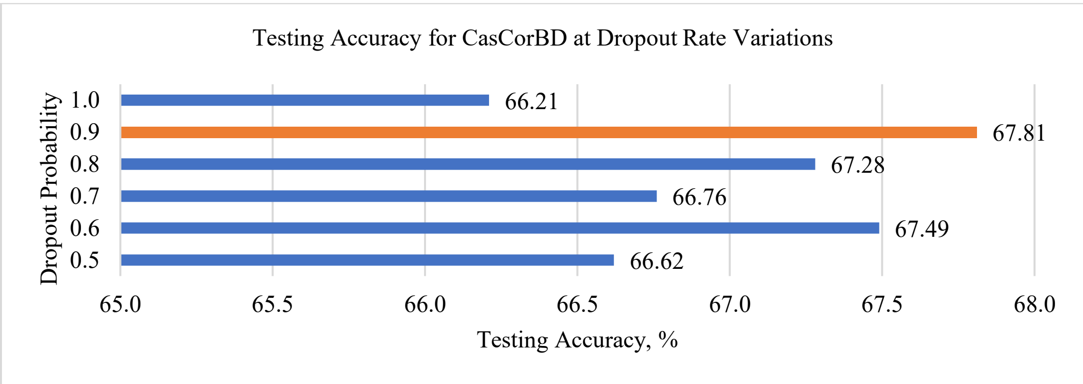

Fig. 2. CasCorBD dropout probability (y axis) against testing accuracy (x axis). Each value is tested against 10 evaluations, with 10-fold validation on each evaluation.

The dropout rate at 0.9 in CasCorBD proves to provide the best testing accuracy, ahead of the wost performing dropout probability at 1.0 by a -2.36% edge.

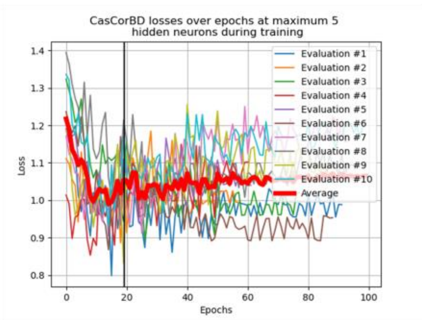

Fig. 3. Best CasCorBD network at 5 hidden neurons with 10 evaluations each averaged over 10-fold validation.

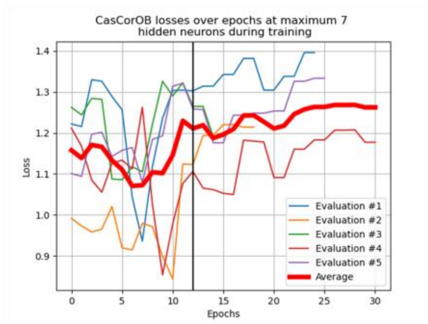

Fig. 4. Best CasCorOB network at 7 hidden neurons with 5 evaluations each averaged over 5-fold validation.

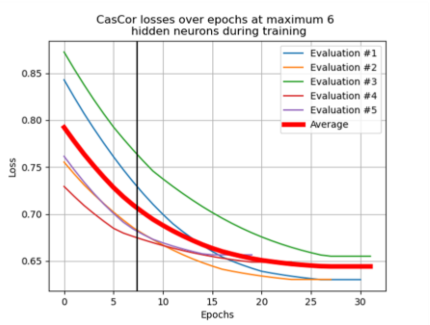

Fig. 5. Best CasCor network at 6 hidden neurons with 5 evaluations each averaged over 5-fold validation.

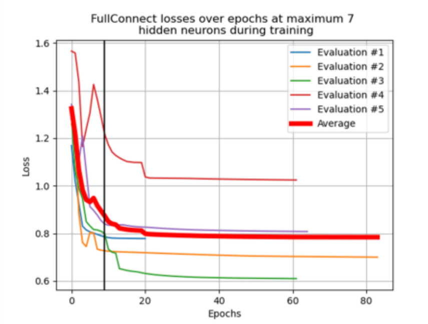

Fig. 6. Best FullyConnect network at 7 hidden neurons with 5 evaluations each averaged over 5-fold validation.

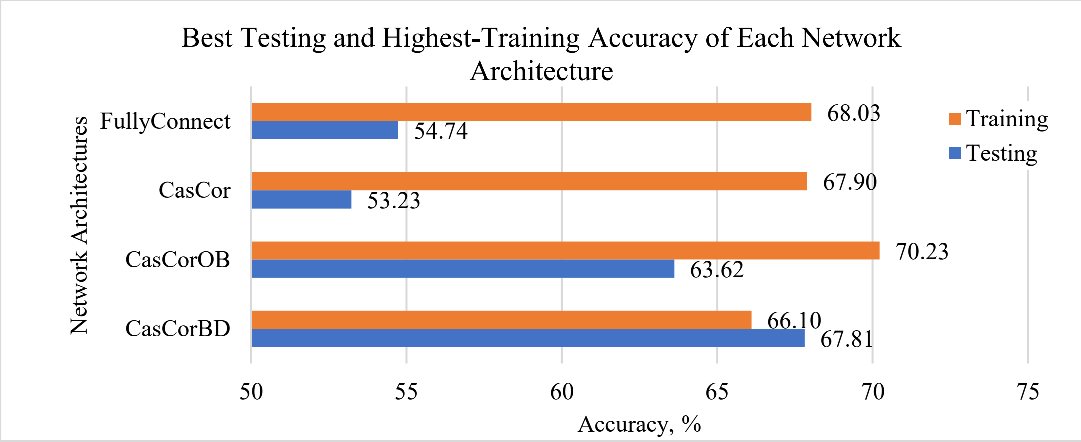

Fig. 7. Averaged best testing accuracies and highest-training (of each evaluation) results of each the network architectures. Best as defined by the average highest testing accuracy for some hidden neuron counts of its own architecture (5, 7, 6, 7 for CasCorBD, CasCorOB, CasCor, and FullyConnect).

CasCorBD is best at image manipulation detection at a testing accuracy of 67.81%, whereas CasCorOB has the greatest (highest-) training accuracy of 70.32%, on average. The lowest testing accuracy is 53.23% from Cascor, meanwhile the lowest training accuracy is 66.10% from CasCorBD. CasCor training and testing accuracy percentage difference is 14.67, FullyConnect is 13.29, CasCorOB is 6.61, and CasCorBD is 1.71.

Table 4. Precision and Recall of the Best networks evaluations

	CasCorBD	CasCorOB	CasCor	FullyConnect
Precision	0.65	0.62	0.54	0.56
Call	0.80	0.72	0.57	0.48

CasCorBD is the strictly the most relevant architecture, which returns the most relevant and most of the relevant predictions at 0.65 precision and 0.80 call. This can also be seen from the confusion matrices in Appendix 2, 3, 4, and 5. CasCorOB is strictly the second. FullyConnect returns most relevant results at 0.56 than CasCor at 0.54 precision. CasCor returns most of the relevant results at 0.57 than FullyConnect at 0.48 call.

Table 5. Efficiency for Each Evaluation

	CasCorBD	CasCorOB	CasCor	FullyConnect
Fewest epochs	19	7	8	9
Seconds, s	4.48	15.02	175.35	0.27

The best CasCorBD, CasCorOB, CasCor, and Fully connect networks required 19, 7, 8, 9 epochs respectively for each evaluation. However, in a more realistic view, it takes 4.48, 15.02, 175.32, and 0.27 seconds respectively to train the model on each evaluation, 5-fold each. FullyConnect required the least time for each evaluation, while CasCor required the most. CasCorOB required only 8.57% of the time CasCor used to train the network. Meanwhile, CasCorBD required only 29.83% of the time CasCorOB used to train the network.

Table 6. Percentage Difference in CasCorBD from CasCorOB, CasCor, and FullyConnect.

	from CasCorOB, %	from CasCor, %	from FullyConnect, %
Testing Accuracy	6.59	27.40	23.89
Fewest Epoch count	171.43	-12.50	-11.11
Seconds	-70.18	-97.45	1571.53

CasCorBD is strictly better than CasCorOB and CasCor in a realistic approach, where both test accuracy is higher and time cost is lower to train the network. The most significant improvement for test accuracy is CasCorBD from CasCor, at 27.40%. In the contrary, although CasCorBD is 23.89% more accurate than FullyConnect, it also requires ~15times more seconds to train the network.

4 Discussion

As hypothesised, the binary prediction oriented networks ("BPO networks") – CasCorBD and CasCorOB both performed better than the baselines – two output CasCor and three layer network (FullyConnect) in regard to testing accuracy as seen in Table 6. CasCorBD even outperformed CasCorOB in generalizability, fewer hidden neuron required, and training time required as a result of the self-induced loss technique from network thinning, as shown in Table 3 and 6. This process had made the output neuron connections to be more robust, such that they are better at mitigating knowledge gaps from not yet known inputs. Although CasCorBD had noisier data as seen in Fig 3 compared to Fig 4, 5, and 6, the testing accuracy as shown in Fig 7 have an inversed expectation where testing is better than the training accuracy.

Regardless, the BPO networks performed more accurately than CasCor due to either the change in loss function or the 0 averaged value passing. As suggested, the Pearson's correlation loss provides more relevant value differences between the candidate to the target loss, whereas the 0 averaged values mitigate the exploding gradients. While the evidence clearly implies a flat 14.58% and 10.39% improvement from such changes, it is however unclear what are the distribution of the benefits between the two changes.

The efficiency improvement is greater, however debatable, from BPO networks against the baseline networks. The number of epochs required is sometimes way worse, where CasCorBD required around twice the amounts of epochs as others due to the unsettling loss noise. However, in a more realistic sense, the BPO networks performed faster convergence from CasCor. From CasCor to CasCorOB, passing a one-dimensional value in the network, as opposed to multi-dimension proved to improve the speed by 91.43%. From CasCorOB to CasCorBD, applying the dropout technique proved to improve the speed by 70.18%. CasCorBD was not able to outperform FullyConnect in speed, at ~15times slower, however the performance gain may be of more significant value.

In certain cases, epoch count maybe high, but time taken to train is low as shown in figure 3, 4, 5, 6 and table 5. It shows that CasCorBD is extremely efficient in each epoch as opposed to CasCorOB and CasCor. CasCorBD training time efficiency cannot be compared to FullyConnect because the latter could not reach the reasonably same level of prediction accuracy. Regardless, each CasCorBD, CasCorOB, and CasCor epoch is evidently more computationally intensive compared to the FullyConnect neural network.

While participants can predict an image manipulation with 56.0% accuracy on average, CasCorBD was able to make a prediction at 67.81% accuracy on average, based on the participants' eye gazing behaviour. CasCorBD was able to indirectly predict image manipulation better than a human using first-hand information.

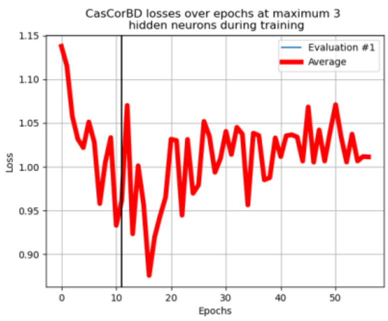

Fig. 8. Example of CasCorBD overtraining at 20 hidden neurons.

CasCorBD is less prone to local minimum (loss) as a result of spikes. In Fig 6, the evaluation #4 (red line) in a fully connected network shows the stuck loss descent at a local minimum, which as a result converges into a poor loss value. Better loss values are evident as other evaluations of the same architecture is able to achieve lower loss values. This is also observable by the large distribution of testing accuracies from the 10 evaluations, as shown in Appendix 2. The variability of the network from dropouts in CasCorBD has shown to overcome this issue. However, the loss value is also noisier at a greater magnitude, where the peak and the troughs of the spikes are distributed at most ±0.1 and stabilized to ±0.05 every 3-5 epochs, as seen from Fig 8 as an example. Similar issue can be seen in the CasCorOB network from Fig 4, but not in CasCor network from Fig 5, suggesting that random candidate unit weight initialization may not be the cause. As a result, the loss becomes difficult to converge, as the dropout of neurons across the net is unpredictable and the suitability of each additional neuron relies on the random weight distribution for the pool of candidates.

5 Future Work

CasCorBD requires further research to support the belief in performance boost. It requires rigorous testing on other datasets that could prove the claim for its better performance against both CasCor and a traditional 3 layer fully connected network.

To finetune the network, excluding the already tested dropout rate and the hidden neuron count, other hyperparameters can be modified for general performance boost. For example, the number of candidates and learning rates may be lowered to reduce processor performance overhead to help with efficiency.

Understanding the Pearson's correlation and the zero averaged variables may reveal the performance benefits of each change made from the CasCor to CasCorOB architectures. This can be approached by training two versions of the CasCor architectures, each with the separate changes. It may help deduce the benefit distributions between the two techniques.

In addition, the dropout rate for each neuron may be modified to vary differently at different connections of the network. A higher dropout rate for earlier hidden neurons may encourage newer hidden neurons to be less reliant to the significant feature sets (from the older hidden neurons). As a result, this greater independency may reduce large loss spikes as seen in the BPO networks – Fig 3 and Fig 4.

Nevertheless, further research for CasCorBD is worthwhile, reasoning to its more generalizable and concurrent property. The architecture is not only able to make a more accurate prediction than a human, but also make an accurate prediction simultaneous to a decision making process by a human, all while the network is retrained and adaptable to the user's own profile. This could be applicable to cybernetics and bionics, including smartphones and XR devices, as accurate and fast predictions of the user intentions would better personalize and support the user experience. It should be noted that no computer vision should be required to perform well, although a labelled dataset is required. As demonstrated by the eye gaze behavioural data, the second hand information is already sufficient to predict image manipulation more accurately than a human with first-hand information.

References

1. Lemley, J., Bazrafkan, S., & Corcoran, P. (2017). Deep Learning for Consumer Devices and Services: Pushing the limits for machine learning, artificial intelligence, and computer vision. *IEEE Consumer Electronics Magazine*, *6*(2), 48-56.
2. Alsop, T. (2020). *AR/VR investment worldwide by use case 2024*. Statista. Retrieved 30 May 2021, from https://www.statista.com/statistics/1098345/worldwide-ar-vr-investment-use-case/.
2. Fahlman, S. E., & Lebiere, C. (1990). *The cascade-correlation learning architecture*. CARNEGIE-MELLON UNIV PITTSBURGH PA SCHOOL OF COMPUTER SCIENCE.
3. Fahlman, S. E. (1991). *The recurrent cascade-correlation architecture*. Carnegie-Mellon University. Department of Computer Science.
4. Srivastava, N. (2013). Improving neural networks with dropout. *University of Toronto*, *182*(566), 7.
5. Caldwell, S., Gedeon, T., Jones, R., & Copeland, L. (2015, July). Imperfect understandings: a grounded theory and eye gaze investigation of human perceptions of manipulated and unmanipulated digital images. In *Proceedings of the World Congress on Electrical Engineering and Computer Systems and Science* (Vol. 308).
6. Nwankpa, C., Ijomah, W., Gachagan, A., & Marshall, S. (2018). Activation functions: Comparison of trends in practice and research for deep learning. *arXiv preprint arXiv:1811.03378*.
7. Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., ... & Lerer, A. (2017). Automatic differentiation in pytorch.

Appendix

Appendix 1. Full testing accuracy against different (max) hidden neuron counts on CasCorBD, CasCorOB, CasCor and FullyConnect architecture.

Hidden Neurons	CasCorBD, %	CasCorOB, %	CasCor, %	FullyConnect, %
1	63.74	67.29	66.58	57.35
2	61.98	64.83	67.11	62.45
3	**67.77**	67.16	68.12	62.96
4	65.12	66.26	67.03	65.95
5	66.10	**70.32**	68.05	67.45
6	65.66	66.69	67.90	61.85
7	63.39	70.23	66.80	**68.03**
8	67.16	68.48	**69.20**	66.85
9	65.35	68.88	66.38	65.66
10	64.45	69.83	67.63	66.69
Avg	65.07	68.00	67.48	64.52

Appendix 2. Best FullyConnect Confusion Matrix

	Predicted Positive	Predicted Negative
True Positive	452	493
True Negative	349	566

Appendix 3. Best CasCor Confusion Matrix

	Predicted Positive	Predicted Negative
True Positive	534	411
True Negative	459	456

Appendix 4. Best CasCorOB Confusion Matrix

	Predicted Positive	Predicted Negative
True Positive	682	263
True Negative	413	502

Appendix 5. Best CasCorBD Confusion Matrix

	Predicted Positive	Predicted Negative
True Positive	755	190
True Negative	409	506