

# Exploring and Evaluating Constructive Cascade Neural Networks in Classification Problems

Yanxi Lu

College of Engineering and Computer Science, Australian National University  
u5952700@anu.edu.au

**Abstract.** Classification is one of the most frequently met decision tasks of human activity. Neural network techniques of varying complexity such as multilayer feed-forward network[1] and deep convolutional networks[2] have achieved notable success in performing these tasks. However, the success of these approaches depends heavily on choosing the right network topology beforehand, which is often hard for complex problems. Constructive algorithms, which construct the neural network dynamically during training, have the potential to produce networks matching the inherent complexity of the problem. In this paper, a constructive cascade network is proposed for a classification problem involving predicting whether a picture is manipulated given human eye gaze on it. The focus of investigation lies on the network structure generated. Performance benchmarks including convergence speed and prediction accuracy are compared against multilayer feed-forward networks. The results confirm that constructive cascade networks have good generalization ability and network construction properties at the expense of longer training time. Based on these observations, it is reasonable to assume that constructive cascade network technique can be combined with other techniques. By first finding a reasonably good network topology, advanced learning strategies can be applied more effectively for all kinds of tasks.

**Keywords.** Neural network, Constructive cascade network, Classification problem

## 1 Introduction

Classification Problems have been one of the most actively researched domains due to its presence in daily life. Statisticians have developed numerous approaches to solve the problem. However, Traditional statistical approaches such as discriminant analysis and generative models are limited by their assumption of the underlying probability model[3]. Neural networks, given their adaptive learning nature, is perfect for general classification problems when there is little information about the data[4]. Among the successes, multilayer feed-forward networks trained with error back-propagation is one of the most well-known algorithms. Nevertheless, it is often limited to searching for the set of weights in a fixed network topology[5], which requires the selection of an appropriate network beforehand.

Unfortunately, there is no easy way to determine the optimal structure. Undersized models may not learn the problem well and lead to under-fitting while oversized ones tend to overfit and fail to give a proper generalization of the problem.

There are two main approaches to estimate the appropriate size of a neural network. One is pruning, which begins with an oversized network and reduces its size by eliminating hidden neurons[6]. One limitation of pruning is the difficulty in deciding what is a big enough network for a problem. If the model complexity is not enough, pruning on the model may lead to worse under-fitting. The other approach is constructive algorithms. They incrementally build a minimal network during training until a satisfactory topology is found. Constructive algorithms avoid the size initialization problem and potentially take less time to train than oversized networks in pruning.

This paper focuses on constructive cascade network[7], one of the constructive algorithms. It is applied to the classification problem to predict if an image is manipulated given experiment data of how human participants view the images with their eye gaze[8]. By analyzing the properties and performance of the network and comparing to typical

multilayer feed-forward networks, this research primarily aims to investigate the resultant network structure and generalization ability of the technique on classification problems. Positive results would present the potential of constructive cascade network being used in hybrid with other techniques. Solving more complex problems can be potentially easier through first obtaining the prior knowledge of the problems' approximate complexity.

## 2 Method

### 2.1 Data Set

The data set used is originally proposed by Caldwell et al. based on her eye-gaze tracking experiment of 80 participants viewing a combination of manipulated and unmanipulated images[8][9]. The data set contains 372 lines of data, each containing 8 features. Feature 1 is the participant's id. Feature 2 to 5 describe eye gaze of the participant when viewing the image, the eye gaze is represented by the number of fixations and duration the participants spend looking at the whole picture and regions of manipulation respectively. The last three features indicate the picture id, whether the picture is manipulated and the vote by the participant on whether the picture is manipulated.

In this classification task, the goal is to determine whether the participant is looking at a manipulated image based on their eye gaze on it. Hence, the first five features are taken as inputs and the feature indicating if the image is manipulated is the output.

All data used for this classification task is in numerical form but with different domains. The output feature uses 1 to denote a manipulated image and 0 otherwise. While for the five input features, a wide range of values is observed. Sola and Sevilla [10] points out that data normalization is crucial to obtaining good results as well as to accelerate the learning process. The input data is hence normalized using z-score normalization[11] given by formula (1). It is

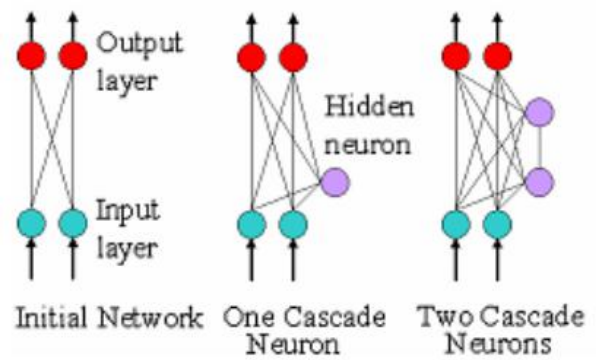
$$z = \frac{x - \mu}{\sigma} \quad (1)$$

worth noting that this technique is sensitive to outliers; hence outlier removal strategies can be used to improve the classification result potentially.

The data set is randomly split into a training set, a validation set and a test set with a 6:2:2 ratio based on the network's performance in experiments.

### 2.2 Neural Network Topology

Typical constructive algorithms such as Cascade Correlation(CasCor) and Casper often starts with a minimal network structure containing only a pair of fully-connected input and output layers as shown in Fig 1[7]. During training, hidden neurons are dynamically added [12]. In the case of each new hidden neuron forming a new layer, each neuron added is connected fully to the input and output layer, as well as all the hidden neurons before. The process continues until a satisfactory solution is found.



**Fig 1** CasCor

The technique adopted in this paper is a slight modification of the algorithms mentioned above. Instead of adding one hidden neuron, a cascade layer containing several hidden neurons is added each time. The rationale of doing so is to speed up the convergence at the expense of higher computational cost each layer. Also, adding some complexity to the network can potentially break local maximas at early stages of the algorithm when the network structure is simple. Similar to CasCor, these newly added cascades are fully connected to all existing layers in the network. The number of hidden neurons each layer is set to 3 so that it is smaller than the number of inputs in order to restrict the functionality

of each cascade layer and reduce the total number of hidden neurons[7].

### 2.3 Training Methodology

The methodology adopted is tuned to fit the purpose of this training task. The weights and bias of the initial network is initialized to small random values by default. The same goes for each new cascade layer. The activation function used after each hidden layer is sigmoid and binary cross entropy loss is used for evaluation as this is a two-class classification problem in the range  $[0,1]$ . Resilient propagation(RPROP) [13] is used for learning. The main difference between RPROP and traditional back-propagation is that RPROP uses the sign of gradients only. This allows the learning process to become smoother and less prone to outliers. The adoption of RPROP in Casper has been proven successful for producing smaller networks and better generalization than CasCor.

During training, the network starts from the initial state containing only the input and output layers for a maximum of 800 epochs. A new cascade layer is added whenever the maximum number epoch is reached or the training error is smaller than 0.2. The values are chosen based on prior testing with typical multilayer networks. Each time when a new cascade needs to be added, the network is used on the validation set to measure the validation error. The network construction halts when either the maximum number of cascades is reached or validation error fails to decrease after the addition of two more cascades. The exact halting criterion is set to accommodate the uncertainty in the correlation between validation and test set. However, if the validation error fails to decrease after two more cascades, it is likely that learning is complete. Both addition and halt criteria adopt an early stopping approach which has been proven to be able to increase network generalization ability[14][15]. The output layer comprises 2 units for the classes manipulated and unmanipulated, and a winner-takes-all strategy is used to decide the output of the network.

## 3 Results and Discussion

The constructive cascade network is tested on the classification task described in Section 2.1. As mentioned before, the key interests of investigation lie in the structural property of the resultant network as well as the generalization ability on the task.

### 3.1 Structural property of the network

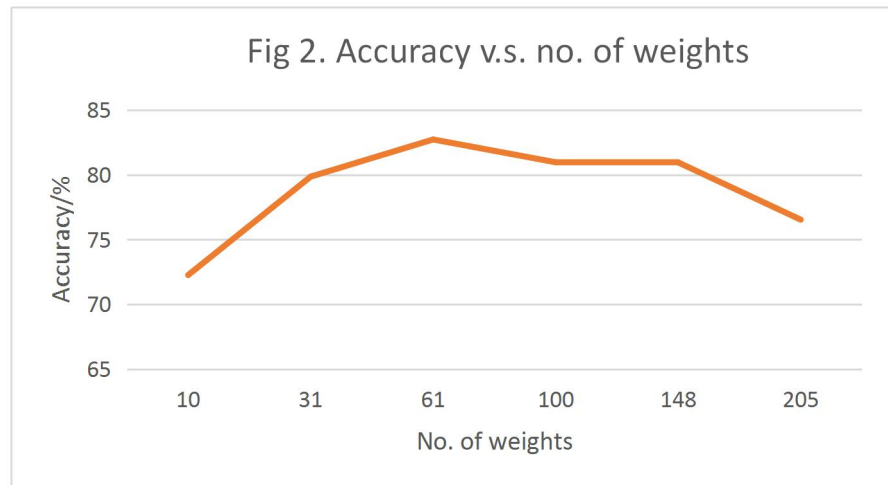
The property of the network is analyzed based on its performance on the task. Table 1 shows the average performances of the network on the classification task with learning rate 0.01 and a maximum of 5 cascade layers over 20 runs.

**Table 1:** Average performance on the classification task over 20 runs

Cascade layers	No. of weights	Generalization performance % (only counts results exceeding the number of layers)	No. of times halting with this structure (average of 2*20 runs)
Initial network	10	72.25	0
1	31	79.85	0
2	61	82.72	6
3	100	80.96	5
4	148	80.95	5
5	205	76.53	4

As shown in the table, an increase in one layer of network leads to around an addition of  $(n+1)*10$  weights, where  $n$  is the number of cascades in the network. The testing accuracy mostly lies in the range of 75% to 85%. The average testing accuracy over 20 runs is plotted in Fig 2. against the complexity of the network generated denoted by the number of weights to learn. From the figure, some patterns can be observed. The curve is a convex upward, with its maximum at 61 weights, which correspond to 2 cascade layers in the network. Beyond that, the accuracy decreases despite the increase in the network complexity, potentially indicating overfitting. The result with 0, 1 and 2 cascade

layers is more interesting. From 0 cascade layer to 1, accuracy increases around 8%, or equivalently a 28% decrease in error rate at the expense of a 200% increase in the number of weights. From 1 cascade layer to 2, a 12% decrease in error rate costs 100% increase in the number of weights. The tradeoff seems too heavy to bear but does present a decent result. The choice should be up to the precision requirement of the task.



There is one more important observation from the result. Sometimes the training loss is stuck until the next cascade is added. The problem usually happens when the number of cascades is very small, hence likely to be caused by local maximas. Constructively adding cascades help break local maximas.

More experiments are done by tuning the parameters including setting the initial number o cascade layers to 1 and adjusting the learning rate, the result is the same as in Fig 2. This could mean that the network with 2 cascade layers is reasonably good for this classification problem. To confirm this, a comparison is made between multilayer adopting a similar structure and random initial structures.

### 3.2 Comparing against typical neural networks

The result obtained above is compared to typical neural networks. The result Sabrina presented in her thesis on this classification problem[9] adopts a neural network designed with a feed-forward back-propagation algorithm. The overall accuracy of successfully classifying the pictures is 66%. The constructive cascade network seems to have much stronger generalization ability than the one proposed.

To validate the conclusion, the constructive neural network is compared with several other multilayer feed-forward networks. The networks used for comparison have two hidden layers with different number of hidden neurons. The learning rate is 0.01 and the number of epochs is adjusted accordingly to show the best performance of each case. Training set now takes 80% of the original data set to give the networks more training resource. Table 2 presents the average of 5 best outcomes of 20 runs.

**Table 2:** Average of best 5 performances on the classification task by typical networks

Hidden neuron number In layer1, layer2	Generalization performance % (5 best runs)	Remark
1,1	51.22	All of these configurations get stuck in local maximas readily.
3,3	75.73	
5,5	66.15	
10,10	76.06	

From table 2, it is clearly shown that the best performance of a 3-layer network described is below 80%, which is worse than the performance of the constructive cascade network. It is also worth noting that the structure similar to the result optimal structure found before, i.e., two hidden layers with 3 neurons each performs reasonably well among the tested structures. This observation could potentially support the assumption before that constructive algorithms being capable of modeling the complexity of the problem. However, the result is not consistent since all the network structures examined here run into local maximas occasionally.

## 4 Conclusion and Future Work

This paper investigates the possibility of using constructive cascade networks in classification problems. In general, it produces better results than typical multilayer feed-forward networks with predefined network structure. It has the potential to generate network topology matching the intrinsic complexity of problems. The algorithm also excels at breaking local maximas which is a common problem in neural networks. However, there is a tradeoff between the generalization power and the computational cost; hence the choice should be based on the characteristic of the task.

In the future, I will try to validate the effectiveness of constructive networks on more complex data sets and problems beyond classification. In addition, constructive algorithms involve heavy weight and bias initialization procedures. It is proven that good initialization techniques such as Nguyen-Widrow method[16] can substantially speed up the training process. Hence, adopting these techniques can be especially helpful for constructive algorithms. Furthermore, I will try to explore possibilities of using hybrid techniques such as combining it with convolutional neural network for tasks like image recognition.

## References

1. Huang, G. B., Chen, Y. Q., Babri, H. A.: Classification ability of single hidden layer feedforward neural networks. In: IEEE Transactions on Neural Networks, vol. 11, no. 3, pp. 799-801, May 2000.
2. Krizhevsky, A., Sutskever, I., Hinton, G. E.: ImageNet Classification with Deep Convolutional Neural Networks. In: NIPS Advances in Neural Information Processing Systems 25, pp. 1097-1105, 2012.
3. Zhang, G. P.: Neural networks for classification: a survey. In: IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), vol. 30, no. 4, pp. 451-462, Nov 2000.
4. Benediktsson, J. A., Swain, P. H., Ersoy, O. K.: Neural Network Approaches Versus Statistical Methods In Classification Of Multisource Remote Sensing Data. In :IEEE Transactions on Geoscience and Remote Sensing, vol. 28, no. 4, pp. 540-552, July 1990.
5. Parekh, R., Yang, J., Honavar, V.: Constructive neural-network learning algorithms for pattern classification. In: IEEE Transactions on Neural Networks, vol. 11, no. 2, pp. 436-451, March 2000.
6. Taskin, K., Paul, M.: Assessing Artificial Neural Network Pruning Algorithms. In: 24th Annual Conference and Exhibition of the Remote Sensing Society At: Greenwich, UK 1998
7. Khoo, S., Gedeon, T. Generalisation Performance vs. Architecture Variations in Constructive Cascade Networks. In: Köppen M., Kasabov N., Coghill G. (eds) Advances in Neuro-Information Processing. ICONIP 2008. Lecture Notes in Computer Science, vol 5507. Springer, Berlin, Heidelberg 2009.
8. Caldwell, S., et al.: 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. 2015.
9. Caldwell, S.: Framing digital image credibility: image manipulation problems, perceptions and solutions. In: ANU Theses Open Access.
10. Sola, J., Sevilla, J.: Importance of input data normalization for the application of neural networks to complex industrial problems. In: IEEE Transactions on Nuclear Science, vol. 44, no. 3, pp. 1464-1468, June 1997.
11. Jain, A., Nandakumar, K., Ross, A.: Score normalization in multimodal biometric systems. Pattern recognition, 38(12), pp.2270-2285. 2005
12. Fahlman, S. E., Lebiere, C. The cascade-correlation learning architecture. In Advances in neural information processing systems pp. 524-532 1990.
13. Riedmiller, M., Braun, H.: A Direct Adaptive Method for Faster Backpropagation Learning: The RPROP Algorithm. In: IEEE Int. Conf. on Neural Networks, pp. 586-591 1993.
14. Finnoff, W., Hergert, F., Zimmermann, H.G.: Improving model selection by nonconvergent methods. In: Neural Networks, Vol 6. pp.771-783. 1993.

15. Treadgold, N.K., Gedeon, T.D. : Exploring constructive cascade networks. In: IEEE Transactions on Neural Networks, Vol 10, pp.1335-1350. 1999
16. Corrêa D.C., Levada A.L.M., Saito J.H.: Improving the Learning Speed in 2-Layered LSTM Network by Estimating the Configuration of Hidden Units and Optimizing Weights Initialization. In: Kůrková V., Neruda R., Koutník J. (eds) Artificial Neural Networks - ICANN 2008.