# Constructive Cascade Neural Network with Evolutionary Algorithm

Xuning Tan

The Australian National University
u6792826@anu.edu.au

**Abstract.** ConstrCasc is less common Neural Network architecture that could decide topology itself. In this paper, based on our previous work, we first improve our usage on the data set, then we implement a Constructive Cascade Neural Network[Khoo and Gedeon, 2009], and changes its strategy of deciding new hidden units with evolutionary algorithm. We run a binary classification task on a data set[Chen et al., 2017], which is the same task described in the same paper. The result of original ConstrCasc was improved 20% compared to our last study, We compare the ConstrCasc with evolutionary algorithm with the original one, but introducing the evolutionary algorithm does not bring desired improvement. It even draw 10% back on the original ConstrCasc Network at the end.

**Keywords:** Constructive Cascade Neural Network · Anger · Evolutionary Algorithm · Classification.

## 1 Introduction

Emotions are pretty hard to detect, especially when facial expressions are deliberately faked. Anger is one of 27 natural emotions classified by [Keltner et al., ]. In 2017, Researchers[Chen et al., 2017] has developed an approach to use Machine Learning techniques to detect whether the anger of someone in videos is posed or not, based on observers' physiological signals. Normal classification methods with Neural Networks, requires predefined topology, where weights and thresholds are then trained within it. Setting parameters and structures have always been one of the most obscure aspects of implementing Machine Learning algorithms. Consequently, a network that can decide its own structure can be extra valuable. [Khoo and Gedeon, 2009] proposed Constructive Cascade (or ConstrCasc) Neural Network to train the weights and thresholds of the networks itself automatically. It starts with a small network and automatically trains and adds implicit units, eventually forming a multi-layer structure. The algorithm offers ConstrCasc the ability to learn quickly and decide the number of neurons and the depth itself.
The construction of ConstrCasc is based on training candidate units, each candidate unit would be trained until the correlation between its output and the residual of the network is maximized, before it is added to the Neural Network. The original process of choosing satisfying candidate unit is done with traditional backpropagation, which could possibly result in vanishing gradient, and may stop the new hidden unit from further training. Besides, in the original work, each candidate unit in the pool is trained separately, which seems to be lack of efficiency.
In this paper, to solve the possible vanishing gradient problem and to make it possible to parallel the chosen process of new hidden unit, we would like to see whether evolutionary algorithm could improve the performance of ConstrCasc. We proposed a new variants of ConstrCasc, CCEA, by introducing Evolutionary Algorithms to it's candidate unit search strategy. CCEA uses evolutionary algorithm as an optimization, to maximum the correlation between the output of hidden neurons and the residual error of the network. Then we consider the original ConstrCasc as a baseline, and discuss the pros and cons of the newly proposed Neural Network.

## 2 Method

### 2.1 Technique

ConstrCasc was based on the work of [Fahlman and Lebiere, 1990], where they proposed the Cascade-Correlation(Cascor) learning architecture. Here we impelement the a Cascor network first, and move forward to ConstrCasc step by step. Figure 1 illustrate the basic idea behind ConstrCasc Neural Network. It starts with the simplest network, where there is no hidden layers, each input is directly connected to the output, it is a fully connected layer since we are handling a binary classification, only one unit with Boolean output would be enough. The Network trains all the connections within the fully connected until the error of the net is not decreasing. On the next step, the Network generates a set of so called candidate units that for each unit, it connects all the input units as well as the hidden units(The left more figure of Figure 1 shows the process when there is an existing hidden unit). Then the algorithm leaves the original net alone and trains every candidate units and generate the correlation between the activation of the candidate units and the residual error of the network, it picks the candidate with the highest correlation and add it to the hidden layers, with all the incoming weights frozen(as showed with double lines in the pictures). The difference here is that Cascor adds hidden neurons to the network one by one throughout the learning process, while ConstrCasc some cascade chunks, with size set manually before the training.

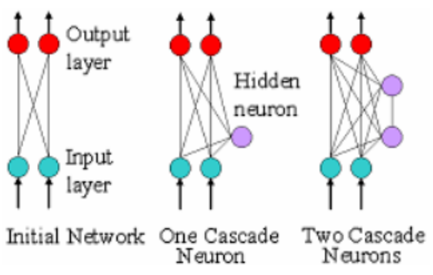


**Fig. 1.** Construction of Cascade Correlation from [Khoo and Gedeon, 2009]

**Candidate Unit** Only the candidate unit with the largest correlation between the activation of the candidate units and the residual error of the network will be selected as the new hidden unit and be added to the existing network, accordingly the calculation of correlation and residual error becomes exceptional important.

*Correlation* The correlation describes how strong the association is between two variables, here we use Pearson Correlation Coefficient to obtain a scaleless statistic.

$$r = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \overline{x})^2(y_i - \overline{y})^2}} \tag{1}$$

*Residual* Residual is the deviation of observed values from fitted values, it reflects the performance of a model's prediction. To get the average residual on a batch, we use Residual Standard Error(RSE) as the criteria.

$$RSE = \sqrt{\frac{1}{n-2}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{2}$$

**Evolutionary Algorithm** Evolutionary algorithm(EA) is inspired by Charles Darwin's theory of natural evolution process. Compared to backpropagation that based on a static algorithm such as stochastic gradient descent, GA is based on random generated numbers. Noting that GA is not usually suitable for a classification task, as stated by [**?**], it is more like a black-box technique that searches solution to maximize or minimize some functions. In this paper, we transformed the binary classification into a optimization task, by using GA to decides the weights of those new added neurons to maximize the correlation between the output of the targeting candidate units and the residual error of the network.
In ConstrCasc, the weights of hidden neurons are changeable during the training, but are frozen as soon as it was added into the Neural Network, the training process of candidate unit is basically deciding fixed input weights of that unit, which means in the actual training process of ConstrCasc, only the output weights of the newly added hidden neuron are being trained. This makes it possible to decide those fixed input weights in ways other than backpropagation, and turns the problem into a optimization, where the target is the maximum the correlation between the output of the hidden unit, and the residual of the current network.

*Fitness Function* Fitness function evaluates how close a given solution is to the optimum solution, here we want to maximize the correlation between the output of a hidden unit, and the residual of the network, our fitness function is exactly the calculation of their correlation.

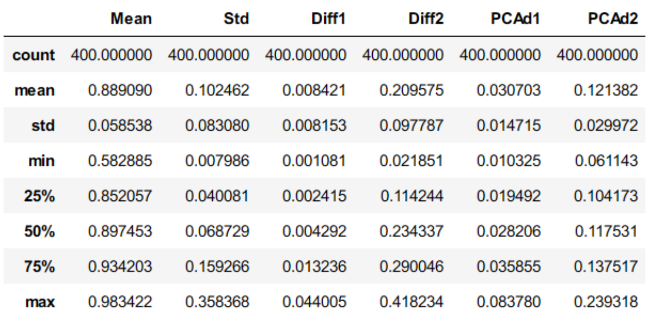
*Selection* In nature, the more adapted an individual is, the more likely it is to reproduce offspring. But one cannot say that the more adapted one is, the more offspring one is sure to have, only more in terms of probability. To establishing this probabilistic relationship, we uses Roulette Wheel Selection.

*Encoding* The input of our approach is the input and output weights of a hidden unit, since the weight is always a float number, here we use a type of float encoding the same as [Péter, 2018] . Suppose we need to find the maximum of $f(\omega)$ where $\omega \in [-10, 10]$, we could divide $[-10, 10]$ into $20 \times 10^6$ sub zones. Since $2^{24} < 20 \times 10^6 < 2^{25}$, we could use a 25-bit binary number to represent all those sub zones. In addition, the binary could be converted into decimal number with

$$f(c) = -10 + c \times \frac{10 - (-10)}{2^{25} - 1}$$

### 2.2 Dataset

The dataset was previously collected and proposed by [Chen et al., 2017], the experiment gathered the verbal response of 22 participants and their papillary response by observing two types of anger stimulation. With trained machine classifiers, they shows that by detecting physiological signals from emotional perceptions, machines can achieve high precision in differentiating genuine anger and acted one. The Dataset is consist of six key elements, the observer number, mean of pupillary response from each observer, the standard deviation of in pupillary response from each observer, the change of left or pupillary size after watching a video, an orthogonal linear transformation with first or second principal component, and whether the anger in the video is geninue or posed.
The dataset has $9 \quad columns \times 400 \quad rows$, where the first two columns describes the observer number and video number. The dataset has $9 \quad columns \times 400 \quad rows$, where the first two columns describes the observer number and video number. Different to our previous work, this time we make usage of the column "Video", by ordering it from 1 to 20, where the first 10 are T1-T10, the last 10 are F1-F10. Including the Observers would strangely leads to a 100% accuracy at the beginning of training and prevent new hidden units from adding, thus column of "Observer" was dropped. The label is recorded as "Posed" or "Genuine", it is unreadable for out Nerual Network, since the network can only run mathematical operations, we need to transform "Genuine" and "Posed" before load them to the network. As figure 2 shows, the dataset is not well-balanced, and therefore



| | Mean | Std | Diff1 | Diff2 | PCAd1 | PCAd2 |
|---|---|---|---|---|---|---|
| **count** | 400.000000 | 400.000000 | 400.000000 | 400.000000 | 400.000000 | 400.000000 |
| **mean** | 0.889090 | 0.102462 | 0.008421 | 0.209575 | 0.030703 | 0.121382 |
| **std** | 0.058538 | 0.083080 | 0.008153 | 0.097787 | 0.014715 | 0.029972 |
| **min** | 0.582885 | 0.007986 | 0.001081 | 0.021851 | 0.010325 | 0.061143 |
| **25%** | 0.852057 | 0.040081 | 0.002415 | 0.114244 | 0.019492 | 0.104173 |
| **50%** | 0.897453 | 0.068729 | 0.004292 | 0.234337 | 0.028206 | 0.117531 |
| **75%** | 0.934203 | 0.159266 | 0.013236 | 0.290046 | 0.035855 | 0.137517 |
| **max** | 0.983422 | 0.358368 | 0.044005 | 0.418234 | 0.083780 | 0.239318 |

**Fig. 2.** Anger dataset stastics

a normalization is required. We normalized and ramdomly shuffled the data as soon as it imported to the data loader, and randomly pick 70% of them to train our network, and left 30% of the data to cross validate our output. [Chen et al., 2017] used the same classifying techniques as their research in [Hossain and Gedeon, 2017], which includes k-nearest neighbor (KNN), support vector machine (SVM). They achieved a 95% for automatically detecting the observers' response, though only 60% of observer themselves can precisely tell verbally whether the anger is pretended or not, based on the video they are watching.

### 2.3 Hyper parameters

In this study, we limits the number of hidden units by "`k`" and the number of epochs by "`n$_$epochs`", through which we could avoid the network getting too large on the training set. The training set accounts for 60%, and the validation set, the test set each takes part 20% of the whole data set. In particular, we compares the original ConstrCasc with the GA involved ConstrCasc by setting the "`k`" to 15 and "`n$_$epochs`" to 2500, where both approaches seems to converge at this point.

## 3 Results

Table 3 demonstrates a comparison between different approaches, where the first two approaches come from [Chen et al., 2017], the third approach is from our previous work, and the last two approaches are the original

| Approache | Accuracy |
|---|---|
| Verbal | 60.0% |
| Chen et al. | 95.0% |
| ConstrCasc from previous work | 54.7% |
| ConstrCasc in this study | 74.2% |
| ConstrCasc with Evolutionary Algorithm | 65.1% |

**Table 1.** Accuracy from different approaches

ConstrCasc and ConstrCasc with evolutionary algorithm in this study.
Figure 3 and Figure 4 shows the trending of loss as the training goes. It shows that introducing evolutionary

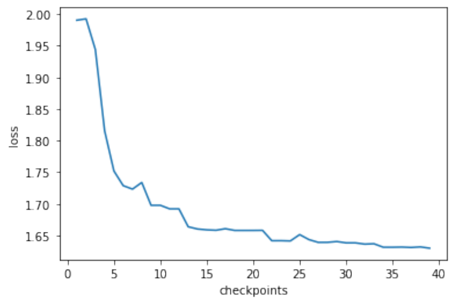


**Fig. 3.** ConstrCasc

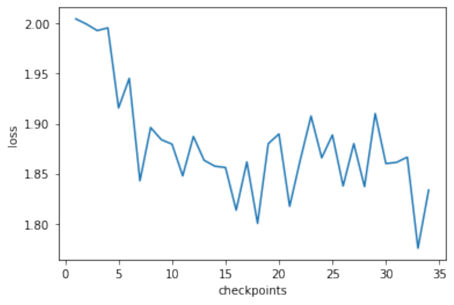


**Fig. 4.** ConstrCasc with evolutionary algorithm (CCEA)

algorithm actually makes the performance of ConstrCasc even worse. Although the loss of CCEA also decreases as the training goes, it does not always decrease every time a new hidden neurons is added. This is not normal since the idea behind introducing correlation here is to make those newly added hidden neurons as feature extractors, which would improve model's ability to capture features and improves accuracy. Figure compares difference between the network we firstly developed in figure 1 and the Constructive Cascade Neural Network with jump connections. It shows that the accuracy of original work would even decrease as the training goes. This might because when there are more training data, the network goes too deep with too many "Bad neurons", it makes the network more and more clumsy. However the accuracy for Constructive Cascade Neural Network seems to be much better, and the accuracy grows as the training goes.

### 3.1 Discussion

After introducing Video number as the seventh feature, the performance of the algorithm improves about 20%, finally it gets more accurate than human's subjective feedback from observers. However, the accuracy is still more than 20% worse than the original study on anger, the reason might be that we are still using the pre-processed data set for this study, in this way some of the more explicit associations may thus have been overlooked. More disappointingly, bringing evolutionary algorithm into ConstrCasc does not meet our expectations, in fact, it becomes a big drag on the original algorithm. Meanwhile figure 4 looks pretty much like the diagram that illustrates the loss of ConstrCasc from our last work. Based on previous experience, we assume the process of deploying the evolutionary algorithm brings too many "Bad neurons", which do not serves as features extractors but make the network more and more clumsy.

## 4 Conclusion

The general objective of this study is based on our previous work, where we applied a classification to determine whether the anger of someone in the video is faked or not, based on physiological signals collected from observers. It was basically a reproduce of [Chen et al., 2017]'s work, but with Constructive Cascade Neural Network. In this study, we improved our usage of the data set, and introduced evolutionary algorithm to ConstrCasc, it was a good attempt, but the result is barely satisfactory. Consequently, the reuse of "Video" in the data set brings an improvement of "19.5%", although it is still much lower than the original work, but finally gets higher than observers' verbal response, which makes it more valuable. Unfortunately, in our implementation, the evolutionary algorithm was not able to improve the ConstrCasc, but even holds back about 10%, after reading through the possible reasons written in [Forrest and Mitchell, 1994], we think the reason might lies in the setting of parameters, the "evolving" was not enough, thus makes those newly added neurons not able to represent the feature correctly.

### 4.1 Future Work

We are still optimistic about applying evolutionary algorithms to automatically topology deciding. In the next step, we would focusing on setting up the hyper parameters more carefully and reasonably, we may also try AutoML techniques as an validation to see if the evolutionary algorithm can actually improve ConstrCasc. We would also extend our usage on the data set, by analysing the raw data and apply it on our existing approaches.

## References

[Chen et al., 2017] Chen, L., Gedeon, T., Hossain, M. Z., and Caldwell, S. (2017). Are you really angry?: detecting emotion veracity as a proposed tool for interaction. *Proceedings of the 29th Australian Conference on Computer-Human Interaction.*

[Fahlman and Lebiere, 1990] Fahlman, S. E. and Lebiere, C. (1990). *The Cascade-Correlation Learning Architecture*, page 524–532. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

[Forrest and Mitchell, 1994] Forrest, S. and Mitchell, M. (1994). What makes a problem hard for a genetic algorithm? some anomalous results and their explanation. *Machine Learning*, 13.

[Hossain and Gedeon, 2017] Hossain, M. Z. and Gedeon, T. (2017). Classifying posed and real smiles from observers' peripheral physiology. In *Proceedings of the 11th EAI International Conference on Pervasive Computing Technologies for Healthcare*, PervasiveHealth '17, page 460–463, New York, NY, USA. Association for Computing Machinery.

[Keltner et al., ] Keltner, D., Oatley, K., and Jenkins, J. M. *Understanding emotions.*

[Khoo and Gedeon, 2009] Khoo, S. and Gedeon, T. (2009). *Generalisation Performance vs. Architecture Variations in Constructive Cascade Networks.* Springer Berlin Heidelberg, Berlin, Heidelberg.

[Péter, 2018] Péter, S. (2018). Exploratory factor analysis of wireline logs using a float-encoded genetic algorithm. *Mathematical Geosciences*, 50:317–335.