The Application of Constructive Cascade Neural Networks in Detecting Genuine/Posed Anger

Zhetao Zhong[1]

Australian National University, Acton ACT 2601, Australia
email: zhetao.zhong@anu.edu.au

Abstract. Neural Networks in one of the powerful tools in data science which is widely used in processing data. Among the variants of neural network structures and training methods, Constructive cascade algorithms are powerful ones in improving the training of feed-forward neural networks. By starting with the minimal networks and adding hidden neurons and connections during the learning process.

In this work, I adapted the constructive cascade algorithm to the linear neural network for predicting genuine or posed anger based on pure numeric data. Starting from no cascade level, one cascade level and two cascade levels are implemented. Also, the performance is compared to my initial work of using normal full-connected linear network to predict the genuine.

As shown in the result, the modified network using constructive cascade shows an increased performance. Comparing to the one without any cascade modification, the model's performance increases by 6-8% with first cascade layer and the increment is still around the previous value after adding the second cascade layer.

Keywords: Constructive Cascade · Neural Networks · Anger Recognition · Machine Classification.

1 Introduction

Facial expressions are important in social interactions. They are a central part of human communication [1], providing significant help in understanding each other as well as the conversation situation as it usually reflects the internal mental state of the displayer of the emotion [2]. Sometimes a huge misunderstanding may just begin from a wrongly-understood facial expression. In HCI area, understanding users' affects better often leads to better decisions.

Anger, which is a kind of normal emotions, is very to express and to be understood through facial expression. But is all the anger we find on someones face are real? Someone may just pretend to be angry and this kind of facial expression is called Posed Anger here. Theres already some work showing that even for the same emotion, genuine and fake one have different physiological signals which can be used by machine to distinguish them [4][5].

Constructive cascade enables networks to build during learning [6][7]. This work is aiming to figure out how this technique can improve or optimise the linear neural network in distinguishing the fake anger. The performance difference between constructive cascade modified neural networks and plain full connected network is compared and analysed in this work. The experiment is based on a dataset retrieved from different videos in which 20 data entries are created for a single video. The dataset is balanced with half genuine (real) expression of anger and half posed (fake) entries.

Result of this work can provide supports for designing the structure and setting up neural networks distinguishing and analysing facial expressions helpful in areas such as HCI user feedback [8], handling user affects [9].

2 Related Works

There has already been a number of work done on facial expression recognition and distinguishing real/fake expressions.

Md Zakir and Tom[3] introduced a high-accuracy system to distinguish real and fake smiles by sensing observers galvanic skin response (GSR, indicating electrical changes measured at the surface of human skin). Different combination of feature-selecting algorithms and network structures including SVM, KNN and NN were experimented. They succeeded in finishing up with the accuracy of 96.5% using simple NN network and selected features.

The work on detecting emotion veracity[2] provided a method to tell the real or fake anger by using emotion perceivers physiological signals. In the experiment, 22 participants were required to view two types of anger stimuli. During the viewing, their physiological signals including skin conductance, blood volume pulse, heart rate, eye gaze and pupil size were recorded. Several designed questions were asked after their viewing to test how they recognise the real/fake anger. In the end, it proved that machines can retrieve useful information from emotion perceiver's physiological signals, especially pupillary response, to achieve a much higher accuracy to differentiating genuine and acted anger.

In the work provided in [10], they proposed an approach to classify videos of real emotions from fake ones. The mechanism used CNN-Resnet and Rank SVM structure, along with regression trees to retrieve features and attributes. The result stated in their work also supported that the machine did better than human beings in most situations.

Two new algorithms called Local Feature Constructive Cascade (LoCC) and Symmetry Local Feature Constructive Cascade (SymLoCC) and their results were introduced in the work of [11]. Their work mainly focused on improving the performance of face recognition using Convolutional Neural Network structure. By adding cascade layers with fixed size and neuron numbers each time when the certain condition is triggered, as well as using symmetric parameters on more symmetrical images, the number of free parameters and weights are reduced significantly but still preserve the good generalisation properties.

3 Method

All the work is based on the dataset 'Anger'. For working on detecting genuine or posed, video numbers in the dataset is ignored.

To maximise the accuracy, improve the performance and increase the reliability of the result, several techniques are applied in this work.

Stated in [12], normalisation methods widely used in back propagation neural networks to enhance the reliability and accuracy. The normalisation is usually done to data which is in non-linear patterns and difficult to classify. In the dataset used in this work, all the attributes sit between 0 to 1, but only expect data from one attribute, data under other attributes are all around 0.2, with most data under one attribute is below 0.1. To balance the range, all data are processed using Min-Max Normalisation mentioned in [12]. Mean and Standard Deviation Normalisation is not applied here as the result shows that this can easily cause the model to be over-fitting even without adding cascade layer.

The activation function between all layers as well as the output is hyperbolic tangent function which was also used in [11]. As discussed in [13], compared to sigmoid, this function has similar curve but the output range is between -1 and 1. Providing higher accuracy, it was also shown that the hyperbolic tangent worked better than sigmoid as the activation function. Additionally, it is suggested in [11] that hyperbolic tangent usually helps when a quick convergence is expected while training the model.

To make the work more credible, Cross-validation [14][15][16] which is one of methods most widely used of evaluating predictive performances of a model [17] has been used as a part of the application. It basically split the dataset into several pieces. Among these pieces of data, part of them is used to fit into the model for training while the rest part is used to measure the performance. The prediction value is compared with the actual value in validation data to calculate the accuracy. The validation set is not involved in the training so that this shows how the model works on independent data. Because the number of data is not significantly large in this case, the result is coming from the average of running the program 10 times manually.

In choosing the optimiser for network learning, Resilient propagation (RPROP) [18], which also appeared in [11], is used in all kinds of network learning appeared in this work. The RPROP was an adaptive learning algorithm for multi-layer feed-forward network, designed to overcome pure gradient-descent's inherent disadvantages. The size of derivatives can not be a big unforeseeable influence factor to this algorithm. Instead of taking care of the error gradient's magnitude, it only cares sign of the gradient whose change is a key factor in controlling the learning process and only local behaviour is taken into account.

Similar to the one mentioned in [11], the training did not start with the network directly linking the input to the output. In this application, the starting point is a linear NN network, fully connected, with a single hidden layer. It is shown in Fig 1, the left one. The size of input layer is exactly the same as the number of attributes in the data while the hidden layer has a fixed size of 15. Like many other binary classification, the output size is 2 and keeps unchanged during

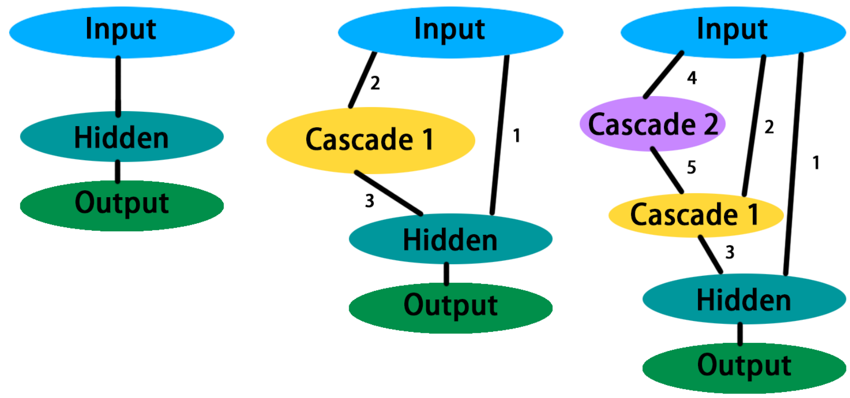

Fig. 1. The construction process of the cascade NN network

the construction. The hidden layer is not removed or modified while adding new cascade layers and all adding process happens between the input layer and the initial hidden layer.

The adding process is triggered by the number of epochs already run with a threshold of 200. That is, regardless of the loss, this mechanism will add a cascade layer after running 200 epochs on current network. The training stops when the last structure (the right one in Fig 1) is achieved and 200 epochs have been run on it.

All the layers and the links between them are labelled using numbers and colours, both shown in Fig 1. Unchanged number or colour means that link or layer should keep the same as the previous model. During the addition of cascade layers, all the properties of current model are saved. When a new cascade layer is put into the model to form an updated model, the data saved above will be loaded into the new model as its starting point. For instance, transforming from the middle one to the right one in Fig 1, only the layer 2 (in purple) and link 4 & 5 are brand new to be trained from the very beginning. Others are inherited from the middle one. This structure, whose similar applications can be found in ResNet [19] and DenseNet [20], provides a shortcut for the training in my work.

As all the cascade layers have a size of 15, when 2 links are linked to the same entry, the merge is necessary. The solution in this work is adding 2 same size outputs together and use the average value as the result.

Finally, the model to compare with is a fully-connected linear NN model with 2 hidden layers. The size in order from the input to the output is 6-15-7-2. All other techniques such as activation, loss and optimiser functions are kept the same as the cascade model.

4 Result and Discussion

Result of this application is given also based on the same dataset as the training one. The accuracy was measured using cross validation as mentioned above. I took the average of 10 running outcomes as the final result.

Additional layers increase the size and performance of the network at the same time. While the training time of the last model is obviously longer, some notable performance increase can be found as the result of cascade layers. The accuracy of my model sits above 80% which is an quite acceptable score. But compared to work in [2], there's still a big gap around 10%. Fig 2 shows how accuracy grows as it learns. The blue, orange and green line stands for the initial, one-cascade-layer and two-cascade-layer models respectively. The floats on x-axis is the epochs already run, 1.0 equals to 50 epochs. Table 1 compares the time used in training each model which can be understood as the computing resources needed and their performance.

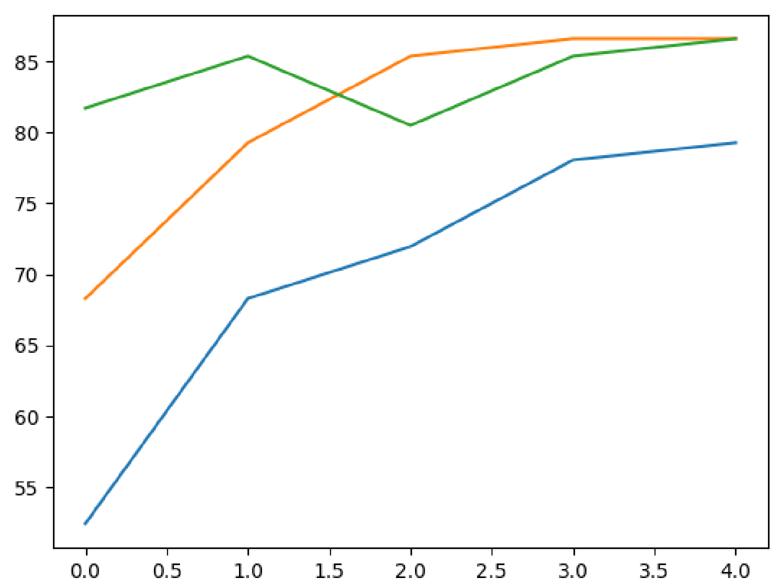

Fig. 2. Result of 3 network structures using cascade layers.

From Fig. 2, it is easy to find that the orange and green line end up almost at the same point much higher than the end point of blue line meaning that the one-cascade and two-cascade models deliver nearly the same performance after training and they over perform the initial model to a great extent in this case. Observing the trend of growth, both orange and blue lines show the normal pattern of starting low and converging to a certain level. But the green one is kind of floating around, not showing notable increment in the accuracy. We could not even find a stable pattern of convergence and there's considered a potential risk of over-fitting if further training is applied to this model.

Then from Table 1, comparing the cost of these models, the simplest no-cascade model used only 0.205s, just bit over one-fourth of time used by the most consuming two-cascade model which took 0.740s to train, but the performance is not satisfying. Between them is the one-cascade model and full connected NN without cascade technique. Their time consumptions are around 2 and 3 times of the no-cascade model respectively but considering the improvement from no-cascade to one-cascade, it is worth spending this amount of more time for the higher accuracy.

Table 1. Comparison of the time cost and performance of different models

Model Name	Accuracy	Time Cost
No Cascade	79.27%	0.205s
One Cascade	86.59%	0.441s
Two Cascade	87.08%	0.740s
Full Connected 6-15-7-2	78.21%	0.663s

It is true that the most complicated model presented here is the most capable one to do the classification. However, it should be taken into account that there's a trade off between performance and model simplicity. The best one consumes over 1.5 times than the second best one and only provides around 0.5% better in the accuracy. Whether trade off like this worthwhile and should we add more cascade layers to the model is a question to discuss, hard to find the answer. Size of the dataset used here is limited but there can be much larger datasets in other cases. As the size scales up, the time consumption difference can go up to hours even days. It is suggested in this case that adding a second cascade layer is not beneficial.

5 Conclusion

I introduced the application of Constructive Cascade Neural Networks in detecting genuine or posed anger. The cascade layers are constrained with a fixed size of 15 and added to the network based on the number of epochs run on the previous network. I have examined and compared the performance of networks with different number of cascade layers added into them as well as an example of non-cascade NN network. My result shows that the Constructive Cascade technique can improve the performance and runs better than the normal NN to compare in this case. But while thinking about adding layer, we should always take care of the overhead and possibility of over-fitting. More layers may bring us better accuracy but not always beneficial.

The future work will be improving this model using more data. Current result is good but there should still be some possibilities to improve. Additionally, it is worthwhile to figure out how this works on data much larger than current dataset.

References

1. Mri, R.M. (2016), Cortical control of facial expression. *J. Comp. Neurol.*, 524: 1578-1585. https://doi.org/10.1002/cne.23908
2. Chen, L., Gedeon, T., Hossain, M. Z., Caldwell, S. (2017, November). Are you really angry?: detecting emotion veracity as a proposed tool for interaction. *In Proceedings of the 29th Australian Conference on Computer-Human Interaction* (pp. 412-416). ACM.
3. Md Zakir Hossain and Tom Gedeon. Observers Galvanic Skin Response for Discriminating Real from Fake Smiles. *Australian Conference on Information Systems (ACIS '16)*, Wollongong, 1-8. Retrieved from: http://ro.uow.edu.au/acis2016/papers/1/33
4. Masood Mehmood Khan, Robert D. Ward, and Michael Ingleby. 2009. Classifying pretended and evoked facial expressions of positive and negative affective states using infrared measurement of skin temperature. *ACM Trans. Appl. Percept.* 6, 1, Article 6 (February 2009), 22 pages. https://doi.org/https://dx.doi.org/10.1145/1462055.1462061
5. Chi Jung Kim, and Min-Hyuk Chang. 2015. Actual Emotion and False Emotion Classification by Physiological Signal. In *Signal Processing, Image Processing and Pattern Recognition (SIP '15)*, 21-24. IEEE. https://doi.org/http://doi.ieee.org/10.1109/SIP.2015.17
6. Tin-Yau Kwok and Dit-Yan Yeung, "Constructive algorithms for structure learning in feedforward neural networks for regression problems," in *IEEE Transactions on Neural Networks*, vol. 8, no. 3, pp. 630-645, May 1997.
7. N. K. Treadgold and T. D. Gedeon, "Exploring constructive cascade networks," *IEEE Transactions on Neural Networks* , vol. 10, no. 6, pp. 1335-1350, Nov. 1999.
8. Zhihong Zeng, Jilin Tu, Ming Liu, Tong Zhang, Nicholas Rizzolo, Zhenqiu Zhang, Thomas S. Huang, Dan Roth, and Stephen Levinson. 2004. Bimodal HCI-related affect recognition. *In Proceedings of the 6th international conference on Multimodal interfaces (ICMI 04).* Association for Computing Machinery, New York, NY, USA, 137143. https://doi.org/https://doi-org.virtual.anu.edu.au/10.1145/1027933.1027958
9. Eva Hudlicka. 2003. To feel or not to feel: the role of affect in human-computer interaction. *Int. J. Hum.-Comput. Stud.* 59, 12 (July 2003), 132. https://doi.org/https://doi.org/10.1016/S1071-5819(03)00047-8
10. F. Saxen, and P. Werner and A. Al-Hamadi, "Real vs. Fake Emotion Challenge: Learning to Rank Authenticity from Facial Activity Descriptors," *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*, Venice, 2017, pp. 3073-3078.

11. Khoo, Suisin and Gedeon, Tom. (2008). Generalisation Performance vs. Architecture Variations in *Constructive Cascade Networks.* 236-243. 10.1007/978-3-642-03040-6_29.
12. Jayalakshmi, T. and Santhakumaran, A. (2011). Statistical normalization and back propagation for classification. *International Journal of Computer Theory and Engineering*, 3(1), 1793-8201.
13. Karlik, B.,and Olgac, A. V. (2011). Performance analysis of various activation functions in generalized MLP architectures of neural networks. *International Journal of Artificial Intelligence and Expert Systems*, 1(4), 111-122.
14. Stone, M. (1974). Crossvalidatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society: Series B (Methodological)*, 36(2), 111-133.
15. Geisser, S. (1975). The predictive sample reuse method with applications. *Journal of the American statistical Association*, 70(350), 320-328.
16. Allen, D. M. (1974). The relationship between variable selection and data agumentation and a method for prediction. *Technometrics* , 16(1), 125-127.
17. Zhang, Yongli and Yang, Yuhong. (2015). Cross-validation for selecting a model selection procedure. *Journal of Econometrics.* 187. 10.1016/j.jeconom.2015.02.006.
18. Riedmiller, M., Braun, H.: A Direct Adaptive Method for Faster Backpropagation Learning: The RPROP Algorithm. In: *IEEE Int. Conf. on Neural Networks*, pp. 586591 (1993)
19. Keeler, J. D., Hartman, E. J., Liano, K., and Ferguson, R. B. (1994). *U.S. Patent No. 5,353,207.* Washington, DC: U.S. Patent and Trademark Office.
20. Iandola, F., Moskewicz, M., Karayev, S., Girshick, R., Darrell, T., and Keutzer, K. (2014). Densenet: Implementing efficient convnet descriptor pyramids. *arXiv preprint arXiv:1404.1869.*