Best match between SARS classification and Constructive Correlation Neuron Networks

Jieli Zheng
U6579712@anu.edu.au

Research School of Computer Science
Australian National University ACT, Australia

Abstract. The Constructive Cascade Networks [2] are flexible and expressive deep learning algorithms especially on real-world complex tasks. It is what we expect to build a medical diagnosis classifier with appropriate network complexity and good accuracy. In this article we intend to use Cascor [1] algorithm to classify several real-world diseases by few physiological indicators and analyse the performance of Cascor on this classification problem in many aspects. We will provide some experimental results and analyse them. Finally, we will talk about the advantages and drawbacks on CasCor to explore future works.

Keywords: Computer Science, Artificial Intelligence, Classification, Deep neural network, Cascade networks, Constructive networks, Cascor, COVID detection.

1 Introduction

The deep neural network has got unparalleled success in Artificial Intelligence due to its generalization ability. Pre-defined network architecture and hyper-parameters are necessary before training classic neural networks algorithms. What's more, the performance of neural networks like CNN are usually sensitive to these settings [3]. However, it's difficult for human researchers to analyse the complexity of a real-world task and balance between expressiveness and efficiency of the network.

SARS-CoV-2, also known as COVID-19, has become a global pandemic disease from 2020 spring and also a landmark event in history. In this article we are intended to build a classification deep learning model on the first SARS subspecies: SARS-CoV, which has more severe symptoms, higher fatality rate and lower transmission rate [4]. Although Nucleic acid amplification tests, or NAATs [5] is usually the fastest way to detect coronavirus diseases. It takes several months for medical scientists to develop it for each specific coronavirus. Also, Medical observation and examination are much slower when we are facing outbreak of infection caused by new unknown coronavirus. SARS-CoV-2 shares mild symptoms with SARS-CoV like fever and cough [4], therefore, we intended to train the classifier by few physiological indicators which are easy to measure (e.g. sequential body temperature) in order to guide the early detection of COVID-19 for human doctors.

The algorithm we choose here is one of the subclasses in Constructive Cascade Networks [2], Cascade Correlation Networks, or CasCor which is also described in [8]. We will firstly show that the constructive topology of CasCor Networks has the ability to generate feature extractors and freeze them. Secondly, we will show the layers to be frozen in the topology graph and the main process of the CasCor algorithm. Then, we will reduce the information, or dimensions in training set to show that CasCor can extract deep features from only one attribute: sequential body temperature. Finally, we will explain these frozen feature extractors are high expressiveness layers by a much lower time cost, loss and weight size than traditional fully connected neuron networks.

2 Cascade Correlation Method

2.1 Neural Network Topology

Constructive neural networks are constructive because they will change their network architecture due to some conditions. Here firstly Cascade Correlation networks start as a simple fully connected network with only input and output layer in Fig.1.1. The weight between input neuron and output neuron is live and accepts backpropagation gradient change. It's obvious that such a vanilla fully connected one-layer neuron network is not expressive enough to extract the features in training data.

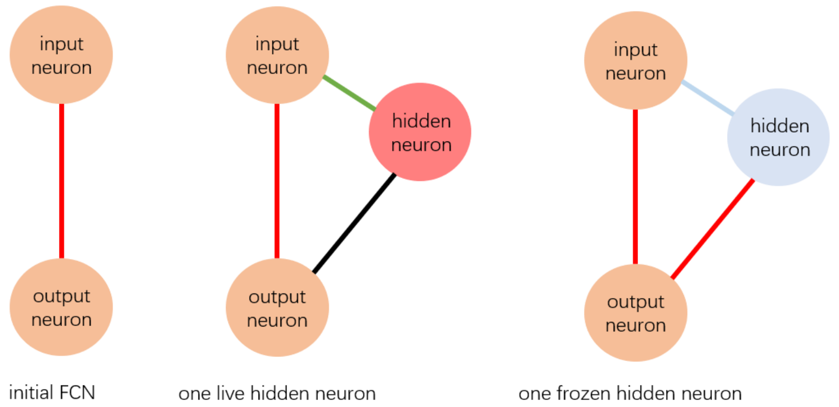

Fig.1.The process to add a hidden neuron / layer into the network

Once the loss of the training data does not decrease, the CasCor will introduce a new candidate neuron from the candidate pool. The optimizing strategy we use here is different from ordinary optimizing steps. Red line in Fig1. indicates a live weight during main optimizing process, and green line indicates that the weight is optimized by correlation strategy in sub-process which is exactly the reason for naming this network as CasCor. The black line indicates that this weight is not used in sub-process, and the light blue line indicates that this weight will forever be frozen and maintain its value till the algorithm terminates. From Fig1.2., we find that the green weight between input and latest added hidden neuron is optimized by correlation loss between network classification result and the difference of the output of this hidden neuron and ground truth label. Once this correlation optimizing process is done, the input to hidden weight from latest added neuron will be frozen in the entire training process. The effect of freezing this weight is to extract unsupervised and fixed features and remains the contribution of output (hidden to output weight) trainable. The reason that this fixed extractor is beneficial for the performance of the CasCor networks is that the optimizer can maximize the correlation between the expressiveness of the latest hidden neuron and frozen weight. Therefore, we can assert that the frozen weight is positively related to the final loss decreasing and the algorithm will go back to main optimizing process. As is shown in Fig1.3., the red line is still trainable in CasCor.

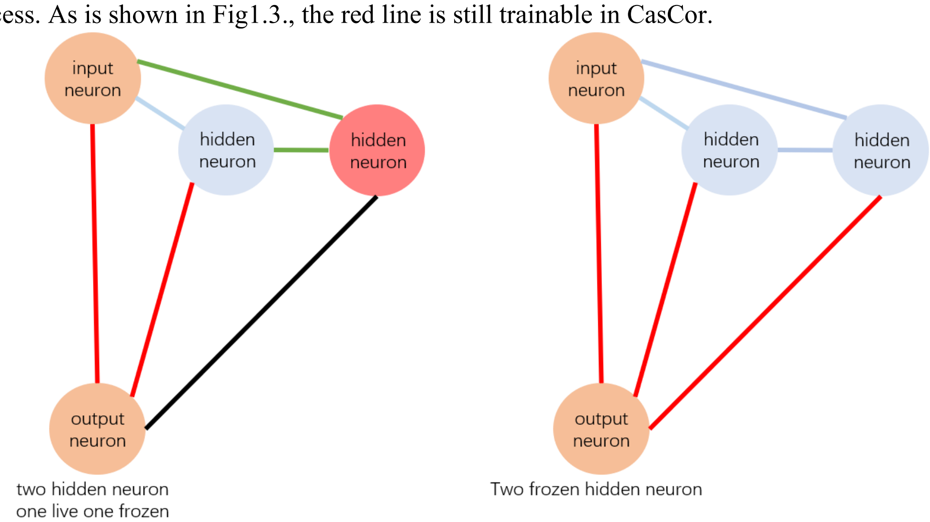

Fig2. The process to add another hidden neuron

As is shown above in Fig2., The process is nearly the same in addition to train hidden to hidden weights in sub-process correlation training. It means that each hidden neuron should connect to all previous frozen hidden neurons and freeze connecting weights after correlation training. The reason is that the model may generate a poor feature extractor (e.g. early added hidden neurons may not learn deep and determining features), and latest hidden neurons can make use of the hidden to hidden weight to correct them.

2.2 Pseudo Code of the Method

Here is the main process and sub-process of the training pseudo code.

```
Load data from files
Generate dataloader from sequential data
network = CasCor()
For epoch in #_of_epochs:
    For batch_index, data, labels in dataloader:
        forward results = forward(data)
        Optimize on forward results and labels
        Backpropagation
    If loss has no improvement and conditions*:
        add new neuron
        optimize the correlation on new hidden neuron
        freeze the weight of the new hidden neuron
END
```

conditions*: Since the result will become unstable if we instantly calculate the test accuracy just after adding new hidden neurons which are randomly initialized, we may restrict the method not to add new hidden neuron under some conditions:

1. has already added neuron several epochs before
2. is in the last several epochs
3. has already reached the maximum threshold of adding hidden neurons

2.3 Dataset

The dataset we use for CasCor Algorithm is SARS-CoV dataset from B. Sumudu U. Mendis1, Tamás D. Gedeon1, László T. Kóczy [9], which is a tiny dataset containing physiological indicator data. One datapoint in this dataset is a 23 tuple:

```
(temp@8am-slight,
temp@8am-med,
temp@8am-high,
temp@12pm-slight,
temp@12pm -med,
temp@12pm -high,
temp@8pm-slight,
temp@8pm -med,
temp@8pm -high,
BP-Systolic-slight,
BP-Systolic-med,
BP-Systolic -high,
BP-Diastolic-slight,
BP-Diastolic-med,
BP-Diastolic-high,
Nausea-slight,
Nausea- med,
Nausea- high,
Abdominal-Pain-No,
Abdominal-Pain-Yes
)
```

In SARS-CoV dataset, there are 4000 datapoints for 4 labels: SARS patients, Normal people, High BP patients and Pneumonia patients. Since some of the attributes are so iconic that if some conditions are satisfied, we can directly assert the label of the datapoint, e.g. Only SARS patients have abdominal pain, high body temperature and nausea at the same time. Therefore, this classifier problem will become a linear problem and no need to use neural networks. Hence, we slice only the temperatures in data in order to extract deep features from small samples. Also, body temperature is significant medical data which are easy to detect and highly related to SARS. [10] As a result, the dimension of the datapoint will be 12x1 which are sequential body temperatures. We will also use one-hot encoding for labels to do correlation optimization in sub-process. We split 7.5% of the entire dataset as test set, which is only 308 datapoints in it. Therefore, this training and test set is a small sample training dataset.

3 Experimental Results and Analysis

We choose vanilla one-layer FCN as our baseline because initially the CasCor Network has the same network structure as FCN. Since the dataset is a small sample dataset with 4000 datapoints in total, the performance of the model is not very consistent and highly depends on the shuffle batch of the dataloader. Larger dataset may have stable performance. The dimension of the input datapoints is 12 as mentioned before, and the labels use one-hot encoding. We set the number of epochs to 90 for both networks, and main optimizer is SGD with learning rate = 0.001 and momentum=0.9. The main loss function is the cross-entropy loss function which is usually for classic classification problems loss function for correlation process in CasCor, I use a correlation loss function, which is the negative value of ordinary correlation. The adding neuron restrictions we use here are: 1) not add neuron for the last 20% epochs. 2) not add neuron for the first epoch. 3) not add neuron if maximum of 10 neurons is reached. 4) the interval between adding next neuron is 5.

We are intended to use the accuracy on test dataset to evaluate the performance. The accuracy and loss result of the FCN and CasCor are as follows:

Name	Final Training Loss	Final Test Accuracy
CASCOR	0.472	74.351%
VANILLA FCN	0.756	53.846%

It is obvious that CasCor has both lower training loss and better test accuracy than Vanilla FCN. The main reason that CasCor is more powerful than FCN is that frozen hidden neurons are high label-related feature extractors and can maintain their extracted features by freezing themselves after each correlation training process.

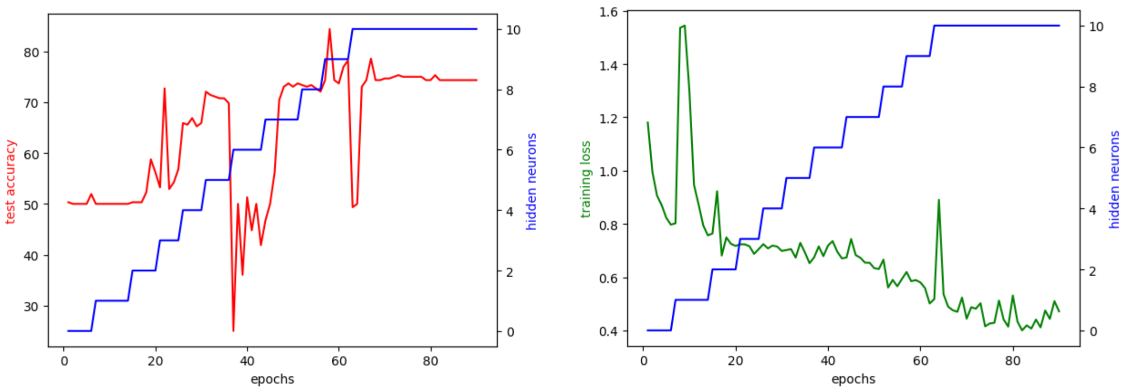

Fig3. Experimental Result of CasCor

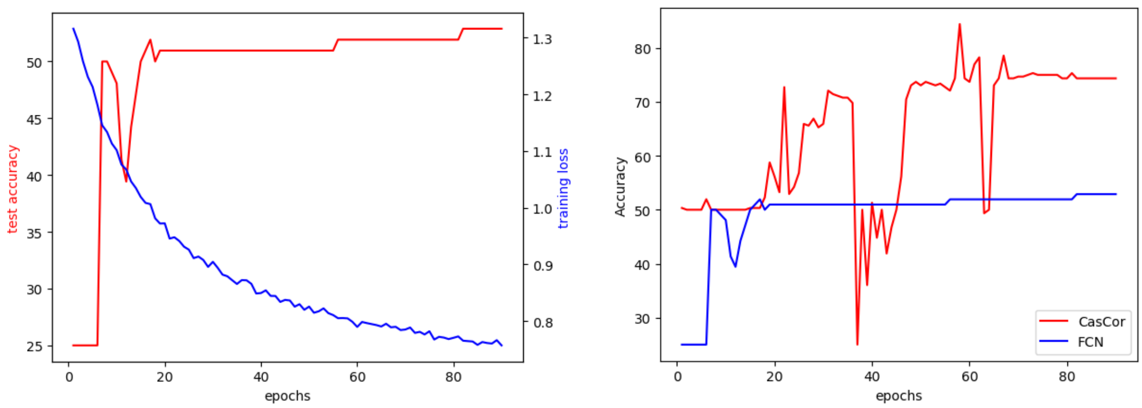

Fig4.Baseline (FCN) and Comparison between FCN and CasCor

The baseline model is a vanilla fully connected neural network and the loss and test accuracy curve of the baseline model is shown in Fig4.1., it has consistent but fairly bad final performance on SARS classification problem because there are four labels under this problem and the accuracy of randomly selection is 25% compared to 53%. We noticed that vanilla FCN also faces huge test accuracy drop. We think the reason is the momentum of SGD optimizer is a bit large.

From Fig3.1 and Fig3.2., we can observe that the model performs an accuracy drop when adding the sixth and ninth hidden neuron, and the final accuracy in epoch 90 is not the highest accuracy among all epochs. Firstly, the reason for this performance drop is that the new hidden neuron is randomly initialized, then the model trains its weight by correlation of outputs, which is even negative related to the classification accuracy. The model will need several optimizing steps to train the hidden to output weights to apply the frozen feature extractor neuron on the input data correctly. Secondly, the final accuracy is not the highest and training loss is not consistent because CasCor may not extract enough key features from such a small dataset. Since correlation optimizing uses different targets in loss functions, this sub process result is inconsistent for the main optimizing loss. Also, the increase of the cross-entropy loss is because the new random initialized hidden to output weights are not optimal in the following several epochs.

From Fig4.2. compared to the baseline, we find that despite CasCor will face test accuracy decreasing just after adding a new correlated neuron, the accuracy will quickly increase because the frozen feature extractors are highly correlated to final prediction, then small gradient descent in the hidden to output weights can have 30% test accuracy increase in few epochs.

As a result, finally the CasCor will have better performance than vanilla FCN in dataset-SARS-CoV-1.

4 Conclusion and Future Work

We have introduced Cascade Correlation Network, which is one of the constructive cascade neural networks [1]. In this article we have a quick review on the topology of CasCor and try to apply CasCor on SARS dataset in order to train a patient classification algorithm. The result of CasCor has an edge over vanilla neural networks by around 20% test accuracy increase in SARS dataset. Therefore, this model can be efficient for people to check the self-risk of SARS or other diseases since it only requires body temperature data. Also, it is a light neuron network that does not require much space and computational power [11] (like GPT-3) to boost the prediction performance. The main point of CasCor is it can automatically match the network complexity to the problem true complexity by adding highly correlated feature extractors, or frozen hidden neurons. Because the hidden neurons are highly correlated to the prediction outputs and they were frozen once they were introduced into the network, the CasCor can easily extract features via optimizing the hidden to output weights. CasCor can also save training time since the network architecture is one-layer fully connected network at the beginning and will grow depending on the problem complexity. The CasCor algorithm also faces problems such as accuracy drops due to bad feature extractors, strange shape of the outputs for neurons to saturate and too many neurons if the features are deep, high dimensional and too hard to extract early [1].

In the future I am going to find a heuristic strategy of adding neurons firstly since when to add a new hidden neuron highly affects the final performance of the model. If the neuron is added just before the algorithm terminates, or we add the hidden neuron too often, the model will have poor performance. Also, the initialization of the CasCor should be considered because the random initialized weight is hard to persist small loss penalty. And it's interesting if we transfer the initial FCN network to RNN or Transformer to handle higher dimension data since vanilla neuron network can hard extract time series, language and image data features like long sequential body temperature or face emotion detection. Additionally, we may focus on retrieve more SARS-temperature data to have better experimental results. If we are still in lack of data, we may explore the few-shot learning [12] and transfer learning [13]. And it's another improvement to be made that we can try different optimizing strategy like Adam [14] or use better loss function to describe the correlation when adding neurons. Another method is to bring the frozen neurons back to live, which is exactly what CasPer [6] does.

In conclusion, CasCor is a dynamic generative neuron network strategy which is timesaving and highly matches the complexity of the real-world problems. We also show that CasCor can solve SARS classification problem well in few-shot.

5 Acknowledgements

It is such a hard time for everyone all over the world in COVID-19 pandemic. I'm really appreciated to the Prof. Gedeon and other researchers for their patient instructions in piazza. Academic communication is still unobstructed there. Also, I'd like to thank my parents, friends to support me when I felt frustrated. Finally, I appreciate Arknights for their amazing upcoming 2nd anniversary event "Under Tides" which brings me passion during this work.

References

1. Fahlman, S.E., and Lebiere, C. (1990) The cascade-correlation learning architecture. In *Advances in Neural Information Processing II*, Touretzky, Ed. SanMateo, CA: Morgan Kauffman, 1990, pp. 524-532.
2. Khoo S., Gedeon T. (2009) 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. https://doi.org/10.1007/978-3-642-03040-6_29
3. Zhang, Y., & Wallace, B. (2015). A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. *arXiv preprint arXiv:1510.03820*.
4. Eskild, P., Marion, K., Unyeong, G., Davidson, H, H., Nicola, P., Francesco, C., Merete, S., Sulien, Al, K., Lone, Simonsen. (2020). Comparing SARS-CoV-2 with SARS-CoV and influenza pandemics. *The Lancet Infectious Diseases*, 20, 238-244. Retrieved April 24, 2021, from DOI: https://doi.org/10.1016/S1473-3099(20)30484-9
5. Nucleic Acid Amplification Tests (NAATs). (n.d.). Retrieved April 24, 2021, from https://www.cdc.gov/coronavirus/2019-ncov/lab/naats.html
6. Treadgold, N. K., & Gedeon, T. D. (1997, June). A cascade network algorithm employing progressive RPROP. In *International Work-Conference on Artificial Neural Networks* (pp. 733-742). Springer, Berlin, Heidelberg.
7. Treadgold, N. K., & Gedeon, T. D. (1998, May). Exploring architecture variations in constructive cascade networks. In *Neural Networks Proceedings, 1998. IEEE World Congress on Computational Intelligence. The 1998 IEEE International Joint Conference on* (Vol. 1, pp. 343-348). IEEE.
8. Treadgold, N. K., & Gedeon, T. D. (1999). Exploring constructive cascade networks. *IEEE Transactions on Neural Networks*, *10*(6), 1335-1350.

9. Mendis, B. S., Gedeon, T. D., & Koczy, L. T. (2005). Investigation of aggregation in fuzzy signatures, in Proceedings, 3rd International Conference on Computational Intelligence, Robotics and Autonomous Systems, Singapore. http://users.cecs.anu.edu.au/~Tom.Gedeon/pdfs/Investigation%20of%20Aggregation%20in%20Fuzzy%20Signatures.pdf
10. Huang, C., Wang, Y., Li, X., Ren, L., Zhao, J., Hu, Y., Zhang, L., Fan, G., Xu, J., Gu, X., Cheng, Z., Yu, T., Xia, J., Wei, Y., Wu, W., Xie, X., Yin, W., Li, H., Liu, M., . . . Cao, B. (2020). Clinical features of patients infected with 2019 novel coronavirus in wuhan, china. The Lancet (British Edition), 395(10223), 497-506. https://doi.org/10.1016/S0140-6736(20)30183-5
11. Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., ... & Amodei, D. (2020). Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.
12. Rahman, S., Khan, S., & Porikli, F. (2018). A unified approach for conventional zero-shot, generalized zero-shot, and few-shot learning. *IEEE Transactions on Image Processing, 27*(11), 5652-5667. https://doi.org/10.1109/TIP.2018.2861573
13. Gao, Y., & Mosalam, K. M. (2018). Deep transfer learning for Image-Based structural damage recognition. *Computer-Aided Civil and Infrastructure Engineering, 33*(9), 748-768. https://doi.org/10.1111/mice.12363
14. Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. December 2014