Variational CasPer Network is better in the SARS classification

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 I expect to build a medical diagnosis classifier with appropriate network complexity and good accuracy. In this article I intend to use Variational Casper [1] algorithm to classify several real-world diseases by few physiological indicators and analyse the performance of Var-Casper on this classification problem in many aspects. I will give out some experimental results and analyse the improvements compared to CasCor and other advantages and drawbacks of Var-Casper. Finally, I will talk about the possible improvements of the experimental problems to explore future works.

Keywords: Computer Science, Artificial Intelligence, Classification, Deep neural network, Cascade networks, Constructive networks, Casper, Var-Casper, 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 I am 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 I am facing outbreak of infection caused by new unknown coronavirus. SARS-CoV-2 shares mild symptoms with SARS-CoV like fever and cough [4], therefore, I 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 I choose here is a modified version of Constructive Cascade Networks [2], Variational Cascade Network Employing Progressive RPROP, or Var-Casper which is inspired by [6] and [15]. I will firstly show the constructive topology of Casper Networks has the ability to generatively add feature extractors. Secondly, I will show that the variational encoder block can help compress the sparse data dimension. Finally, I will discuss on the advantages and drawbacks of the Var-Casper compared to CasCor in my previous work, and the possible improvements of Var-Casper

2 Methodology

2.1 Casper Network Topology

Constructive neural networks are constructive because they will change the architecture of themselves when loss is no longer improved. Here firstly Casper networks start as a simple fully connected network with only input and output layer in Fig.1.1. It's obvious that such a vanilla fully connected one-layer neuron network with VAE encoding is not expressive enough to extract the features in training data. Once the loss is non-decreasing, the Casper will add a new neuron into the network. Compared to Cascor, Casper will not freeze previous additional neurons and set difference learning rate to each weight. In Fig.1.2, I can see the difference of each learning rate: L1, L2 and L3. Generally, the value of L1, L2 and L3 are 0.2, 0.005 and 0.001 referring to the technique paper. Learning rate of weights connected to the latest neuron will be set to L1 since the latest neuron is expected to be the feature extractor, and a larger learning rate can speed up the feature learning process. Similarly, the new extracted feature output should reduce the loss without too much interference from the previous weights [1]. Hence it should be set to L2 which is slightly larger than the L3. Since no neurons are frozen, the previous neurons can still be modified if it's necessary and beneficial. Then the model can both obtain the benefit of the weight freezing and the correlation techniques of Cascor, while avoiding early poor hidden neurons due to weight freezing and the saturation problems due to correlation measure. [15]

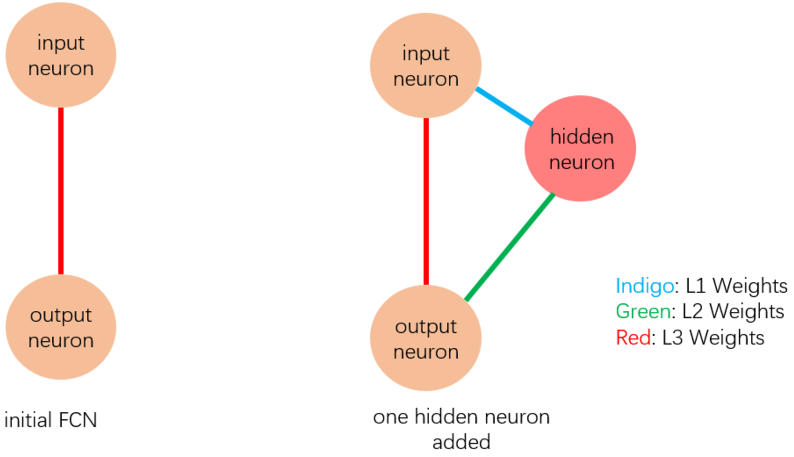


Fig.1.The process to add a hidden neuron / layer into the network

As is shown above in Fig2., it is almost the same when the model add another neuron into the network. Input neuron to 2nd hidden neuron and 1st hidden neuron to the 2nd hidden neuron use the largest weight to significantly change their value to learn features. L2 learning rate is used for the weight between 2nd hidden neuron to the output neuron, which is the output of the latest neuron and needs to give out a more flexible output than previous outputs. In general, the weights connect between input, the previous hidden neurons and the latest hidden neuron share the largest L1 learning rate, and L2 learning rate is used for the weight connects between the latest hidden neurons to the output, which is larger than L3. Therefore, the rest weights perform L3 weight which is the smallest and difficult to change its state.

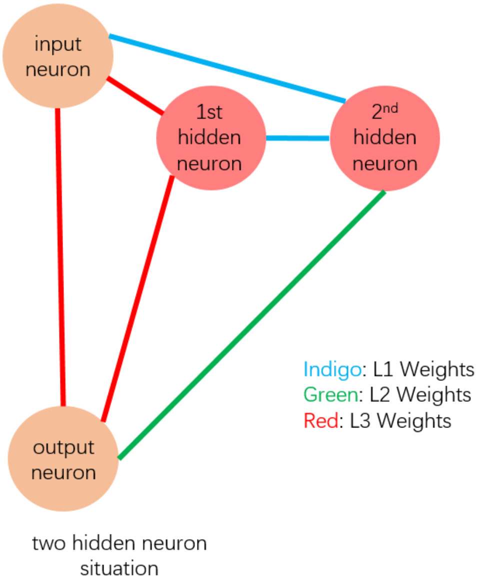


Fig2. The process to add another hidden neuron

2.2 Variational AutoEncoder Topology

AutoEncoder is usually used to find efficient data encodings in an unsupervised way. Variational AutoEncoder, or in short VAE, are a subclass of AutoEncoder [15]. VAE provides a probabilistic manner for describing an observation in latent space. Thus, rather than encode the data to a single vector to compress information, I try to make use of the probability distribution to encode the data with uncertainty. Here normal distribution is assumed for the following SARS-CoV Dataset. This is quite beneficial if the raw data has large variance and large dimension. A standard VAE should have the architecture in Fig.3. The data is compressed to its latent vector representation by variational encoder. The major difference between variational encoder and standard encoder is reparametrizing data to find the best mean and variance of latent distributions instead of finding latent representations blindly. Hence the latent vector is sampled from the latent calculated distributions in encoder. Thus, the decoder can sample the latent vector from the latent distributions to reconstruct the original data. Since I am using VAE to preprocess the SARS-CoV data instead of reconstruction, the only architecture to be used is the encoder of the VAE. The architecture of the variational preprocessing is shown in Fig.4. clearly. Usually AutoEncoder requires deeper architectures to extract the mean and variance, and since for task mentioned in this article, I expect to give a consistent accuracy curve during Casper training, then I will use 2 layers to obtain a balance between efficiency and good encodings. Compared to directly feed the data into Cascor in I previous work [17], here I use variational encoder in Fig.4 to reduce the dimension of the data in a probabilistic manner. After preprocessing the model will retrieve the 4-dimension data for Casper to classify.

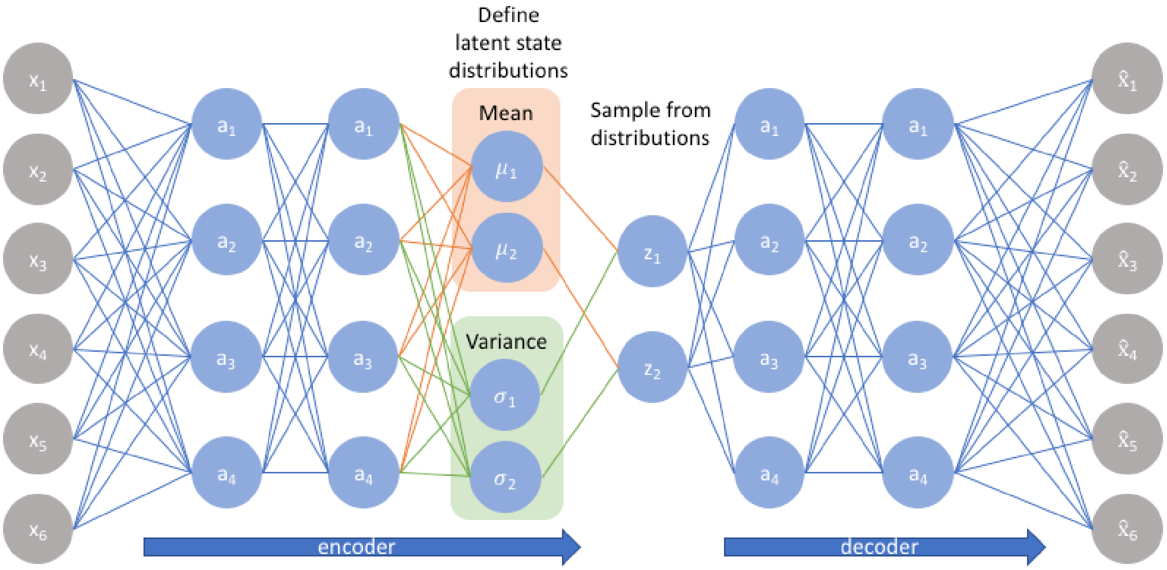


Fig.3. A standard VAE architecture from[16]

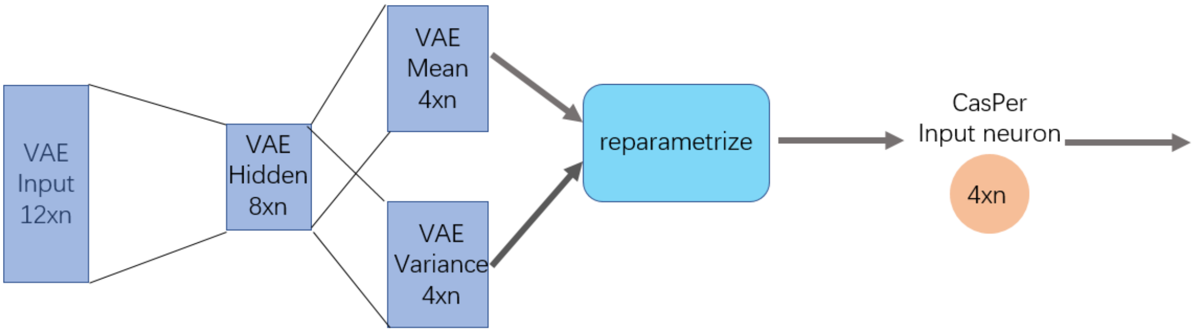


Fig.4 The architecture of Variational Encoder for Casper

2.3 Dataset

The dataset I use for Var-Casper 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 here is a 23-dimension data, and the range of each element has already been normalized. For example, temperature at 8am has three attributes: slight, medium, high, and they are normalized by default. Also, there isn't any noise datapoints in the dataset. Hence, I can just feed the data into I model without data preprocessing.

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, I 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. Also, some physiologic indexes like blood pressure are hard for people to track in daily life. Hence, I try to choose only body temperature by time as the input data. Also, body temperature is significant medical data which are easy to track and highly related to SARS. [10] As a result, the dimension of the input data will be 12x1 which are sequential body temperatures. I will show how the temperature is encoded by a Variational AutoEncoder and classified by Casper. I 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.

(temp@8am-slight, range from [0,1]
temp@8am-med, range from [0,1]
temp@8am-high, range from [0,1]
above three are normalized to 1.

temp@12pm-slight, range from [0,1]
temp@12pm -med, range from [0,1]
temp@12pm -high, range from [0,1]
above three are normalized to 1.

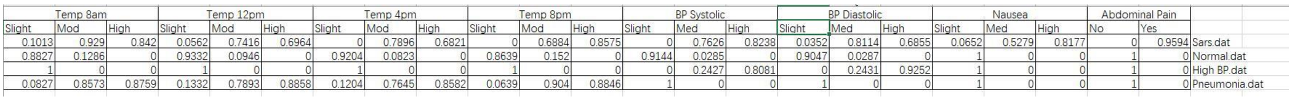
temp@8pm-slight, range from [0,1]
temp@8pm -med, range from [0,1]
temp@8pm -high, range from [0,1]
above three are normalized to 1.

BP-Systolic-slight, range from [0,1]
BP-Systolic-med, range from [0,1]
BP-Systolic -high, range from [0,1]
above three are normalized to 1.

BP-Diastolic-slight, range from [0,1]
BP-Diastolic-med, range from [0,1]
BP-Diastolic-high, range from [0,1]
above three are normalized to 1.

Nausea-slight, range from [0,1]
Nausea- med, range from [0,1]
Nausea- high, range from [0,1]
above three are normalized to 1.
Abdominal-Pain-No, range from [0,1]
Abdominal-Pain-Yes range from [0,1]
above two are normalized to 1.
)

From Chart.1., it's observed that the data space is sparse. For each category (e.g. temperature at 8am), usually only one attribute will have a significant larger number (around 0.9) among three attributes. Hence, variational encoder is introduced to encode data to latent vectors (In this article is a 4-dimensional distribution vector.) while keeping useful information.



Temp 8am			Temp 12pm			Temp 4pm			Temp 8pm			BP Systolic			BP Diastolic			Nausea			Abdominal Pain		
Slight	Mod	High	Slight	Mod	High	Slight	Mod	High	Slight	Mod	High	Slight	Med	High	Slight	Med	High	Slight	Med	High	No	Yes	
0.1013	0.929	0.842	0.0562	0.7416	0.6964	0	0.7896	0.6821	0	0.6884	0.8575	0	0.7626	0.8238	0.0352	0.8114	0.6855	0.0652	0.5279	0.8177	0	0.9594	Sars.dat
0.8827	0.1286	0	0.9332	0.0946	0	0.9204	0.0823	0	0.8639	0.152	0	0.9144	0.0285	0	0.9047	0.0287	0	1	0	0	1	0	Normal.dat
1	0	0	1	0	0	1	0	0	1	0	0	0	0.2427	0.8081	0	0.2431	0.9252	1	0	0	1	0	High BP.dat
0.0827	0.8573	0.8759	0.1332	0.7893	0.8858	0.1204	0.7645	0.8582	0.0639	0.904	0.8846	1	0	0	1	0	0	1	0	0	1	0	Pneumonia.dat

Chart 1. A quick peek of datapoints in SARS-CoV Dataset

3 Experimental Results and Analysis

The baseline I choose here is CasCor Network since Var-Casper I introduce in this article is a modified version of Casper [6], and the task for my previous work [17] is the same as here which is to classify the SARS patients, Pneumonia patients, people with high blood pressure and normal people. Since the dataset is a small sample dataset with 4000 datapoints in total, I expect that choosing 2-layer variational encoder is appropriate for this SARS classification problem. The dimension of the input datapoints is 12 as mentioned before, and the labels use one-hot encoding. I set the number of epochs to 90 to control variables and compare Var-Casper to my previous work. The optimizer used here is RMSprop which is the same as described in Casper [6], with momentum=0.9, weight_decay=0.00001, centered=True. The main loss function is the cross-entropy loss function, which is usually for classic classification problems [18]. I am intended to use the accuracy on test dataset to evaluate the performance. The accuracy and loss result of the previous CasCor and Var-Casper are as follows:

Algorithm	Final Training Loss	Final Test Accuracy
CASCOR	0.472	74.351%
VAR-CASPER	4.967×10^{-8}	100%

From the chart above, it's obvious that Var-Casper has dominant performance on SARS-CoV dataset. The final test accuracy is 100% which indicates that the Var-Casper can solve more complex problem than SARS classification. Also, the final training loss of Var-Casper is close to zero, which indicates that generative model Casper can precisely match the problem complexity with the complexity of the neural network.

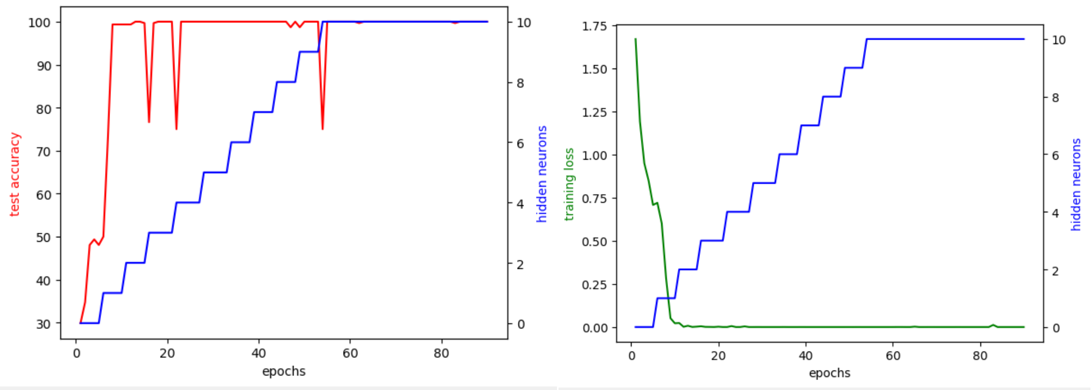


Fig5. Experimental Result of Var-Casper

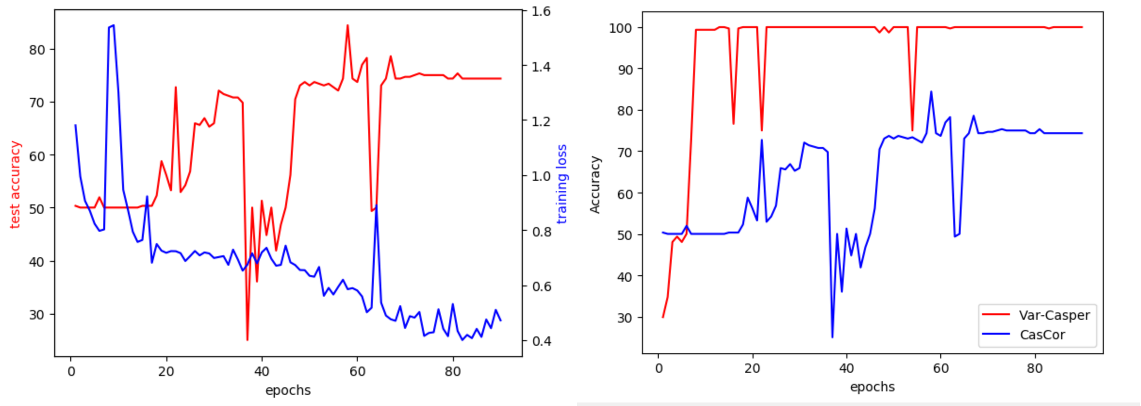


Fig6. Previous work CasCor and Comparison between Cascor and Var-Casper

The baseline I choose is CasCor from my previous work. The loss and test accuracy curve of the baseline is shown in Fig4.1. The performance is not very consistent due to its randomly initialized latest neurons. The lowest test accuracy can be nearly the same as random answers.

From Fig5., It's observed that test accuracy of Var-Casper rises from 30% up to 95% in the first 10 epochs with 2 neurons added. Afterwards, the model quickly finds the global minimum of the loss function in around 15 epochs, where the test accuracy reaches 100%. Usually the algorithm should converge and halt if the test accuracy reaches the top, here the algorithm runs the entire 90 epochs to make comparison with the baseline. There are several accuracy drops in the following epochs due to adding bad initialized neurons. From Fig5.1., adding two neurons is enough to get an appropriate 95% test accuracy. After fine-tuning the accuracy can reach 100% in a dozen of epochs and the algorithm should halt there. There are several reasons for the dominant performance of Var-Casper compared to CasCor. First of all, Casper does not freeze any previous added neuron, and CasCor may freeze poor feature extractors which is difficult for output weight to adjust their poor features. Since the algorithm still allows the previous poor hidden neurons to update their input weights by a small learning rate (L3), the poor neurons still have chance to update themselves. Secondly, the original data space is quite sparse that tiny changes in these sparse space may lead to the inconsistent classification results of the model. Hence, variational encoder is introduced to solve the sparse problem in this article. Thirdly, according to Gedeon, T. D. (1997), Casper can resume the property of CasCor that the latest neurons are the latest feature extractors since the learning rate of the previous hidden neurons is small with respect to the learning rate (L1) of the latest hidden neurons.

Noticed that if Var-Casper is over-trained, there may be an accuracy drop due to latest hidden neurons are lack of training. If the algorithm stops at these epochs, the model will have a relatively bad performance compared to other well-trained epochs. According to the Gedeon, T. D. (2005), the article that introduced this SARS-CoV-1 dataset, the original work is focused on the Fuzzy Signatures instead of classification, hence the results in this article can't be compared with the original paper.

As a result, the Var-Casper will have much better performance than CasCor in dataset-SARS-CoV-1.

4 Conclusion and Future Work

I have introduced a new architecture called Var-Casper, which is one of the constructive cascade neural networks with variational encoding inputs [1]. In this article I have a quick review on the topology of Casper and VAE, and also try to apply Var-Casper on SARS dataset in order to exceed the patient classification algorithm by CasCor in my previous work. It seems that the result of Var-Casper has an edge over my previous work CasCor model by around 25% test accuracy increase in SARS dataset. Therefore, this model can be more 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 Var-Casper 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 can quickly learn from data by a large learning rate while they are firstly added, the Var-Casper can easily extract features via optimizing the hidden to output weights. Var-Casper can also save training time since the network architecture is simple at the beginning and will grow depending on the problem complexity. The Var-Casper algorithm also faces problems such as accuracy drops due to bad feature extractors, and unlike CasCor, Var-Casper can improve these bad neurons in the following epochs by a small learning rate.

In the future I am going to explore more difficult classification problem to explore the limit of Var-Casper. The classification problem I analyse here is quite simple for Var-Casper and can not reveal many drawbacks of the model. One of the possible improvements to solve the accuracy drop problem is to use early-stopping and evaluation of the latest hidden neuron. Also, a good initialisation algorithm may be introduced to prevent performance drops due to the bad initialised hidden neurons. Additionally, the dimension of the data can be further pruned, masked to challenge the model. Since there are 4000 datapoints in SARS-CoV-1 dataset, it's possible to 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]

In conclusion, Var-Casper is a dynamic generative neuron network strategy which is timesaving, robust to sparse data and highly matches the complexity of the real-world problems. It's also shown that Var-Casper can ultimately solve SARS classification problem in current dataset with 100% accuracy.

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.

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
15. Weonyoung Joo, Wonsung Lee, Sungrae Park, Il-Chul Moon, Dirichlet Variational Autoencoder, Pattern Recognition, Volume 107, 2020, 107514, ISSN 0031-3203, https://doi.org/10.1016/j.patcog.2020.107514.
16. Jeremy Jordan. (2018). Variational autoencoders. Retrieved from https://www.jeremyjordan.me/variational-autoencoders/
17. Zheng, J (2021) Best match between SARS classification and Constructive Correlation Neuron Networks, *ABCs 2021*
18. Lima, B. V. A., Neto, A. D. D., Silva, L. E. S., & Machado, V. P. (2021). Deep semi-supervised classification based in deep clustering and cross-entropy. International Journal of Intelligent Systems, https://doi.org/10.1002/int.22446
19. Mozerov, M. G., Yang, F., & van de Weijer, J. (2019). Sparse data interpolation using the geodesic distance affinity space. IEEE Signal Processing Letters, 26(6), 943-947. https://doi.org/10.1109/LSP.2019.2914004