Fantastic Neural Network and how to train them

Zhiyuan Chen[1]
[0000−0003−3210−0324]

The Australian National University
this@zyc.ai

Abstract. Training a neural network is not easy. It is hard to fit, especially for a deep neural network, and may overfit very easily. In this work, we test two techniques of network reduction and residual network which could help training a neural network on a binary classification task.

Keywords: Neural Network · Machine Learning, Network Reduction

1 Introduction

Neural networks have shown extraordinary capabilities on most artificial intelligence tasks, such as computer vision (8), natural language processing (2), reinforcement learning (16), meta learning (15), etc. However, as the network goes deeper and deeper, it is more and more difficult to train the network since the number of parameters to train increases very fast, and gradient may vanish or explode.

Gedeon et al. proposed to use the cosine distance between the weights of the trained weights matrix to determine the similarity of neuron functionality, and combine those with similar effectiveness (6). However, their method remove both weights vectors if their angular separation are over 165°, which may results in disadvantages. A further study (3) showed that this technique is not sufficient for differentiating the functionality of the hidden neurons in image compression task.

He et al. proposed to add an identity shortcut connection between the input and output, which makes the convolution layers learn a residual only.

In this paper, we form a binary classification problem and test both techniques.

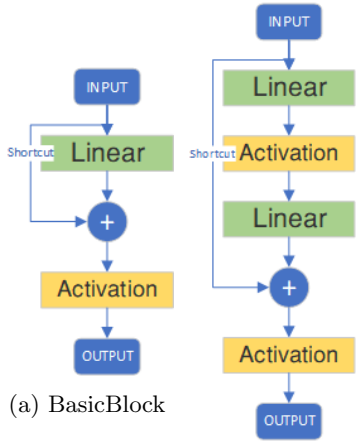
2 Related Work

Redisual Network Shortcut connections have been widely used in computer vision (13; 8). In which, input data is passed through a shortcut, and the convolution layers only learns a residual. This method enables the ability to train a deep neural network with more than 100 layers.

Yang et al. studied residual fully connected networks and showed the shortcut connection made signals decay more slowly during the back propagation, allowing effective training of deeper networks (17).

Bachlechner et al. proposed ReZero (1), indicating a learn-able parameter applied when combining the residual and all information can significantly increase the speed of train.

Network Reduction (7; 4; 5) proposed methods to reduce the size of network and training set.



(a) BasicBlock

(b) BottleNeck

Fig. 1: Residual Block

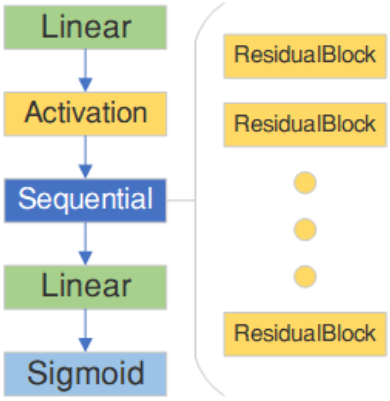


Fig. 2: Network Architecture

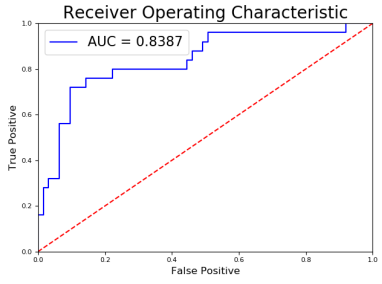


Fig. 3: ROC curve of the best performed combination

3 Train a Neural Network efficiently

3.1 Dataset

Eye-gaze (10) researched how size of snippets influence the user experience. They conducted the experiment with a series of questionnaires and eye gaze device, and concluded that long snippets will not increase the accuracy of search but increase search time since it takes longer to read the summary and resulted in frequent scroll.
Eye-gaze dataset consists of many different attributes, including subject, task type, task number, shown task number, snippet length, time to first click, accuracy, satisfaction, scroll, clicked rank, and the fixation of title, url, snippet.
In this work, we test both methods on eye-gaze dataset, and determine if the object scrolled. To fit the real world environment, information such as subject, task number is deprecated. Only one of the attributes with the same meaning is left, such as time to first click and LOG(time to first click). We chose to let the neural network decides which one it prefer, hence, the data actually used is decided by the network architecture search technique.

3.2 Residual Block

Residual block is basically the same as a normal fully connected layer. The only difference is that there is a short cut connection in residual block. Since we will be adding the shortcut data with the output of fully connected layer, it is mandatory to ensure the number of input channels equals to the number of output channels as shown in Fig. 1a. This design might not be the best choice, since there are too many parameters to be trained for a fully connected layer. Inspired by Kaiming et al., we designed a bottleneck block, which combines two fully connected layer. The first fully connected layer downsamples to half of the number of input channels, and the second fully connected layer upsamples it back to the number of input channels as shown in Fig. 1b.

3.3 Network Reduction

Gedeon et al. suggested that weights vectors with an angular separations of up to 15° can be considered as sufficiently similar, thus, remove one of the weights vector and add its value to the other will not damage the performance of the network. And if two weights vectors have an angular separation of over 165°, both of them can be removed. (3).

We believe that the reason why two weights vectors have opposite direction is because of the imperfect initialisation. And the network has learned proper weights to make them cancel each other out and left with correct outputs. Roughly remove both weights vectors will damage weights had learned, and the correct way is to add them the same way we treat the weights vectors with an angular separations less than 15°.

4 Experiments

4.1 Network Architecture Search for model design

We use network architecture search to find the best performed neural network for both methods. Thanks to the advancement of GPU accelerating, we are able to search over 10, 000 combinations of different channels number,

activation functions, and dropout weights. We set the minimum number of channels to 16 and maximum number of channels to 8192 for both input channels and output channels of the hidden layer. Note that the number of input channels must match the number of output channels for residual network.

The initial weights is one of the most important prerequisites for fast convergence of feed-forward neural network (14). We searched different initialisation methods including xavier init, kaiming init, etc. Hyper-parameters such as random seeds and learning rate are also searched. Moreover, we searched the non-linear activation functions between Sigmoid, tanh, ReLU, PReLU and RReLU, and optimizer between Adam, AdaGrad and AdaDelta.

Each combination of network reduction is trained for 500 epochs, and since the deep neural network is generally hard to train, they are trained for 1, 000 epochs. The top 5% result of the AUC score is recorded as the performance to rule out outliers. The ROC curve of the best performed combination is illustrated in Fig. 3.

4.2 Experimental Settings

Dataset We train and test the method of Gedeon et al. (3) and our method on the dataset proposed by Kim et al. (10). We split the dataset to two non-overlapping parts, one with 200 rows for training and one with 88 rows for testing.

Training We formulate the problem as a binary classification problem, i.e. to find out if the subject scrolled. Hence, we add a sigmoid layer after the last fully connected layer to make the network a classifier and used binary crossentropy function as our loss function.

Evaluation We use the area under receiver operating characteristic curve (AUC) to evaluate the performance of our classifier.

Implementation Details We train our models on up to 64 NVIDIA 1080Ti GPUs with a batch size of 200 for training and 88 for testing.

5 Results

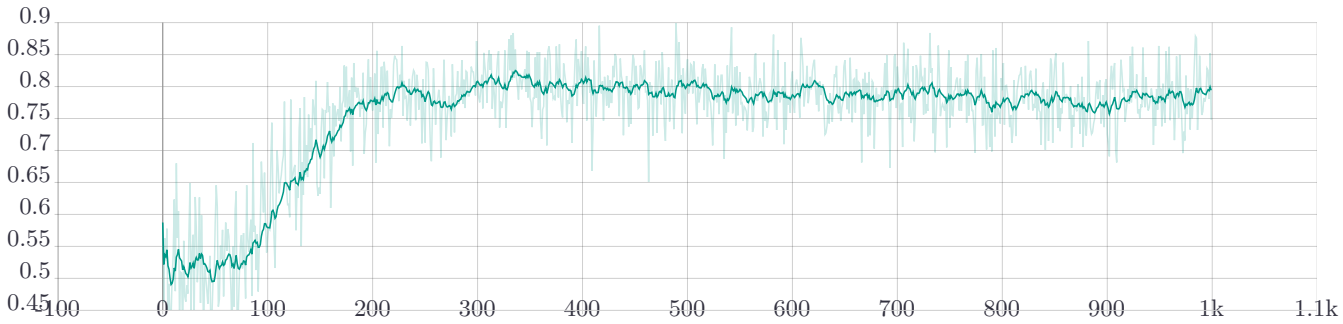


Fig. 4: AUC score with respect to epoch of residual network

5.1 Residual Network

Fig. 4 shows the AUC score of residual network. It can be found that the AUC score swings around 0.6 in the first 100 epoch before it increases. This is more clear in terms of F1 score with a threshold of 0.5 as shown in Fig. 5, which first decrease and approached 0 at epoch ~ 73, since the value of true positive and false positive are both zero. We believe it is a result of the unbalanced dataset, which there are more negative samples then positive samples, and the network have learned such pattern. We further tested our network with 64 to 512 hidden layers and discovered that this situation will get worse as the network goes deeper, and requires a higher learning rate to jump out of the local minima.

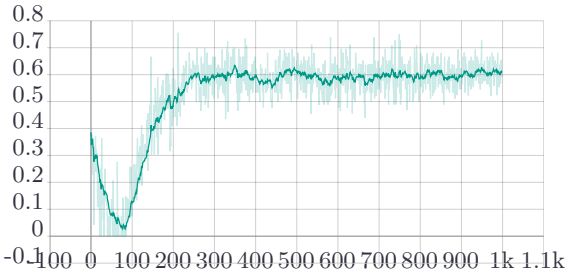


Fig. 5: F1 score on threshold of 0.5 with respect to epoch

layers	AUC	F1	Accuracy
8	0.9225	0.8295	0.6341
16	0.8997	0.7955	0.55
32	0.9092	0.6667	0.8409

Table 1: The performance of residual network with respect to the number of layers.

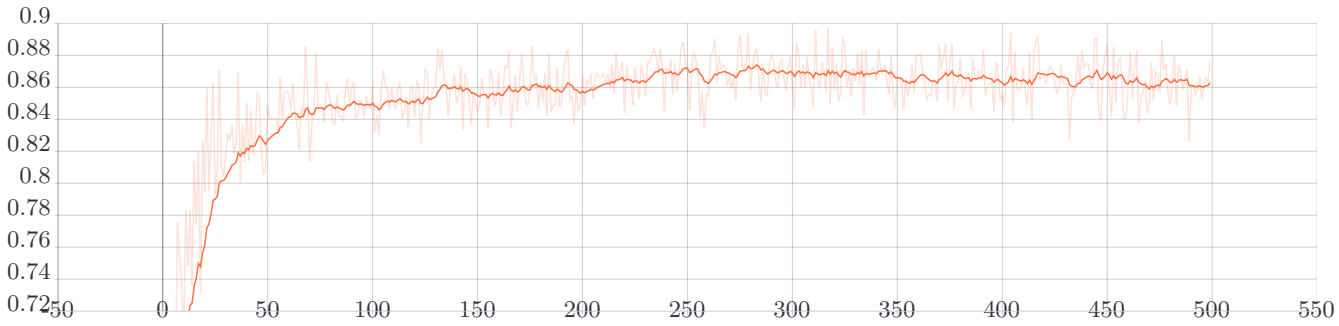


Fig. 6: AUC score with respect to epoch of network reduction

5.2 Network Reduction

Fig. 6 demonstrates the AUC score of reduced neural network differences between the original output and reduced output on degree=15°.

It can be found that there is no performance drop, in fact, it even increased a little for later epochs. We suspect it is resulted because there exists overfitting in later epoch, and the network reduction alleviate these overfitting. However, when we increases the degree to 30°, there would be significant performance decrease. Even though accuracy and precision are still acceptable, the F1 has a drawback of 7% in and the recall fells for 15%. 5.

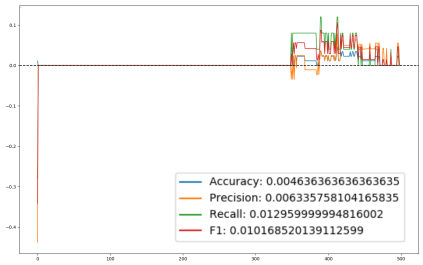


Fig. 7: The evaluation indexes at degree=15°

Apart from this, we also calculate the average differences between original output and reduced output within 1 to 90 degrees to test the ability of network reduction. The results can be found in Table 2.

epoch	F1	Accuracy	Precision	Recall
100	-0.29153	-0.04545	-0.07363	-0.41289
200	-0.264	-0.05467	0.06634	-0.38889
300	-0.18232	-0.03207	0.00833	-0.23467
400	-0.21633	-0.05442	-0.14211	-0.21733
500	-0.14534	-0.03573	-0.04413	-0.14667

Table 2: The reduce efficiency of network reduction technique.

6 Conclusion

In this work, we test the network reduction technique (3) and residual network in a binary classification task (10). Both networks achieve very high results of AUC over 0.88. And we can conclude that

Future Work The computation costs of fully connected layer are extremely high, which makes it less popular nowadays. For example, both AlexNet (11) and VGG (12) have three fully connected layer at the end, however, modern network such as ResNet (8) and DenseNet (9) all reduced the size of the fully connected layer layer at end. Thus, we reserve our views on the need of further improvement.

7 Acknowledgement

We would like to offer our sincerest appreciation to those who are working on the front line of the CONVID-19. Without whom, this paper will not be possible.

Bibliography

[1] Bachlechner, T., Majumder, B.P., Mao, H.H., Cottrell, G.W., McAuley, J.: Rezero is all you need: Fast convergence at large depth (2020)

[2] Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. In: 2019 The annual conference of the North American Chapter of the Association for Computational Linguistics (NAACL) (2018)

[3] Gedeon, T.D.: Indicators of hidden neuron functionality: the weight matrix versus neuton behaviour (1995)

[4] Gedeon, T.D., Bowden, T.G.: Heuristic pattern reduction. In: 1992 International Joint Conference on Neural Networks (IJCNN). vol. 2, pp. 449–453 (1992)

[5] Gedeon, T.D., Bowden, T.G.: Heuristic pattern reduction ii. In: 1993 International Conference for Young Computer Scientists (ICYCS). vol. 3, pp. 43–45 (1993)

[6] Gedeon, T.D., Harris, D.: Network reduction techniques. In: International Conference on Neural Networks Methodologies and Applications. vol. 1, pp. 119–126 (1991)

[7] Gedeon, T.D., Wong, P.M., Harris, D.: Balancing bias and variance: Network topology and pattern set reduction techniques. In: 1998 International Workshop on Artificial Neural Networks (IWANN). pp. 551–558 (1991)

[8] He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2015)

[9] Huang, G., Liu, Z., van der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016)

[10] Kim, J., Tohmas, P., Sankaranarayana, R., Gedeon, T., Yoon, H.J.: What snippet size is needed in mobile web search? In: 2017 Conference on Conference Human Information Interaction and Retrieva (CHILR). pp. 97–106 (2017). https://doi.org/https://doi.org/10.1145/3020165.3020173

[11] Krizhevsky, A., Sutskever, I., Hintons, G.E.: Imagenet classification with deep convolutional neural networks. In: 2012 Advances in Neural Information Processing Systems (NIPS) (2012)

[12] Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: 2015 International Conference on Learning Representations (ICLR) (2014)

[13] Srivastava, R.K., Greff, K., Schmidhuber, J.: Highway networks (2015)

[14] Thimm, G., Fiesler, E.: Neural network initialization. In: 1995 International Workshop on Artificial Neural Networks (IWANN). pp. 535–542 (1995)

[15] Vanschoren, J.: Meta-learning: A survey (2018)

[16] Vinyals, O., Babuschkin, I., Czarnecki, W.M., Mathieu, M., Dudzik, A., Chung, J., Choi, D.H., Powell, R., Ewalds, T., Georgiev, P., Oh, J., Horgan, D., Kroiss, M., Danihelka, I., Huang, A., Sifre, L., Cai, T., Agapiou, J.P., Jaderberg, M., Vezhnevets, A.S., Leblond, R., Pohlen, T., Dalibard, V., Budden, D., Sulsky, Y., Molloy, J., Paine, T.L., Gulcehre, C., Wang, Z., Pfaff, T., Wu, Y., Ring, R., Yogatama, D., Wünsch, D., McKinney, K., Smith, O., Schaul, T., Lillicrap, T., Kavukcuoglu, K., Hassabis, D., Apps, C., Silver, D.: Grandmaster level in starcraft ii using multi-agent reinforcement learning. naturel **575**, 350–354 (2019). https://doi.org/https://doi.org/10.1038/s41586-019-1724-z

[17] Yang, G., Schoenholz, S.S.: Mean field residual networks: On the edge of chaos. In: 2017 Advances in Neural Information Processing Systems (NIPS) (2017)