Machine Learning and Artificial Intelligence
A.J. Tallón-Ballesteros and C. Chen (Eds.)

doi:10.3233/FAIA200780

A Study on the Effect of DropConnect to Control Overfitting in Designing Neural Networks

Hyun-il LIM[1]
Department of Computer Engineering, Kyungnam University, South Korea

Abstract. The neural network is an approach of machine learning by training the connected nodes of a model to predict the results of specific problems. The prediction model is trained by using previously collected training data. In training neural network models, overfitting problems can occur from the excessively dependent training of data and the structural problems of the models. In this paper, we analyze the effect of DropConnect for controlling overfitting in neural networks. It is analyzed according to the DropConnect rates and the number of nodes in designing neural networks. The analysis results of this study help to understand the effect of DropConnect in neural networks. To design an effective neural network model, the DropConnect can be applied with appropriate parameters from the understanding of the effect of the DropConnect in neural network models.

Keywords. DropConnect, neural network, overfitting, machine learning

1. Introduction

As information technology advances, the software is becoming an important factor in solving various types of problems. Machine learning [1-3] is a way to solve problems through training models with data using software technology. This is a way to solve problems using data models trained with previous data. As machine learning has been used in various areas, the design of the machine learning model for solving problems is important in improving the accuracy of the model. Therefore, it is necessary to understand the features of the machine learning model to design it well.

In training neural network models, the accuracy of results should be increased from the information on the training data. The overfitting problem [4-6] may occur when the trained models are excessively customized for the training data, which can lead to poor performance in general data. Therefore, it is needed to reduce such overfitting in training neural network models [5]. The DropConnect [7, 8] is an approach to reducing overfitting y temporarily excluding connected links in the process of training. Such simplification of the link structures of neural networks can reduce the model's over-customization of training data.

In this paper, the effects of DropConnect according to various factors in a deep neural network are analyzed so that the effects of DropConnect can be understood to be

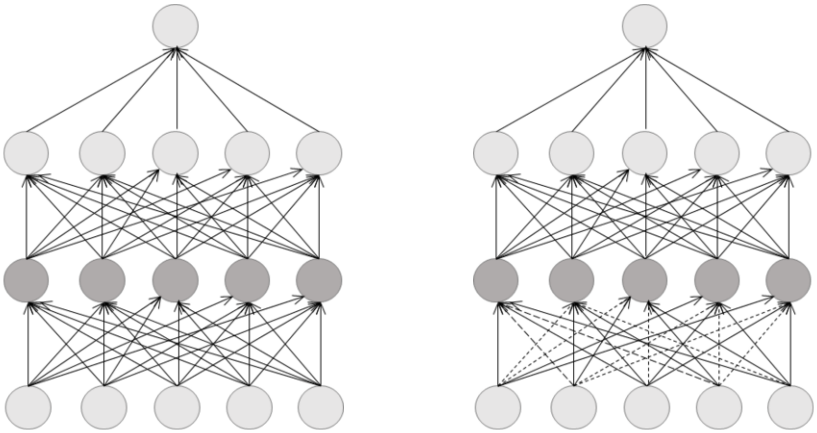
[1] Corresponding Author: Hyun-il LIM, Department of Computer Engineering, Kyungnam University, 7 Kyungnamdaehak-ro, Masanhappo-gu, Changwon, Gyeongsangnam-do 51767, South Korea; E-mail: hilim@kyungnam.ac.kr.

effectively applied in designing neural networks. The experimental results presented in this paper can be considered to apply DropConnect with several factors to reduce overfitting in designing deep neural networks.

2. The Concept of DroConnect in Deep Neural Networks

To create a neural network model, a training process is carried out on training data. At this training process, too dependent customization on the training data results in poor performance in applying in real data. This problem is mainly caused by the dependence of the model on noise data that may be included in the training data. This excessive dependence of the neural network model on the training data is called overfitting. To overcome this problem, it is needed to consider an effective method to reduce overfitting in designing neural networks.

The DropConnect is a way of temporarily excluding several links connected between nodes in the original neural network based on the DropConnect rate at a training phase. The temporary removal of links in training can reduce excessive dependence of the model on the training data. So, an active application of DropConnect can reduce the overfitting in training neural network models. In applying DropConnect to neural network models, several factors should be considered to achieve good performance, such as DropConnect rate and the number of nodes in neural network models.



(a) An example of a deep neural network. **(b)** After applying DropConnect to the neural network.

Figure 1. The structure of a deep neural network after applying DropConnect.

Figure 1 shows the structure of applying DropConnect to the hidden layer of a deep neural network. Figure 1(a) shows an example of a general deep neural network model that is fully connected to the nodes on each layer. By applying DropConnect to the neural network, several links are temporarily removed from the training of the network according to the DropConnect rate at the training phase of the model. Therefore, several dashed links in Figure 1(b) are temporarily excluded according to the DropConnect rate, and only connected links participate in the training of the neural network model. Such simplification of the structure of a fully connected neural network model can reduce the overfitting of the network by reducing excessive customization of the model to the training data. In applying DropConnect to neural network models, the effects of applying DropConnect should be reflected to determine several factors for designing neural networks, such as the DropConnect rate, and the structure of each layer. For example, how many nodes should be placed in the original neural network models or how many links to be excluded in training the neural network models by DropConnect to achieve good performance.

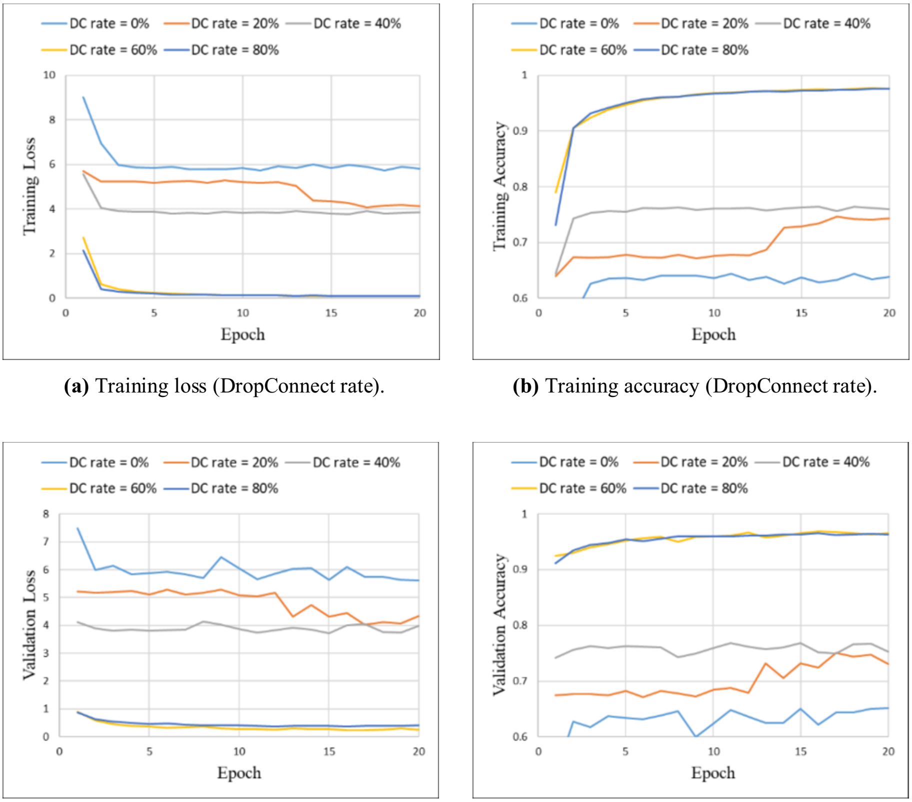
3. The Effect of Applying DropConnect according to the DroConnect Rate

In this experiment, the effect of applying DropConnect according to the DropConnect rate was analyzed in a neural network model. This experiment was performed in a neural network for recognizing the handwritten digits of the MNIST [9]. The MNIST database is popularly used in designing and evaluating machine learning models for image recognition [10-12]. The neural network was designed with one input layer for input data and two hidden layers of 128 and 512 nodes, respectively. The output layer was designed with the softmax function to classify handwritten digits into the corresponding number. The neural network model was implemented in Python with Keras. To analyze the effect of DropConnect in the neural network model, the accuracy and loss values were measured with the varying DropConnect rates between 0% and 80% for the second hidden layer.

Figure 2 shows the experimental results on the effect of applying DropConnect to the neural network model according to DropConnect rates. The DropConnect rate means the ratio of temporarily excluded links at each training epoch, and it should be determined in the design of neural network models. The graphs show how the loss and accuracy change as the training epoch progresses. In the graphs, as the training epoch progressed, training losses decreased and accuracies increased in common. In the training loss and accuracy graphs, the results show that the higher the DropConnect rate, the lower the training loss and the higher the accuracy. The high DropConnect rate means that more connected links will be excluded at each training phase. Therefore, the remaining links can be highly adjusted to the training data because only the weights of links participating in the training are adjusted. Thus, the loss and accuracy improve as the training epoch progresses, and the overall training accuracy of the model is expected to be improved.

Validation loss and accuracy were somewhat different from the training loss and accuracy. The validation loss and accuracy no longer improve at a certain point, even if the training epoch progresses. The high DropConnect rate results in lower loss and higher accuracy, but repeated training does not improve the loss and accuracy. In this experimental result, it can be confirmed that the validation loss and accuracy of the model do not improve after the fifth epoch of training. These results indicate that repeated training no longer affects the improvement of results for the validation data because the model is highly overfitted to the training data. The results show the lowest loss and highest accuracy when the DropConnect rate is around 60%. If the DropConnect rate is determined above the appropriate threshold, it can be confirmed that too many links are excluded to train the neural network model, so the training of the model cannot be performed sufficiently.

In summary, losses show better results when DropConnect is applied to the neural network. It is because the loss reduces the overfitting of the model. However, the excessive DropConnect rate can decrease accuracy because too many links between nodes are excluded in training the model.



(a) Training loss (DropConnect rate).

(b) Training accuracy (DropConnect rate).

(c) Validation loss (DropConnect rate).

(d) Validation accuracy (DropConnect rate).

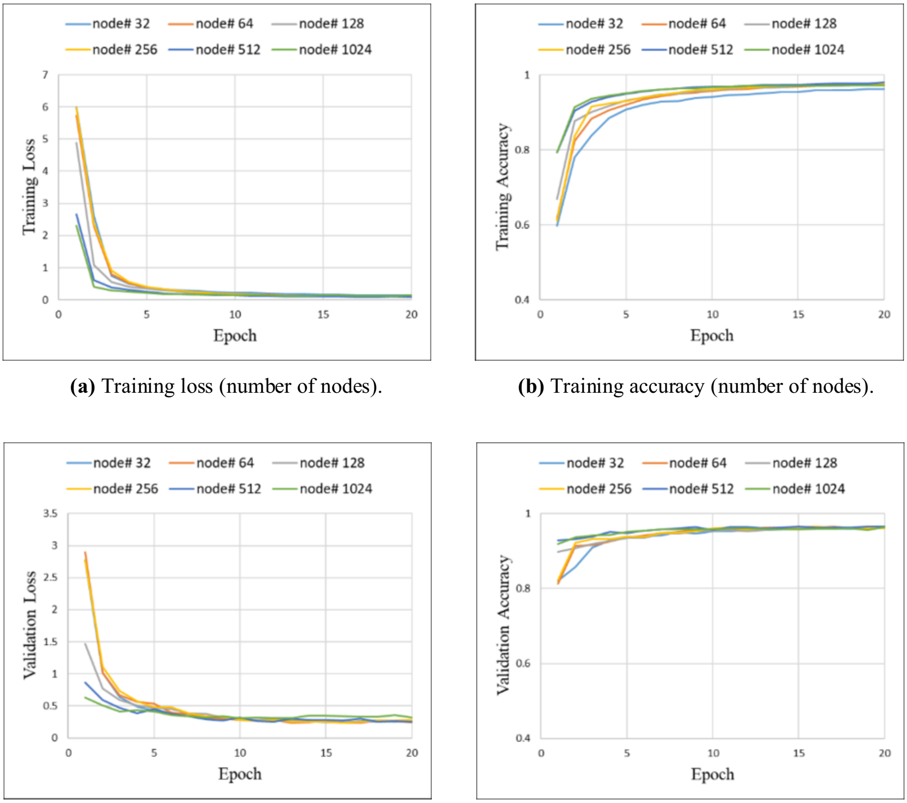
Figure 2. The experiments on the effect of applying DropConnect to a neural network (DropConnect rates).

4. The Effect of the Number of Nodes in Applying DropConnect

In this experiment, the effect of applying DropConnect was analyzed in a neural network model according to the number of nodes. This experiment was performed with a model for recognizing the handwritten digits of the MNIST as in the previous section. The number of nodes of neural network models is an essential design factor to achieve a good training model. So, it should be considered together with the effect of DropConnect to complete the structure of neural network models. In this experiment, the DropConnect was applied with the rate of 60% and the number of nodes was experimented with from 32 to 1024 at the second hidden layer of the neural network.

Figure 3 shows the experimental results of analyzing the effects of the number of nodes of hidden layers in applying DropConnect. In the early epoch of training, the training loss for the neural network model with many nodes was lower than that with the fewer nodes, so the larger number of nodes could reduce training loss. However, as the training epoch progressed, the training loss remained almost alike, regardless of the number of nodes. The training accuracy showed slightly high for the neural network models with large numbers of nodes.

The validation loss for the model with the largest number of nodes (1024) showed the smallest value in the early epochs of training. As training progressed, the validation loss increased rather than the other models with less number of nodes. Unlike the training loss, high validation loss implies the possibility of overfitting in the trained model. Therefore, a complex model with a larger number of nodes than necessary can be more likely to be overfitted. In the early epoch of the training, the training accuracy was different in terms of the number of nodes. However, as the training epochs progressed, validation accuracy maintained similar regardless of the number of nodes. Therefore, it can be seen that the number of nodes does not have a significant impact on models with well-applied DropConnect rates.



(a) Training loss (number of nodes).

(b) Training accuracy (number of nodes).

(c) Validation loss (number of nodes).

(d) Validation accuracy (number of nodes).

Figure 3. The experiments on the effect of applying DropConnect (the number of nodes).

In the experimental results, in the early epochs of training, the training loss of the model with the largest number of nodes showed the best result. However, as training progressed, the training loss maintained similarly regardless of the number of nodes. In the result of validation loss, the model with many nodes was overfitted, and its validation loss was increased as compared to the other models with less number of nodes. So, it can be seen that too many nodes do not help to train the neural network model.

In applying the DropConnect in designing neural networks, the effect of applying DropConnect should be understood, and the proper application of DropConnect can help to achieve stable results of neural network models. In future work, we plan to analyze the effect of DropConnect in reducing overfitting for applications in various areas together with the other design factors of neural network models, such as the structure of nodes, links, and layers of neural networks.

5. Conclusion

In this paper, the effect of applying DropConnect according to the DropConnect rates and the number of nodes in a neural network model has been analyzed. In the experimental results, it is confirmed that applying DropConnect is helpful to improve the accuracy of the trained model by temporarily excluding several links of the model to reduce overfitting. To apply the DropConnect to neural network models, the effect should be understood and the DropConnect rate should be applied properly to achieve stable results because the inconsiderate application of DropConnect may reduce the accuracy. The proper application of DropConnect is expected to help to design neural network models for controlling overfitting and improving the accuracy of the models.

Acknowledgements

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (Ministry of Education) (No. NRF-2017R1D1A1B03034769).

References

[1] Kevin PM. Machine learning: a probabilistic perspective. The MIT Press. 2012.
[2] Shai SS, Shai BD. Understanding machine learning: from theory to algorithms. Cambridge University Press. 2014.
[3] Pedro D. A few useful things to know about machine learning. Communications of the ACM. 2012; 55(10): 78-87.
[4] Salman S, Liu XW. Overfitting Mechanism and Avoidance in Deep Neural Networks. 2019.
[5] Xue Y. An overview of overfitting and its solutions. J. Physics: Conference Series. 2019. 1168(022022).
[6] Paul C, David J. Overfitting explained. In: Proceedings of the Sixth International Workshop on Artificial Intelligence and Statistics. 2000.
[7] Li W, Zeiler M, Zhang SX, Cun YL, Fergus R. Regularization of neural networks using dropconnect. In: Proceedings of the 30th International Conference on International Conference on Machine Learning; 2013 June; vol. 28, p. 1058-1066.
[8] Jensun R, Sascha S, Thomas V. DropConnect for evaluation of classification stability in learning vector quantization. In: Proceedings of European Symposium on Artificial Neural Networks (ESANN 2019); 2019 April 24-26; Belgium; p. 19-24.
[9] The Mnist Database of handwritten digits, http://yann.lecun.com/exdb/mnist/.
[10] Alejandro B, Yago S, Pedro I. A survey of handwritten character recognition with MNIST and EMNIST. Applied Sciences. 2019 August; 9(15):3169.
[11] Wu M, Zhang Z. Handwritten Digit Classification using the MNIST Data Set. 2010.
[12] F. Siddique, S. Sakib and M. A. B. Siddique. Recognition of handwritten digit using convolutional neural network in python with tensorflow and comparison of performance for various hidden layers. In: Proceedings of the 5th International Conference on Advances in Electrical Engineering (ICAEE), 2019; Dhaka, Bangladesh; p. 541-546.