International Journal of Advanced Smart Convergence Vol.9 No.2 173-178 (2020)
http://dx.doi.org/10.7236/IJASC.2020.9.2.173

IJASC 20-2-20

A comparison of methods to reduce overfitting in neural networks

Ho-Chan Kim*, Min-Jae Kang**

**Professor, Department of Electrical Engineering, Jeju National University, Korea, hckim@jejunu.ac.kr*
***Professor, Department of Electronic Engineering, Jeju National University, Korea, minjk@jejunu.ac.kr*

Abstract

A common problem with neural network learning is that it is too suitable for the specificity of learning. In this paper, various methods were compared to avoid overfitting: regularization, drop-out, different numbers of data and different types of neural networks. Comparative studies of the above-mentioned methods have been provided to evaluate the test accuracy. I found that the more data using method is better than the regularization and dropout methods. Moreover, we know that deep convolutional neural networks outperform multi-layer neural networks and simple convolution neural networks.

Keywords: *neural networks, overfitting, regularization, drop-out, test accuracy.*

1. Introduction.

The neural network contains several nonlinear hidden layer expression models for learning very complex relationships between inputs and outputs. However, many of these complex relationships caused by sampling noise due to limited training data exist in the training set, but in practice they are not. This is an over-fitting problem in neural networks and many methods have been developed to reduce it [1]. Commonly used methods are normalization, drop-out and data augmentation.

In this paper, we attempt to compare methods of reducing over-fitting. Section 2 briefly describes the process of supervised training of neural networks and an over-fitting, Section 3 discusses the methods of reducing over-fitting, and in Section 4, we use MNIST dataset for an experiment to compare the different approaches for reducing overfitting. We recognized that the neural network is improved as more data is used in training. And also it has been shown that a deep convolution neural network achieved the best test accuracy among other type of neural networks.

2. Over-fitting in Supervised Training

The process of supervised training in neural networks updates the weights in a way that reduces a loss function. The most common loss functions are mean squared error and cross-entropy error. The mean squared

Manuscript Received: May. 12, 2020 / Revised: May. 15, 2020 / Accepted: May. 21, 2020
Corresponding Author: minjk@jejunu.ac.kr
Tel: +82-64-754-3666, Fax: +82-64-756-1745
Professor, Department of Electronic Engineering, Jeju National University, Korea

error is

$$J(w) = \frac{1}{2}\sum_k (y_k(w) - x_k)^2 \quad (1)$$

And the cross-entropy error is as follows

$$J(w) = -\sum_k t_k \log y_k(w) \quad (2)$$

Where y_k is the output of the neural network, t_k is the true value, and k is the number of dimensions of the data. Neural network training is the process of finding the optimal weight (w) so that the output (y_k) of the neural network is close to the true value (t_k), that is, the loss function is minimum [2].

The one of biggest problem in training neural networks is the over-fitting of training data. Overfitting occurs when the graph fits the training data too accurately, as shown in Figure 1. Linear data with little noise is suitable for linear and polynomial functions. Polynomial functions fit perfectly, but linear functions can be more generalized. If both functions were used to estimate the fit data, the linear function should make better predictions.

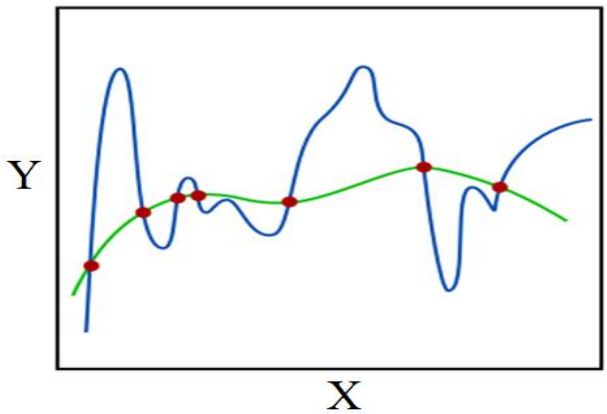


Figure 1. The blue line represents an overfitted model and the green line represents a regularized model. **(Source:** https://en.wikipedia.org/wiki/Regularization_(mathematics)**)**

Overfitting is caused by models with too complex parameters, with which models tries to predict the trend of data that is too noisy. The overfitting model is not accurate because the trend does not reflect the reality present in the data. The over-fit model can give good results for visible data (training set), but poor performance for invisible data (test set). The goal of the machine learning model is to generalize from training data to all data in the problem area. This is very important because the model is trying to predict future data that has never been seen before [4]. In this article, I will present four techniques to prevent overfitting while training neural networks.

3. Methods to avoid neural network overfitting

3.1 Use Regularization

Regularization is a technique to reduce the complexity of the model. It does so by adding a penalty term to the loss function. The most common techniques are known as L1 and L2 regularization: The L1 penalty aims to minimize the absolute value of the weights [5]. This is mathematically shown in the below formula.

$$J_R(w) = J(w) + \frac{1}{2}\lambda\sum_i |w_i| \quad (3)$$

The L2 penalty aims to minimize the squared magnitude of the weights. This is mathematically shown in the below formula.

$$J_R(w) = J(w) + \frac{1}{2}\lambda \sum_i w_i^2 \quad (4)$$

Here, these equations may be a little modified. If we interpret the L2 penalty in order, the new cost function adds the regularization part to the existing cost function. The regularization part consists the sum of squares of each weight multiplying by strength constant called lambda. It can be seen as What this means, we need to find the value where the value of the new cost function is minimum. When we find the minimum value, we want the weights to be the minimum. The lower the weight, the lower the probability of overfitting, it can be said that the curved curve is drawn like the green in the Figure 1.

3.2 Use Dropout

Dropout literally omits neurons of the network. If neurons of the network are removed randomly as shown in the Figure 2 while training of the neural network, this technique has proven to reduce overfitting to a variety of problems involving image classification, image segmentation, semantic matching etcetera. the omitted network will not have a significant effect on learning [6].

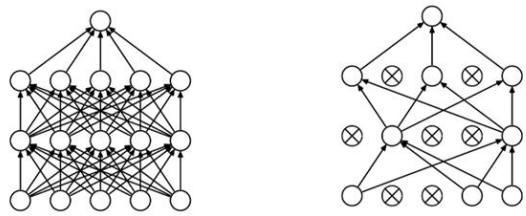


Figure 2. Dropout training for Neural Net a) Standard Neural Net b) After applying dropout **(Source:** https://en.wikipedia.org/wiki/Regularization_(mathematics)**)**

Ensemble learning is a very intuitive approach to addressing the problem of overfitting. If the problem of overfitting is due to our model learning too much from the training data, we can fix that by training multiple models, each of which learn from the data in different ways. If each component model learns a relationship from the data that contains the true signal with some addition of noise, a combination of models should maintain the relationship of the signal within the data while averaging out the noise. However, ensemble learning needs big data and time consuming for training neural network in different ways. But the concept of ensemble learning to address the overfitting problem still sounds like a good idea if there are enough time and data. Dropout is the technique to solve problems of both. Dropout uses same data but shows effect of using multiple models of neural network by randomly dropping neurons from the network during training [5].

3.3 Use Different number of data

Overfitting problem is known to be reduced as training data is increased. In this paper, different number of data are used for training to compare with other overfitting reducing methods. One of powerful methods to increase data is data augmentation for deep learning models. The augmented data will represent a more comprehensive set of possible data points, thus minimizing the distance between the training and validation set, as well as any future testing sets. Data augmentation simply means increasing size of the data that is increasing the number of images present in the dataset. Some of the popular image augmentation techniques are flipping, translation, rotation, scaling, changing brightness, adding noise etcetera. Using these techniques,

a lot of similar images can be generated [7]. This helps in increasing the dataset size and thus reduce overfitting. The reason is that, as we add more data, the model is unable to over-fit all the samples, and is forced to generalize.

3.4 Use Different types of Neural Network

Because of the fundamental limitations of single-layer neural networks, neural networks were inevitably developed in a multi-layered structure. However, it took decades to pursue a multi-layer neural network, because there was a no learning rule for multilayer neural network. In1986, this problem was solved by the development of an inverse propagation algorithm for back propagation and finally the learning problem of a multilayer neural network. The multilayer neural network consists of an input layer, a hidden layer, and an output layer as shown in Figure 3a. CNN became known to the world when CNN was used to win the world's leading institutions at the 2012 International Image Recognition Contest (ILSVRC) in a huge gap. The input data of a neural network composed only of multilayers is limited to a one-dimensional array form. One color photograph is three-dimensional data. If you need to train a multi-layer neural network for photo data, you need to flatten the three-dimensional photo data in one dimension. In the process of flattening the photo data, spatial information is inevitably lost. As a result, due to the lack of information due to the loss of image spatial information, neural networks have limitations in extracting and learning features and increasing accuracy [8]. A model that can be trained while maintaining spatial information of an image is a CNN (Convolutional Neural Network). The typical CNN configuration is shown in Figure 3b. The CNN is divided into a feature extraction part that repeatedly stacks the convolution layer and the max pooling layer, and a fully connected layer, and is divided into a classification part that applies Softmax to the last output layer.

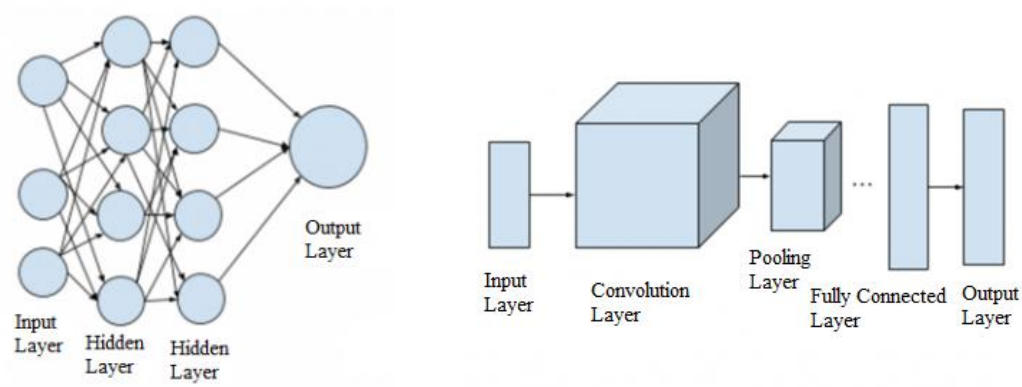


Figure 3. Different types of Neural Nets (a) Regular Neural Net (b) Convolution Neural Net

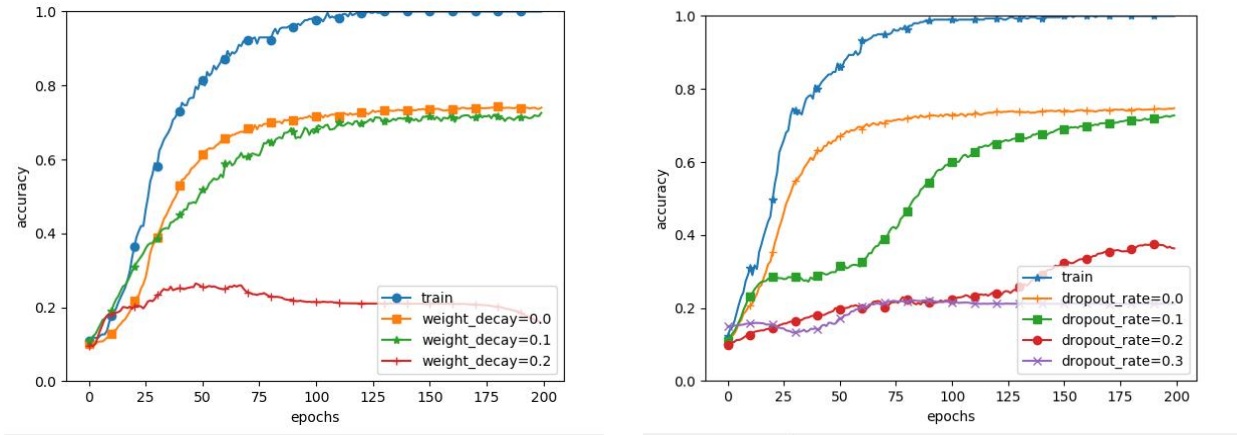
4. Empirical Results and Observation

The MNIST digits (LeCun et al., 1998a), dataset has 60,000 training images, 10,000 test images, each showing a 28x28 grey-scale pixel image of one of the 10 digits. Using this dataset, we tested the different types of reducing over-fitting methods such as weight decay, dropout. Also, we tested different number of data for training and different types of neural network for same purpose.

Figure 4 ,5 and 6 show the evolution of training and test accuracy with using different methods. Figure 4 shows the weight decay and dropout methods using 300 MNIST data. In Figure 4(a), the test accuracy using decay factors have been shown not better than without using decay factor. Figure 4b shows the dropout method

applied to the same data. Dropout method also shows not to improve the test accuracy during 200 training epochs. In figure 5, different number of data are used for experimenting over-fitting problem in neural network. As increased data number, the test accuracy is increased visibly. The test accuracy is seen to approach to 0.967 using 30000 data. As shown in Fig. 6, a convolution neural network (CNN) is tested and compared with a multi-layer neural network. Two types of CNN are used. One is simple and the other is deep neural network. The simple one consists of one stack of Convolution–ReLu–Pooling consecutive layers, while the deep one consists of three stacks of Convolution–ReLu–Pooling consecutive layers. In Figure 6a, the test accuracy of deep CNN is much improved to 0.895 using same data in Figure 4, while those are 0.745 and 0.769 with weight-decay and dropout method respectively as seen above. The results with deep CNN is shown in Fig. 6b using different number of data 300, 1000 and 5000. The test accuracy is close to 0.984 using 5000 data.

Through this experiment, it can be inferred that deep CNN is more suitable than other multilayer neural networks to reduce overfitting. And the number of data is considered more important than the weight-decay and dropout method for overfitting problem.



(a) (b)

Figure 4. Training and test accuracy using (a) Weight-decay (b) Dropout

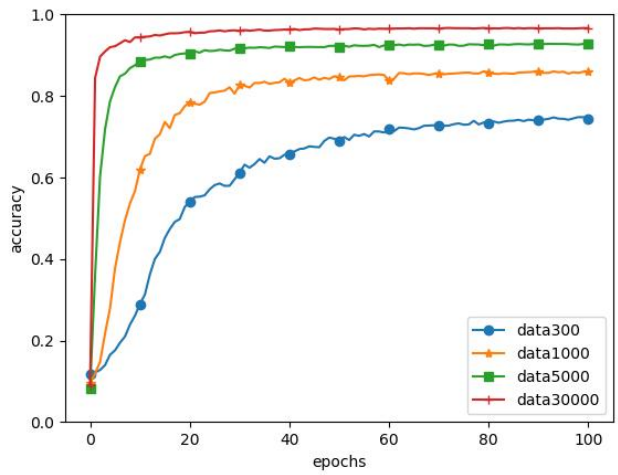


Figure 5. Test accuracy using different number of data for training

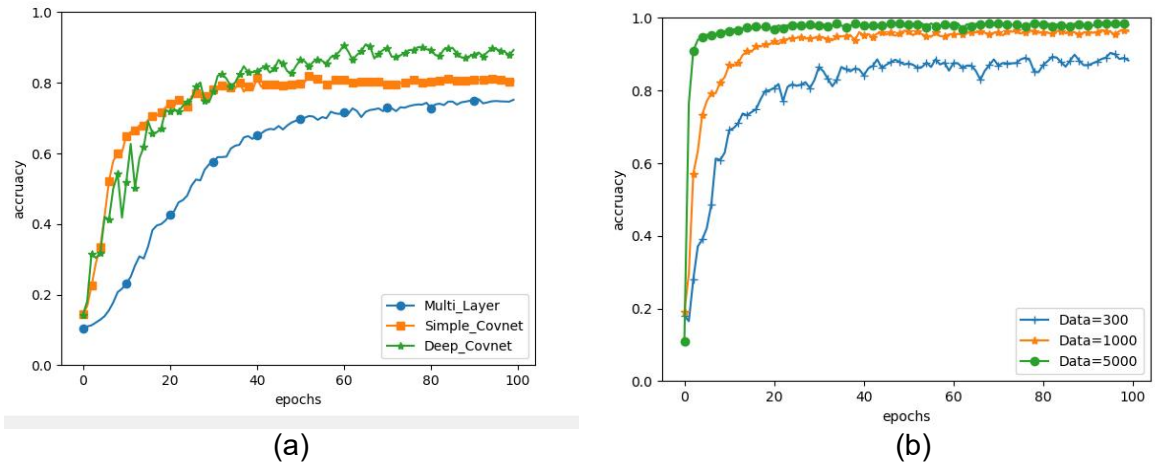


Figure 6. Test accuracy (a) different types of neural network (b) different number of data with CNN

5. Conclusion

In this paper, a study of four over-fitting reducing methods namely weight-decay, dropout, effect of data number and the different types of neural networks were used and compared. The results of our experiments indicate that deep CNN is more suitable than other multilayer neural networks to reduce overfitting. And the number of data is considered more important than the weight-decay and dropout method for overfitting problem.

Acknowledgement

"This work was supported by the research grant of Jeju National University in 2019"

References

[1] Smith, Craig S, "The Man Who Helped Turn Toronto into a High-Tech Hotbed," The New York Times. Retrieved 27 June 2017.

[2] Wu H, Shapiro J L. (2006) Does overfitting affect performance in estimation of distribution algorithms. Conference on Genetic and Evolutionary Computation. ACM, pp.433-434.

[3] Karystinos G N, Pados D A. (2000) On overfitting, generalization, and randomly expanded training sets. IEEE Transactions on Neural Networks, 11(5):1050.

[4] Yann LeCun, Yoshua Bengio & Geoffrey Hinton, "Deep learning," Nature volume521, pages436–444 (28 May 2015).

[5] Srivastava N, Hinton G, Krizhevsky A, et al. (2014) Dropout: a simple way to prevent neural networks from overfitting. Journal of Machine Learning Research, 15(1):1929-1958.

[6] Yip K Y, Gerstein M. (2009) Training set expansion: an approach to improving the reconstruction of biological networks from limited and uneven reliable interactions. State of the art of air pollution control techniques for industrial processes and power generation. Dept. of Civil Engineering, College of Engineering, University of Tennessee, pp.243-250.

[7] T. J. O'shea, J. Corgan, and T. C. Clancy, "Convolutional radio modulation recognition networks," in Proc. International conference on engineering applications of neural networks, 2016, pp. 213–226.