|  |
| --- |
| ***Regularization with Probabilistic Target Encoding*** |
| Sang-Hoon Oh\*  \*Mokwon University, Korea  E-mail:ohsanghoon16@gmail.com |
|  |

# Introduction

Usually we use FNNs(Feedforward Neural Networks) in many applications including classification or regression problems [1]. The proof that FNN can approximate any function with enough number of hidden nodes supports this research direction [2]. During training of FNNs, the EBP(Error Back-Propagation) algorithm minimizes MSE(mean-squared error function) of FNNs which has a weakness with slow learning convergence and poor generalization performance[3]. There are many attempts to propose error functions to improve the performance of EBP algorithm for sigmoidal outputs[4].

Especially, DNNs(deep neural networks) adopt the cross-entropy error function with softmax outputs[5]. Although DNNs have a powerful learning capability for real problems, overfitting to training samples gives us another challenge to improve performance for test samples. Stochatic pooling is a regularition method where the activation within a pooling region is picked randomly according to a multidimensional distribution. We can limit the number of parameters typically limiting the number of hidden nodes in each layer or limiting network depth. Weight decay is a simple form of added regularizer with a additional error term proportional to L1 norm of L2 norm of weight vectors. Among many regularization methods to alleviate the overfitting, dropout is the most famous because of its excellent effect including weight decay [6]. Also, a probabiistic target encoding has been proposed as a regularization method with a simple encoding of desired values for output nodes [7]. In this paper, we investigate the merge of the probabilistic target encoding and the dropout methods for better regularization effects.

# Feedforward Neural Networks (FNNs)

As shown in Fig. 1, FNN consists of an input vector **x**, a hidden node vector **h**, an output node vector **y**, and their connection weights. When an input vector  is presented to the FNN, a weighted sum to is given by and then the hidden node value is given by . Here,  is a bias and  is a weight between and. The *k*-th output node  is calculated through the same procedure of weighted sum and sigmoid transform using the weight and the hidden node value . When  is given for a specific training sample, we usually updates weights  and  to minimize the MSE function . Here, *P* is the number of training samples and *M* is the number of output nodes. EBP algorithm provides updating procedure of  and as follows[2]:

 (1)

 (2)

In classifications, we adopt the one-hot coding of the target value as follows:

(3)

# Cross-Entropy Error with SoftMax Output

In DNNs(deep neural networks) as shown in Fig. 2, we use the softmax output given by

(4)

where is the weighted sum to the output node. For applying DNNs to classification probems, we usually use the cross-entropy error function given by

(5)

Since , minimizing induces overfitting to training samples [3]. Dropout has the regularization effect by randomply drop out hidden nodes. Also, probabilistic target encoding alleviates the overfitting by substituting the desired values of ‘one’ or ‘zero’ with intermediate values between one and zero. Becaues there is no dependency between the regularization strategy of dropout and probabilistic target encoding, we can merge the two regularization methods for better effects. We investigate the merge of the two methods through simulations.

# 4. Discussion and Conclusion

In this presentation, we investigated the merge of dropout and probabilistic target encoding through simulation of various real problems.

5. References

[1] D. E. Rumelhart and J. L. McClelland, Parallel Distributed Processing, MIT Press, Cambridge, MA, 1986.

[2] K. Hornik, M. Stinchcombe, and H. White, “Multilayer Feed-forward Networks are Universal Approximators,”

Neural Networks, vol.2, 1989, pp. 359-366.

[3] S.-H. Oh, “Improving the Error Back-Propagation Algorithm with a Modified Error Function,” IEEE Trans. Neural Networks, vol.8, 1997, pp. 799-803.

[4] S.-H. Oh, “Statistical Analyses of Various Error Functions for Pattern Classifiers,” Proc. Int. Conf. Hybrid Information Tech., Sept. 22-24, Daejon, Korea, 2011, vol. 206, pp. 129-133.

[5] Y. LeCun, “LeNet-5, Convolutional Neural Networks,” Retrieved 16 November 2013.

[6] K. Baldi and P. Sadowski, “The Dropout Learning Algorithm,” Artificial Intelligence, vol.210, 2014, pp. 78-122.

[7] S.-H. Oh, “Probabilistic Target Encoding of Neural Networks with Softmax Output Nodes,” Submitted to Int. Journal of Contents, Aug. 2022.

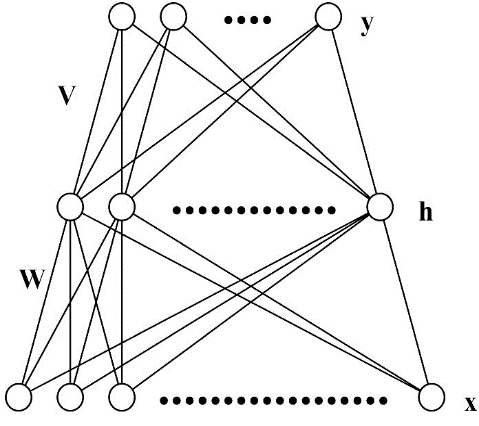


Figure 1 The architecture of FNN.

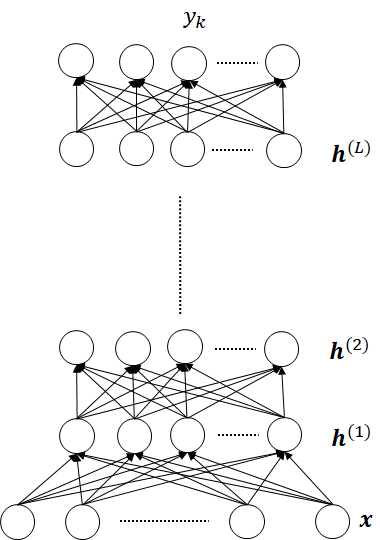


Figure 2. The architecture of Deep Neural Networks.