STUDY ON HANDWRITTEN RECOGNITION NEURAL NETWORK MODEL USING FINE TUNING AND SELU ACTIVATION FUNCTIONS

[1]JONGHO IM, [2]SANGJO KIM, [3]MIKYUNG KIM, [4]EUIYOUNG CHA

[1,2,3,4]Dept of Electricity and Electronic Computer Engineering, Pusan National University, South Korea
E-mail: [1]angelcrown24@gmail.com, [2]kimsjpk@gmail.com, [3]ddosun@pusan.ac.kr, [4]eycha@pusan.ac.kr

Abstract - In this paper, we describe how to implement handwritten recognition system using neural network model. Fine tuning was applied to the neural network model VGG16 to improve performance. As a result, this method achieved an accuracy of more than 80%. We also compared the performance of the model using selu as the activation function of VGG16 top model without input data normalization and the model using relu as the activation function of VGG16 top model after input data normalization. As a result, the loss converges more rapidly when the activation function selu is used, and it is confirmed that the activation function selu is effective for improving the learning speed.

Keywords - Fine tuning, Handwritten recognition, SELU, VGG16

I. INTRODUCTION

Handwritten recognition is one of the most popular applications in the field of artificial intelligence, as computers recognize various types of handwritten input. There is a need for a method of manually inputting a desired character on the screen with a specific pen of the device. This requires a technique in which the latest IT equipment recognizes the handwritten input by the user as an intended character. As a result, most of the recently released electronic devices have a system for recognizing handwritten. In this paper, we describe a handwritten recognition system implementation method using handwritten recognition neural network model.Fine tuning was applied to the neural network model VGG16 to improve performance, and the performance of the model using selu as the VGG16 topmodel activation function and the model using relu as the VGG16 topmodel activation function after input data normalization were compared.

II. DETAILS EXPERIMENTAL

2.1. Dataset used for experiments

We used the English, 62 classes of The Chars74K dataset as learning data. This dataset is a collection of 3,410 images, each with a handwritten number 0 to 9, an alphabetic upper case letter A-Z, and a lower case alphabet letter a-z, with 62 classes each of which consists of 55 images. Of the 3410 images in the dataset, 2273 images were used as training data, 569 images were used as validation data, and the remaining 568 images was used as test data.

2.2. Implementation for experiments

We used the VGG16 model as a pre-trained neural network model. Since the input size of this model is a 224x224 images of 3-channel, we transformed 1200x900 images of 1-channel into 224x224 images of 3-channel.

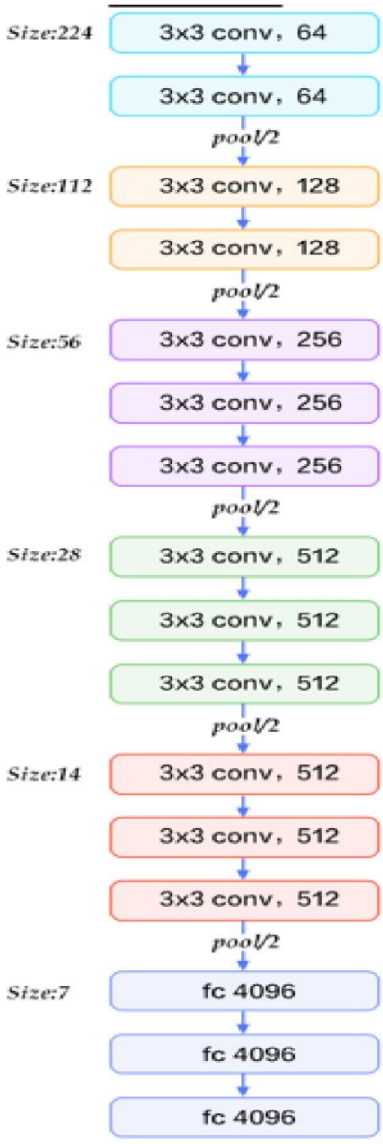


Fig.1. VGG16 used for learning

Input data normalization and fine tuning methods are known as a way to quickly learn neural networks. In the case of input data normalization, the input value is subtracted from the average of the input vector, divided by the variance of the input vector, and the input vector is normalized. Using this, the average of the input data becomes zero, and learning proceeds quickly [1]. In the case of fine tuning, it begins with copying (transferring) the weights from a pre-trained network to the network we wish to train [2].For the model using reluactivation function [3], we fed the input data to the VGG16 fine tuning model after normalization. The learning model excludes the top model of vgg16 (the last three fully connected layers) and is added two fully connected layers (hidden layers) which has 1024 outputs using relu as an activation function, and is added fully connected layer which has 62 outputs to classify the number of classes in a dataset.In order to compare the effect of the activation function selu[4] on the learning, unlike the above model, the model to be compared is not normalized to the input data, but the activation function selu is used with the same parameters as the above model.

III. RESULTS AND DISCUSSION

3.1. Result

In **Fig.2, Fig.3**, we can see that convergence is faster than using relu when using selu as an activation function in the learning process.The test data were input to the learning model using relu activation function after input data normalization and 80.88% accuracy was obtained. 81.43% accuracy was obtained in the learning model using selu activation function.

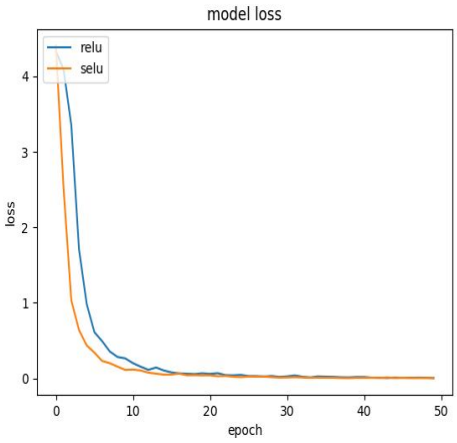


Fig.2. Comparison of loss for validation data per epoch during learning

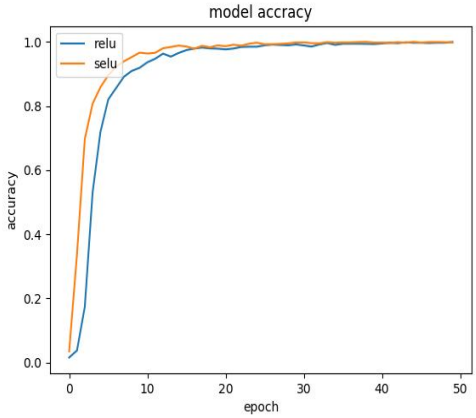


Fig.3. Comparison of accuracy for validation data per episode during learning

3.2. Discussion

In this paper, we describe the implementation of handwritten recognition neural network model using fine-tuning and compare the performance of the model using relu as the top model activation function after the input data normalization and the model using selu as the top model activation function.Using the weight of the pre-trained network that classifies images of general objects, fine tuning in handwritten recognition neural network model shows that test data accuracy is more than 80%, so fine tuning can be applied to handwritten recognition.In the learning process, it is shown that using selu activation function converges faster with validation data accuracy of 100% and loss 0 per epoch than using relu activation function.Therefore, we can confirm that the method of using selu activation function speeds up the learning of the neural network model than the input data normalization method.

REFERENCES

[1] LeCun, Y. A., Bottou, L., Orr, G. B., & Müller, K. R. (2012). Efficient backprop. In Neural networks: Tricks of the trade (pp. 9-48). Springer Berlin Heidelberg.

[2] Tajbakhsh, N., Shin, J. Y., Gurudu, S. R., Hurst, R. T., Kendall, C. B., Gotway, M. B., & Liang, J. (2016). Convolutional neural networks for medical image analysis: Full training or fine tuning?. IEEE transactions on medical imaging, 35(5), 1299-1312.

[3] Nair, V., & Hinton, G. E. (2010). Rectified linear units improve restricted boltzmann machines. In Proceedings of the 27th international conference on machine learning (ICML-10) (pp. 807-814).

[4] Klambauer, G., Unterthiner, T., Mayr, A., & Hochreiter, S. (2017). Self-Normalizing Neural Networks. arXiv preprint arXiv:1706.02515.

★★★