MLP Coursework 1: Activation Functions

s1781323

Abstract

This is the assignment of Machine learning practical lecture. The purpose of this study is to compare the performance of neural network with variants of the ReLU activation function, initialization method, and the number of layers. The author tested a variety of neural network with different activation function, a different number of layers, and different initialization method. The performance is measured by square error and accuracy on a validation set. The result shows that the more the number of layers increases, the more error increases. On the other hand, the accuracy does not change compared to error. Also, the proper initialization decrease validation error.

1. Introduction

How should we tune up neural network? Which function should we choose? This paper shows the performance of different models of neural network regarding depth, weight initialization, and activation function. This paper measures performance by accuracy and mean square error.

The data in the paper is called MNIST. MNIST stands for "Mixed National Institute of Standards and Technology databas," and it is the dataset of labeled images of handwriting numbers. Each image is 28 x 28 pixels, and the network predicts numbers with these 784-dimensional data.

2. Activation functions

ReLU: Restricted Linear Unit (ReLU) is first introduced by Nair and Hinton in 2011. This function is simpler than sigmoid function of tanh function, therefore thei runs fast and reduce the so called "The vanishing gradient problem".

$$\mathrm{relu}(x) = \max(0, x), \tag{1}$$

which has the gradient:

$$\frac{d}{dx}\mathrm{relu}(x) = \begin{cases} 0 & \text{if } x \le 0 \\ 1 & \text{if } x > 0. \end{cases} \tag{2}$$

Leaky ReLU: Leaky Restricted Linear Unit (ReLU) is introdused by Mass in 2013.

$$\mathrm{lrelu}(x) = \begin{cases} \alpha x & \text{if } x \le 0 \\ x & \text{if } x > 0. \end{cases} \tag{3}$$

which has the gradient:

$$\frac{d}{dx}\mathrm{lrelu}(x) = \begin{cases} \alpha & \text{if } x \le 0 \\ 1 & \text{if } x > 0. \end{cases} \tag{4}$$

where α is a constant; typically $\alpha = 0.01$.

ELU: Exponential Linear Units (ELU) is introdused by Clevert in 2015. When the mean of output in each layer is near zero, this function decrease the bias shift and the speed of learning increases.

$$\mathrm{elu}(x) = \begin{cases} \alpha(\exp(x) - 1) & \text{if } x \le 0 \\ x & \text{if } x > 0. \end{cases} \tag{5}$$

which has the gradient:

$$\frac{d}{dx}\mathrm{elu}(x) = \begin{cases} \alpha \exp(x) & \text{if } x \le 0 \\ 1 & \text{if } x > 0. \end{cases} \tag{6}$$

where α is a constant or a tunable parameter; typically $\alpha = 1$.

SELU: Scaled Exponential Linear Unit (SELU) is introdused by Klambauer in 2017.

$$\mathrm{selu}(x) = \lambda \begin{cases} \alpha(\exp(x) - 1) & \text{if } x \le 0 \\ x & \text{if } x > 0. \end{cases} \tag{7}$$

which has the gradient:

$$\frac{d}{dx}\mathrm{selu}(x) = \lambda \begin{cases} \alpha \exp(x) & \text{if } x \le 0 \\ 1 & \text{if } x > 0. \end{cases} \tag{8}$$

In the case of SELU, there is a theoretical argument for optimal values of the two parameters: $\alpha \approx 1.6733$ and $\lambda \approx 1.0507$.

3. Experimental comparison of activation functions

This section shows the rusult of test these 4 different activation functions with MNIST, and compare the performance. In this case, the network is constructed with 2 hidden layers and 100 hidden units per layer.

The results looks like the same. However, the minimum of validation error is minimum on ReLU model. Also the speed of decrease error on train error is fast on this model(Figure1-4).

Next, look at the accuracy (Figure5-7).Only network with Leaky Relu function marks more than 0.98 validation accuracy. Based on these result,contrary to expectations, Leaky

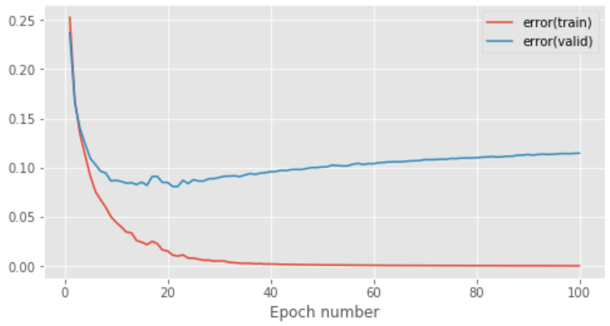

Figure 1. Error of ReLU. The minimum of validation error is 8.09e-02

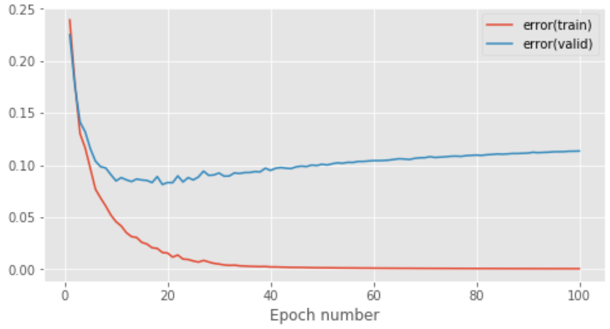

Figure 2. Error of Leaky ReLU. The minimum of validation error is 8.13e-02

Relu function achive the best perfomance on MNIST data although this function is older than ELU and SELU.

When the network has more layers, there might be more differences between models. However, this case there is only two hidden layers, so there are slight differences.

4. Deep neural network experiments

This section shows the experiments on deeper networks for MNIST. The two sets of experiments are to explore the impact of the depth of the network (number of hidden layers), and a comparison of different approaches to weight initialisation.

First, using Leaky ReLU Layer, the author test different networks with 2 - 8 hidden layers. The result is on Table 1. The author expects that deeper network is easier to overfit, and the accuracy decreases. However, the accuracy does not change dynamically, and the validation error increases in proportion to the depth of network. The accuracy is high enough, so the author expects that the accuracy does not improve anymore. Also, error increase because of overfitting, so it grows when there are more hidden layers.

Next, the author compares two initialization method, Glorot and Bengio's combined initialization, and Fan-in, Fan-out.

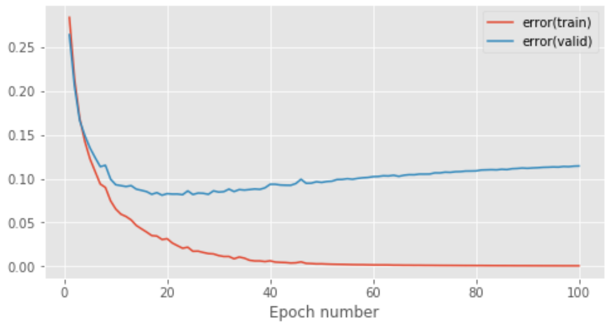

Figure 3. Error of ELU. The minimum of validation error is 8.12e-02

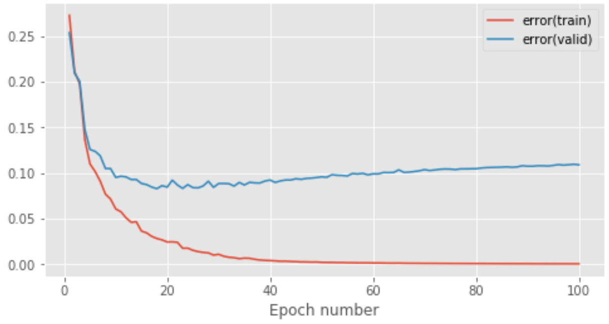

Figure 4. Error of SELU. The minimum of validation error is 8.29e-02

These initialization function is this:

$$
\begin{aligned}
Fan-in: w_i\tilde{}U(-\sqrt{3/n_{in}}, \sqrt{3/n_{in}}) \\
Fan-out: w_i\tilde{}U(-\sqrt{3/n_{out}}, \sqrt{3/n_{out}}) \\
GlorotandBengio's: w_i\tilde{}U(-\sqrt{6/(n_{in}+n_{out})}, \sqrt{6/(n_{in}+n_{out})})
\end{aligned}
\tag{9}
$$

where U is the uniform distribution.n_{in} is the number of incoming connections, and n_{out} is the number of outcoming connections.

This experiments uese layers with tow hidden LReLU layers and eight hidden LReLU layers.The result is on Table 2.

Depth	Min of Valid Error	Last Valid Error	Valid Accuracy
2	8.16E-02	1.12E-01	9.79E-01
3	8.30E-02	1.33E-01	9.80E-01
4	9.11E-02	1.48E-01	9.78E-01
5	9.45E-02	1.48E-01	9.80E-01
6	9.27E-02	1.66E-01	9.78E-01
7	9.38E-02	1.74E-01	9.80E-01
8	8.88E-02	1.74E-01	9.80E-01

Table 1. Relationship between depth of network and performance

Depth	Initializer	Min of Valid Error	Last Valid Error	Valid Accuracy
2	GB	8.16E-02	1.12E-01	9.79E-01
8	GB	8.88E-02	1.74E-01	9.80E-01
2	Fan-in	8.88E-02	1.19E-01	9.77E-01
8	Fan-in	9.51E-02	1.84E-01	9.79E-01
2	Fan-out	7.61E-02	1.04E-01	9.81E-01
8	Fan-out	9.52E-02	1.65E-01	9.79E-01

Table 2. Relationship between initialisation and performance

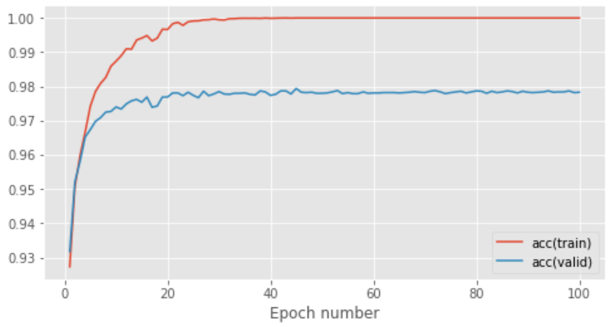

Figure 5. Accuracy of ReLU, the last accuracy of validation set is 9.78e-01

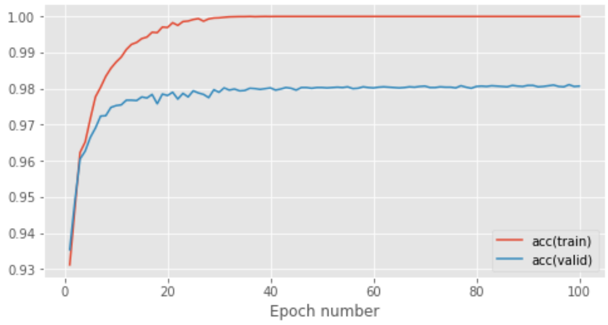

Figure 6. Accuracy of Leaky ReLU, the last accuracy of validation set is 9.81e-01

It show that the nework with 2 hidden layers and Fan-out initialiser mark the best performance. However, the graph of Glorot and Bengio's combined initialization is smoother (Figure 2, 10), so the Glorot and Bengio's approach makes learning stable.

Also, the author test initialiation for SELU layer recommended by Klambauer et al. [2017]. This function is based on Gaussian distribution with mean 0 and variance $1/n_{in}$.

The result is on Table 3.

It shows that the reccomendation strategy does not wrok better than Glorot and Bengio's combined initialization.

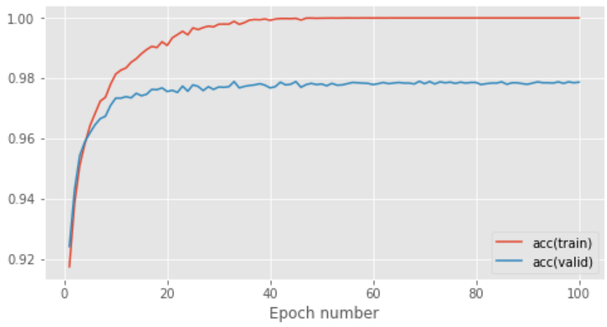

Figure 7. Accuracy of ELU, the last accuracy of validation set is 9.79e-01

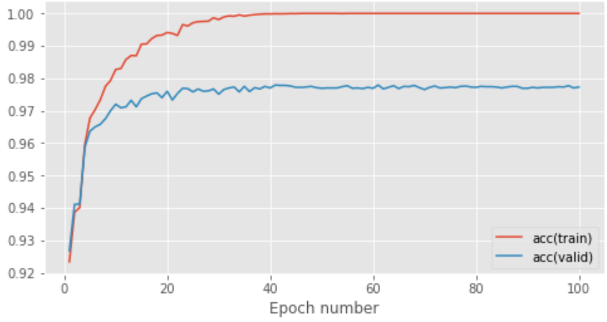

Figure 8. Accuracy of SELU, the last accuracy of validation set is 9.77e-01

5. Conclusions

In these experiments, the network with Fan-out initialization, two layers, Leaky ReLU layer marks the best performance. However, this result is against the recent study. This time, the data set is simple, and networks are extremely deep. The SELU functions and initialization come into their own when predicting more complex data with deeper network.

Depth	Initializer	Min of Valid Error	Last Valid Error	Valid Accuracy
2	GB	8.38e-02	1.09e-01	9.77e-01
2	Gaussian	9.36e-02	1.25e-01	9.76e-01

Table 3. Relationship between initialisation and performance with SELU layer

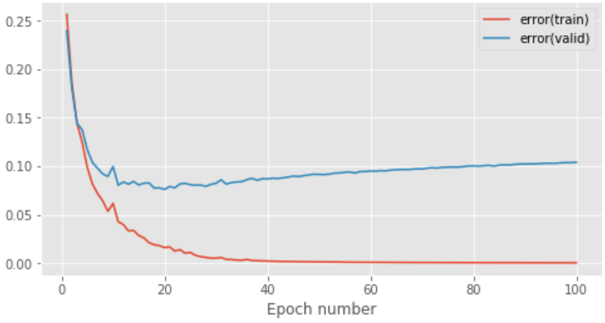

Figure 9. Accuracy or of 2 layers, Leaky Relu, Fan-out initialization

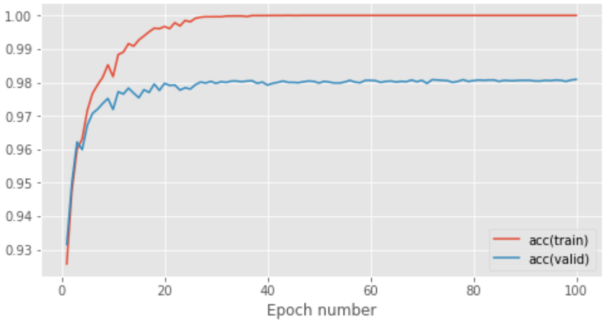

Figure 10. Error of 2 layers, Leaky Relu, Fan-out initialization