MLP Coursework 1: Activation Functions

s1717961

Abstract

Deeper networks have been applied in a broad range of industries Nowadays so that optimizer strategies are increasingly popular. The main objective of this paper is trying to understand potential influence with respect to some hyper-parameters' settings and initialisation schemes. All these experimental results were obtained with training multi-layer networks in terms of MNIST digits dataset. The impact of different activation functions on hidden layers, depth of multi-layers networks and weights initialisers are regarded as major research problems. In the end, related analysis is involved in each sections including corresponding graphs.

1. Introduction

This report aims to study the performance of several variants of the ReLU activation function for hidden units in multi-layer networks though mainly three aspects, error value of training and validation set respectively and validation set accuracies received from the trained models.

Three research questions relating with train scheme and initialisation will be addressed through training multi-layer networks as to the MNIST digit classification problem. Firstly, it will compare the performance of three distinct activation functions Leaky ReLU, ELU and SELU that have been used to obtain values of hidden layers in each multiple layers model. In particular, networks created by sigmoid and ReLU activation functions are treated as baseline systems in comparison with other architectures built by other three functions. Secondly, this paper will discuss how the depth of a network (number of hidden layers the network contains) affect accuracy of this model so that one of the above three activation functions, Leaky ReLU, was chose to investigate possible impact due to different numbers of hidden layers from two to eight. Additionally, in latter experiments, the influence of three initialization procedures with respect to weights, Glorot and BengioâĂŹs uniform weights initialisation used in most experiments, uniform distribution constrained the estimated variance of a unit to be independent of the number of incoming connections and the number of outgoing connections respectively, and Gaussian distribution, are expected to be compared using the SELU hidden units.

Plus, in order to standardize the network architecture, the batch size is set to 50, 100 epochs are trained and 100 hidden units are included for per hidden layer in all experiments. The MNIST digits dataset has 50,000 training images and 10,000 validation images each showing a 28ÃŮ28 grey-scale pixel image of one of the 10 digits. In following experiments, the size of the training and validation sets remain the same for the sake of simple and effective.

2. Activation functions

In general, four activation functions, ReLU, Leaky ReLU, ELU and SELU will be involved in this paper and latter experiments will be conducted for each of them using 2 hidden units, with 100 units per hidden layer. ReLU (Restricted Linear Unit):

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

which has the gradient:

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

However, One obvious weakness to rectiïňĄed linear units is that as for cases for which their activation is zero, they lose learning ability via gradient-based methods because a rectifying linear unit is zero for half of its inputs. A range of generalizations of rectiïňĄed linear units ensure that they can receive gradient(Nair & Hinton., 2010),for instance, a generalization of ReLU, Leaky Restricted Linear Unit, is involved in using using a non zero slope Îś(Goodfellow-et-al, 2016). In this report, alpha Îś is set to o.o1 in ReLU function and 1 in ELU respectively. Leaky ReLU (Leaky Restricted Linear Unit)(Andrew L Maas & Ng, 2013):

$$\text{lrelu}(x) = \begin{cases} alpha * x & \text{if } x \leq 0 \\ x & \text{if } x > 0, \end{cases} \tag{3}$$

which has the gradient:

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

ELU (Exponential Linear Unit)(Djork-ArnÃl' Clevert & Hochreiter, 2015):

$$\text{elu}(x) = \begin{cases} alpha * (exp(x) - 1) & \text{if } x \leq 0 \\ x & \text{if } x > 0, \end{cases} \tag{5}$$

which has the gradient:

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

Regarding SELU, experiments in this report adopts theoretical argument for optimal values of the two parameters: alpha Îś âĽĹ1.6733 and lambda Îż âĽĹ1.0507 SELU (Scaled Exponential Linear Unit)(GÃijnter Klambauer & Hochreiter., 2017):

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

which has the gradient:

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

3. Experimental comparison of activation functions

In addition, various experiments varied the type of nonlinear activation function in the 2 hidden layers networks with 100 hidden units per layer have been implemented with respect to these five activation functions mentioned so as to compare different performance through learning curves in terms of error and accuracy displayed in figure1 and figure 2 respectively. Also, all hyper-parameter settings such as learning rate, training and validation sets are controlled to be the same in all experiments. It can be seen from these two figures that these five functions have similar shapes in accuracy and error aspects. But it is interesting to note that learning curves of other four functions approach stable level quicker except that of Sigmoid architecture, which indicates higher rate of convergence in comparison with those of Sigmoid hidden units though architecture training performance seems not to be particularly sensitive to different activation functions among IReLU, ELU and SELU.

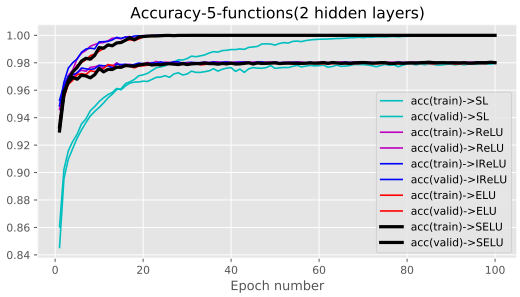

Figure 1. Accuracy figure with regards to five functions with two hidden layers

At the same time, table 1 shows final error and accuracy values for the training set and validation set as well as run time per epoch. It is clear that the run time of baseline system with ReLU hidden layers is the lowest and the another baseline network using Sigmoid layers comes the second, which obtains a small efficiency gain. Also ReLU baseline

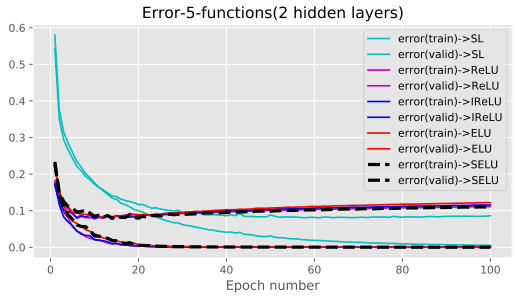

Figure 2. Accuracy figure with regards to five functions with two hidden layers

system achieves the least final error value of training set while Sigmoid network gets the highest training set error. Unlike Sigmoid, ReLU has no positive saturation (Goodfellow et al., 2016). Despite this, Sigmoid function achieves the best validation set error at 8.50e-2 and IReLU function comes second in the table.

Final Results	Sigmoid	Relu	LRelu	ELU	SELU
Error(train)	5.48e-03	1.47e-04	1.48e-04	2.06e-04	2.03e-04
Error(valid)	8.50e-02	1.17e-01	1.12e-01	1.17e-01	1.31e-01
Accuracy(valid)	9.78e-01	9.80e-01	9.81e-01	9.80e-01	9.78e-01
Run Time/epoch	2.06	1.80	2.69	3.02	3.19

Table 1. Results for five activation functions

Another point should be focused on is that the highest accuracy on the basis of the specified validation set are achieved by model of IReLU. Although there is a slight difference between accuracy of the training set of among LReLU, ReLU and ELU networks with only 1e-03, tLReLU system's error value of validation set(1.12e-01) is 5e-03 lower than that of ReLU. In particular, IReLU will have more transparent advantage over ReLU, if ReLU dies in experiments so that it responds zero to everything(Goodfellow et al., 2016). In this case, both ELU and SELU fail to show unique strength in the light of four criteria in table1 compared with ReLU and Sigmoid.

4. Deep neural network experiments

4.1. behaviour of deeper networks

Leaky ReLU activation function is chosen to investigate the impact of distinct depths of networks with 2 to 8 hidden layers,with 100 hidden units per layer when other parameters remain unchanged so as to investigate the performance of deeper networks. According to table 2, the highest accuracy of validation set is obtained at 6 hidden layers just 1e-03 more than that of 7 hidden layers. Although the lowest error figures of training set is in model of 7 hidden layers, the network of 2 hidden layers gains the least error value of

N	Err(train)	Err(valid)	Acc(valid)	Run Time/epoch
2	1.45e-04	1.24e-01	9.80e-01	2.39
3	3.82e-05	1.36e-01	9.80e-01	3.26
4	1.34e-05	1.37e-01	9.80e-01	4.52
5	7.08e-06	1.65e-01	9.80e-01	4.41
6	4.24e-06	1.51e-01	9.83e-01	5.59
7	2.60e-06	1.70e-01	9.82e-01	5.80
8	4.29e-06	1.84e-01	9.80e-01	7.46

Table 2. Results for depth of networks

validation set. Accuracy of validation set reaches a plateau before rising to 9.83e-01 for model of 6 hidden layers and decreasing marginally in the end. Obviously, validation set error rises gradually as the number of hidden layers increases. There is an a downward trend in terms of error of training set with the depth of networks rising though there is a sudden drop regarding model of 7 hidden layers because the total number of free parameters in the model growths, as the number of layers increase controlling a fixed hidden layer width. Hence the model is expected to be capable of fitting the training data better. The validation set error begins to increase when the number of hidden layers is 2 even as the training set error continues to decrease. Generally, validation set error should begin to decrease at first, then rise because of over-fitting(Bishop, 2014), which can be verified from the error trend of validation set in figure 3. This indicates that these models have begun over-fitting to the training data. We could get a better validation set error in these cases by stopping the training early.

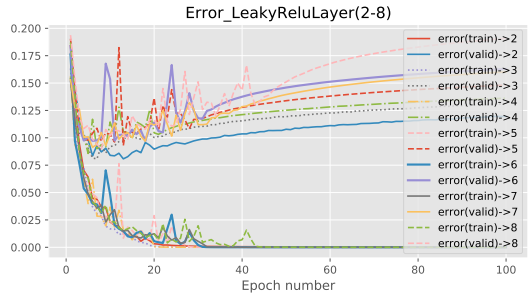

Figure 3. error figure with regards to IReLU functions

4.2. weights initialiser

Normally, weights should be initialised into a small random number and sampled independently from two distribution of Gaussian and Uniform. In experiments of this section, mean is set to0 typically and variance will be different according to five initialisers. One of five activation functions, SELU activation function, is used to study influence of different weights initialisation strategies when other hyperparameters are set to be the same and networks of two hidden layers with 100 hidden units per layer are selected in experiments. Five initialisation methods are picked, including the default Glorot and Bengio (2010) random uniform weights initialiser (GlorotUniform) in previous experiments. The default initialiser initialises weights on the basis of a zero-mean uniform distribution with standard deviation ($\sqrt{2/input - dimension + output - dimension}$): square root of 2 divides the product of input dimension and output dimension of weight matrix , which corresponds to the weight initialisation based on Fan-in and Fan-out that chooses the square root of six divides the sum of input dimension and output dimension as standard deviation.

The following section will introduce the xavier algorithm that automatically determines the scale of initialization in the light of the number of input and output neurons in order to filter weights. Fan-in:

$$weight \sim U(-\sqrt{3/input - dimen}, \sqrt{3/input - dimen}).$$

Fan-out:

$$weight \sim U(-\sqrt{3/output - dimen}, \sqrt{3/output - dimen}).$$

Fan-in and Fan-out:

$$weight \sim U(-\sqrt{6/(input - dimen + output - dimen)}, \sqrt{6/input - dimen + output - dimen})). \quad (9)$$

In addition, Gaussian distribution is selected as the fifth initialiser and explained as follows: Gaussian:

$$weight \sim \mathcal{N}(\mu, \sigma^2).$$

In this case, $\mu = 0$ and $\sigma^2 = i/inputdimension$. Plus, the dimension of inputs(784) and output dimension 10 will change to uniform hidden dimension(100) when weights are computed in the light of above algorithms since model of two hidden layers is applied in related experiments.

Final Results	GlorotU	Fan-in	Fan-out	Fan-in/out	Gaussian
Err(train)	2.03e-04	2.00e-04	2.03e-04	2.11e-04	2.03e-04
Err(valid)	1.31e-01	1.28e-01	1.18e-01	1.23e-01	1.14e-01
Acc(valid)	9.78e-01	9.80e-01	9.77e-01	9.78e-01	9.79e-01
Time/epoch	3.19	3.37	3.13	3.05	3.51

Table 3. Results for weights initialisations

As we can see from table 3, Weight initialisation using Fan-in get the largest validation set accuracy at 9.80e-01 with only 1e-03 better than that of Gaussian distribution. Also, the lowest validation set error is obtained by SELU layer initialising by Gaussian distribution while model using GlorotUniform gets the worse error value of validation set even though this weight initialisation Fan-in method reaches the best training set error. Five experiments were carried out in order and model of Fan-in and Fan-out were run the most quick. Thus, it seems like weight initialiser using Gaussian distribution could fulfil better performance with regard to networks consist of two SELU hidden layers in comparison with other initialisers.

4.3. depth of architecture and weights initialiser

In order to get comparable results between depth of network and weights initialiser, further experiments have been conducted and the following line graph show that final results on a set of number of SELU hidden layers from 2 to 8 with respect to three xavier algorithms, Fan-in, Fan-out and Fan-in and Fan-out. Overflows are likely to happen because of high learning rate setting (0.1) in previous experiments. In this experiment, learning rate is changed to 0.001 with 100 times lower than previous setting, which helps to avoid overflow.

According to figure 4, obviously, The best training set errors are all obtained from Fan-out method from networks with 2 hidden layers to 8 hidden layers, which is continuous for validation set error until model of 6 hidden layers, the latter best error values are achieved by Fan-in and Fan-out method. Apart from this, the best validation set accuracies appear in Fan-in and Fan-out method especially for deeper networks from 4 hidden layers rather than Fan-out even though the other models' best accuracies are fulfilled by Fan-out strategy. By contrast, Fan-in algorithm does not play an important role compared with previous two architectures.

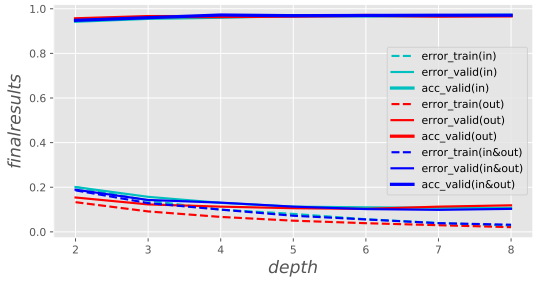

Figure 4. figure of Uniform Distribution

Table 4 consists of three criteria that collected from the above line figure corresponding to different depths of network. As can be seen from this table, the best validation accuracies increase as the depth of networks increase then level off after 6 hidden layers. Apparently, the best error of training set decrease steadily when the depth rises. Such experimental results show the superiority of deeper vs less deep networks. Although the best validation set error see a similar downward trend, it starts to increase after depth of 7 hidden layers from 9.86e-02 to 1.03e-01. It indicates that this network begins to over-fitting and the most appropriate depth for this model given experimental conditions is 7 hidden layers.

5. Conclusions

In conclusion, the deeper architecture tends to achieve better performance on the basis of validation set error and accuracy. However, over-fitting is more likely to appear as the depth of network increases, which is obtained from experiments in section 4.1 and 4.3. Early stopping is one decent way in trying to overcome this problem. At the same time, weights initialisation methods play a large part in affecting the behaviour of learning curves. Also, the models with more layers is expected to take longer to train per epoch. Therefore, on top of cases of potential over-fitting and difficulty of choosing weights initialisers, computational time is needed to be taken into account.

N	Best Err(train)	Best Err(valid)	Best Acc(valid)
2	1.33e-01	1.54e-01	9.55e-01
3	9.16e-02	1.23e-01	9.65e-01
4	6.69e-02	1.13e-01	9.72e-01
5	5.01e-02	1.06e-01	9.69e-01
6	3.88e-02	1.02e-01	9.71e-01
7	2.98e-02	9.86e-02	9.71e-01
8	2.10e-02	1.03e-01	9.71e-01

Table 4. Results for depth of networks

1>architectures built by LReLU hidden units achieved the best performance regarding validation set accuracy in comparison with baseline systems and other networks of ELU and SELU though the advantage is not very obvious in networks of two hidden layers specified in this paper.

2>networks of only two hidden layers get the lowest validation set error though network of six IReLU hidden layers perform excellent for accuracies in validation set. Also, over-fitting can be verified from the opposite trend in training set error and validation set error in table 4 and line trend in figure 3.

3>regarding the impact of weights initialisers, Gaussian method appears to be the best choice with regards to networks of two SELU hidden layers compared with uniform distribution strategies, if validation set error and accuracy are taken into account. In addition, Fan-in appears to take a better role in deeper networks in comparison with other two similar initialisation methods.

As a whole, experiments conducted in this paper are not enough sufficient to approve some outcomes since there are limited number of experiments and ideal experimental conditions. But some theoretical knowledge can be figured out clearer by the means of such experiment procedures. More works still are expected to be carried out in further experiments.

References

Andrew L Maas, Awni Y Hannun and Ng, Andrew Y. Rctifier nonlinearities improve neural network acoustic models. In Proc ICML, 2013.

Bishop, Christopher. *Machine Learning and Pattern Recognition*. Springer-Verlag New York, 2014.

Djork-ArnÃl' Clevert, Thomas Unterthiner and Hochreiter, Sepp. ast and accurate deep network learning by exponential linear units (elus). 2015.

Goodfellow, Ian, Bengio, Yoshua, and Courville, Aaron. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

GÃijnter Klambauer, Thomas Unterthiner, Andreas Mayr and Hochreiter., Sepp. Self-normalizing neural networks. 2017.

Nair, Vinod and Hinton., Geoffrey E. rectified linear units improve restricted boltzmann machines. pp. 807âĂŞ814. In Proc ICML, 2010.