# Activation Function: Absolute Function,One Function Behaves more Individualized

**Abstract:** Inspire by nature world mode, a activation function is proposed. It is absolute function.Through test on mnist dataset and fully-connected neural network and convolutional neural network, some conclusions are put forward. The line of accuracy of absolute function is a little shaken that is different from the line of accuracy of relu and leaky relu. The absolute function can keep the negative parts as equal as the positive parts, so the individualization is more active than relu and leaky relu function. In order to generalization, the individualization is the reason of shake, the accuracy may be good in some set and may be worse in some set. The absolute function is less likely to be over-fitting. The batch size is small, the individualization is clear, vice versa. Through test on mnist and autoencoder, It is that the leaky relu function can do classification task well, while the absolute function can do generation task well. Because the classification task need more universality and generation task need more individualization. The pleasure irritation and painful irritation is not only the magnitude differences, but also the sign differences, so the negative part should keep as a part. Stimulation which happens frequently is low value, it is showed around zero in figure 1 . Stimulation which happens accidentally is high value, it is showed far away from zero in figure 1. So the high value is the big stimulation, which is individualization.

**Keyword:** activation function, absolute function, individualization, universality, over-fitting, abstract network, concrete network, stimulation

## 1.1. Preface

There are many activation function, such as sigmoid, tanh, relu, leaky relu, elu. The function frequently used is relu[1]. The relu function has dead relu problem that is some neuron never be activated, the related parameter never be updated. The absolute function doesn' t have dead relu problem, and gradient disappearance and gradient explosion, however it has some interesting characteristics.

The image of absolute function is showed below. The formula is y=|x|.

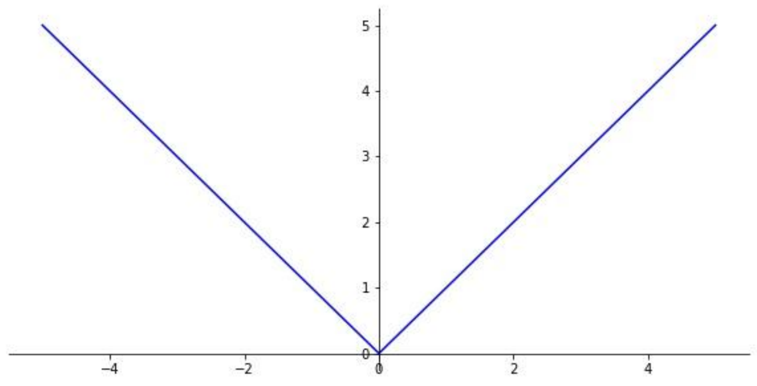

Fig1:The Image of Absolute Function

## 2.1. The test of absolute function on fully-connected neural network

First, I test the activation function which include relu , leaky relu and absolute on mnist dataset and fully-connected neural network. The network is 5 layers. The first 5000 samples are validation set, the left are training set[7]. Optimizer is adam[6], loss is SparseCategoricalCrossentropy, metric is accuracy[2]. The following picture is the test result.

From the figure 2, the accuracy of absolute function in validation set is lower than the accuracy of relu and leaky relu in validation set when epochs are small. But when the epochs grow, the accuracy of absolute function in validation set is almost equal with the accuracy of relu and leaky relu in validation set. The line of accuracy of absolute function is a little shaken that is different from the line of accuracy of relu and leaky relu. The loss of absolution function is almost equal with the loss of relu and leaky relu. Because the absolute function keep the negative parts as equal as the positive parts, so the individualization is more active than relu and leaky relu function. Why? Let’s change the batch size, the results are in figure 3.

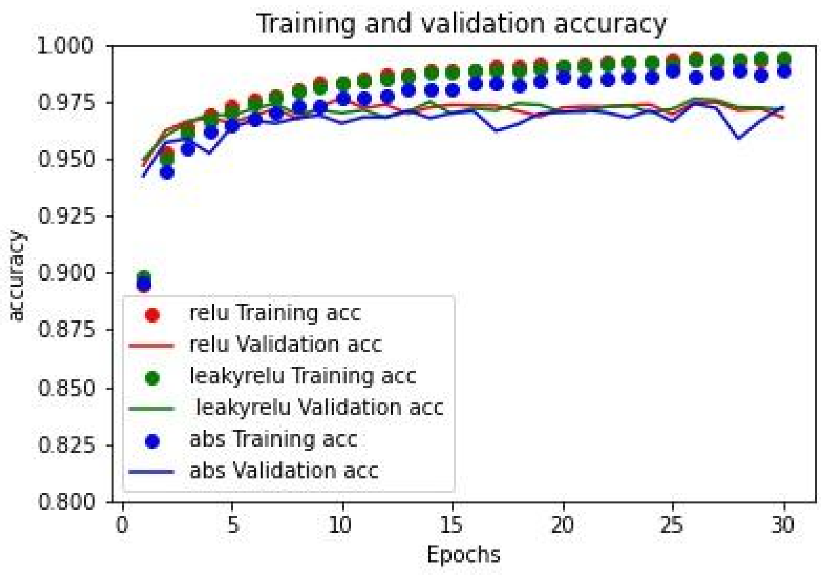

Fig2.1:Training and Validation Accuracy,Batch size is 32

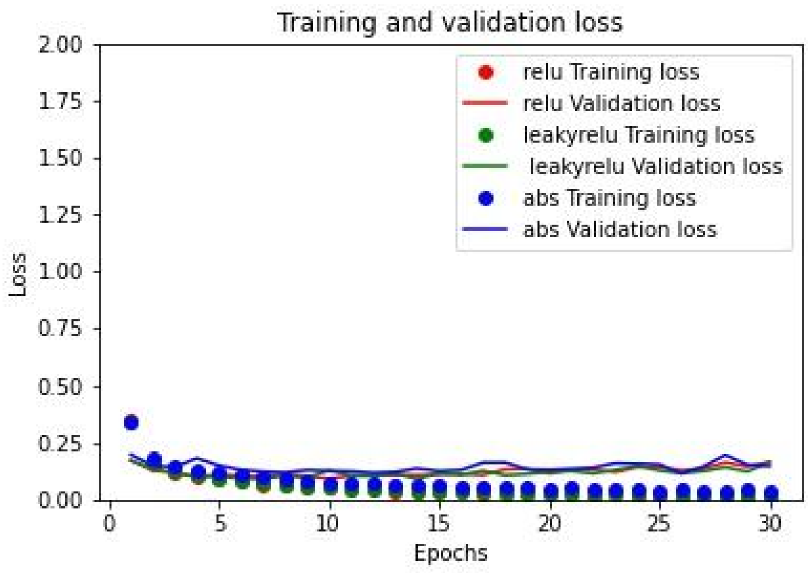

Fig2.2:Training and Validation Loss,Batch Size is 32

In figure 3, batch size is 64, the line is no longer shaken like before, it behaves like the line of relu and leaky relu. The line of relu and leaky relu function are more universality and stable than the line of absolute function. It filters the negative parts, so it loses the individualization partly. In order to generalization, the individualization is the reason of shake, the accuracy may be good in some set and may be worse in some set. The absolute function is less likely to be over-fitting.

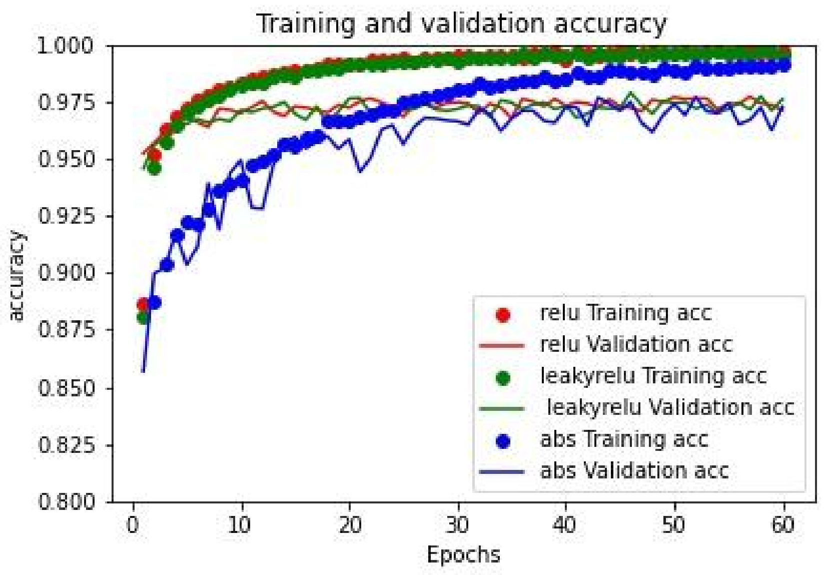

Fig3:Training and Validation Accuracy,Batch Size is 64

## 2.2. The test of absolute function on convolutional neural network

Let's change the neural network, the result on convolutional neural network and mnist dataset is showed in figure 4. The convolutional neural network has 3 layers convolutional neural network and 2 layers fully connected network. The shake phenomenon is more clear than on fully-connected neural network when batch size is 32. When batch_size increase, the shake phenomenon is more weaker. It can be anti-over-fitting when batch_size is small. The loss which decrease firstly but increase then is odd when batch size is 32, however the loss is stable like relu and leaky relu when batch size is 128. Correspondingly, the accuracy is increase step by step. In order to increase the accuracy, the model has to learn the individualization which increases the loss. The relu and leaky relu function can not learn the individualization like the absolute function, so its loss is stable. The batch size is small, the individualization is clear, vice versa. If you want to change the individualization of absolute function, just change the batch size.

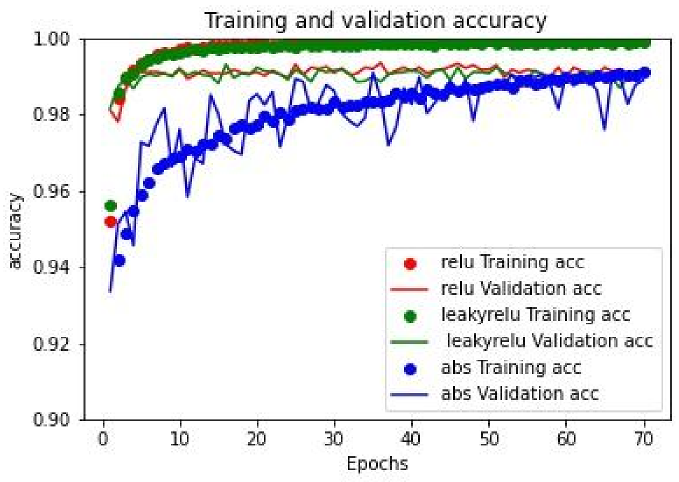

Fig4.1:Training and Validation Accuracy,Batch Size is 32

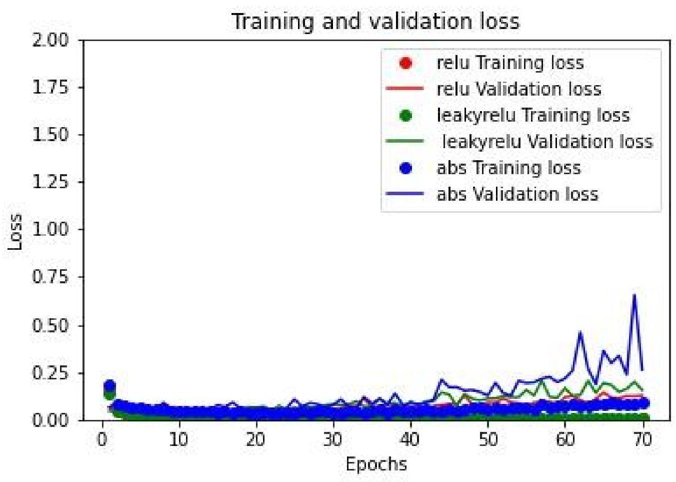

Fig4.2:Training and Validation Loss, Batch Size is 32

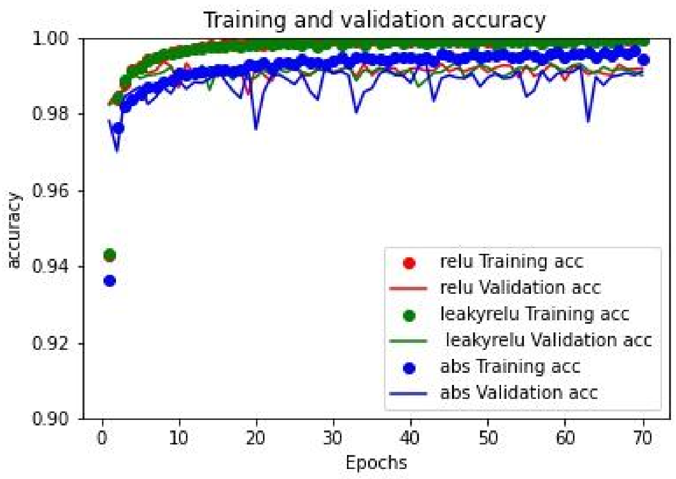

Fig4.3:Training and Validation Accuracy, Batch Size is 64

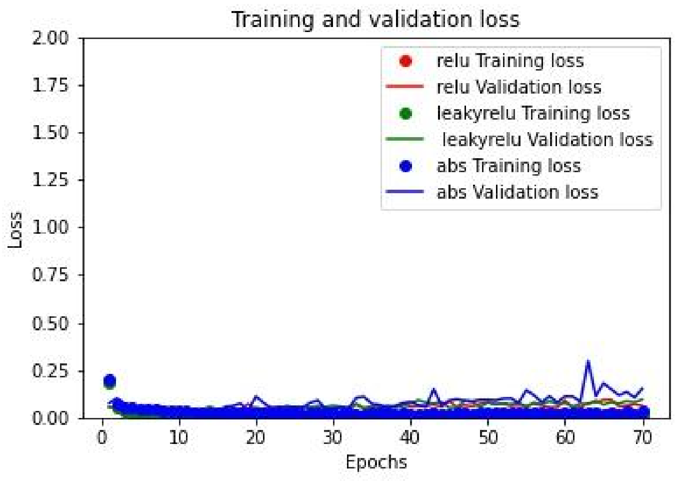

Fig4.4:Training and Validation Loss, Batch Size is 64

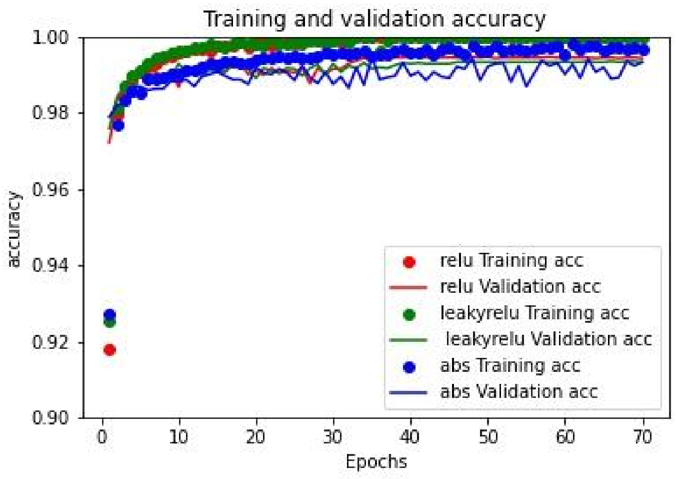

Fig4.5:Training and Validation Accuracy, Batch Size is 128

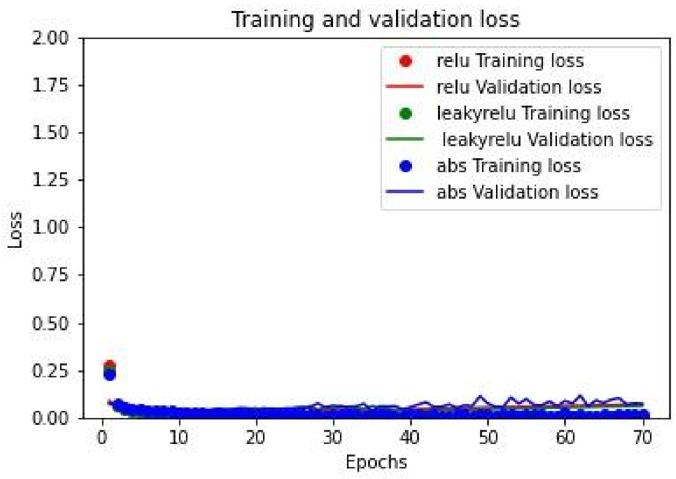

Fig4.6:Training and Validation Loss, Batch Size is 128

In order to visualize the intermediate activation, one more test is conducted. The convolutional neural network has 3 layers convolutional neural network and 2 layers fully connected network. The test result is showed in Fig4.7 and Fig 4.8. In Fig 4.7,the intermediate activation is sparse and blur with relu activation; Oppositely in Fig 4.8, the intermediate activation is dense and clear with absolute function. Because the relu activation function filter the negative part and the absolute activation function keep the negative part. Because the feature the absolute function shows is clear , so the generation work can be done by absolute function.

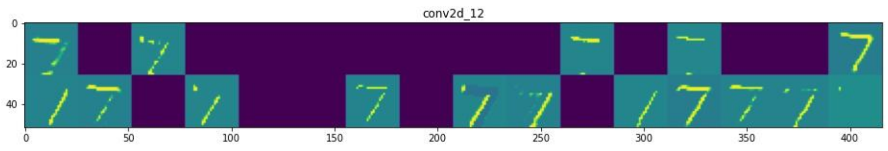

Fig4.7.1 Visualize Intermediate Activation with Relu Activation(First Convolution Layer)

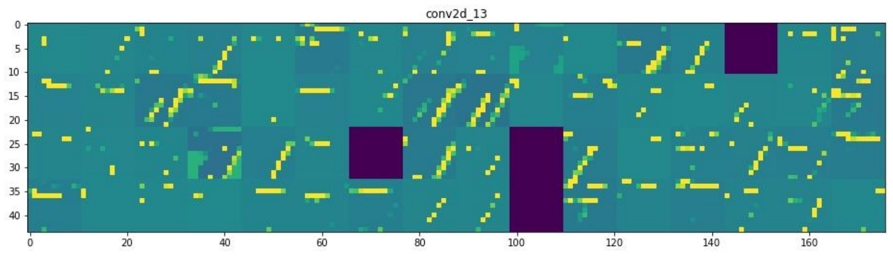

Fig4.7.2 Visualize Intermediate Activation with Relu activation (Second Convolution Layer)

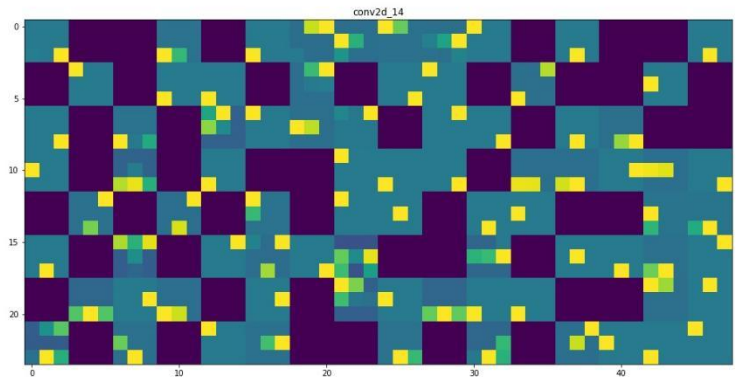

Fig4.7.3 Visualize Intermediate Activation with Relu activation (Third Convolution Layer)

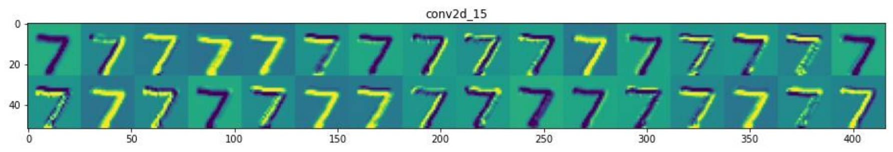

Fig4.8.1 Visualize Intermediate Activation with Absolute activation (First Convolution Layer)

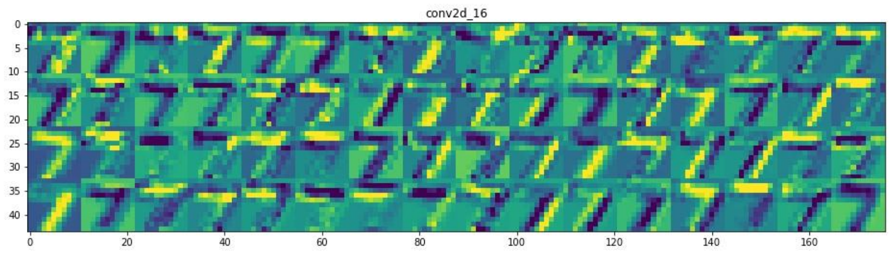

Fig4.8.2 Visualize Intermediate Activation with Absolute activation (Second Convolution Layer)

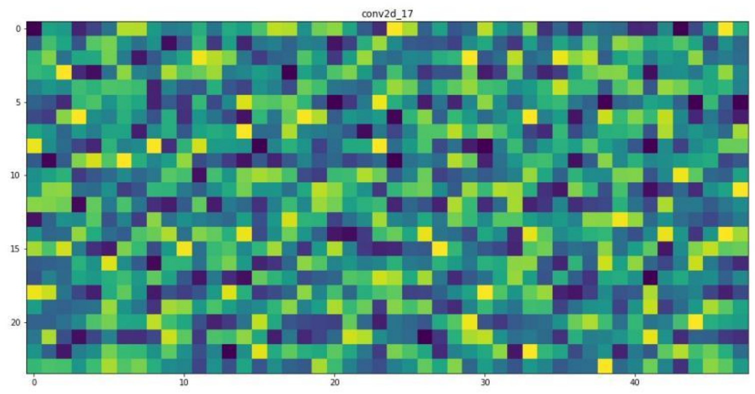

Fig4.8.3 Visualize Intermediate Activation with Absolute activation (Third Convolution Layer)

## 3. The test of absolute function on autoencoder generation

Because the absolute function has more individualization, so I choose it to do autoencoder's generation[3] work.

Abstract network(namely prediction network, always used as classification and regression) is common now. But the concrete network(namely generation network) which generates concrete information from concept or label is rare. Its principle is showed in figure 5. The test is on mnist dataset and convolutional neural network[5]. The convolutional neural network(abstract network) is like LeNet, it has 4 layers convolutional neural network and 2 layers fully connected network. The concrete network is the inverse function of abstract network, it has 6 layers. Optimizer is adam, loss is mse. You can read my paper[3] for the detail. The test result is showed in figure 6. In figure 6, the left image is input, the label is argmax function of the predict of output of abstract network, the right image is the output of concrete network.

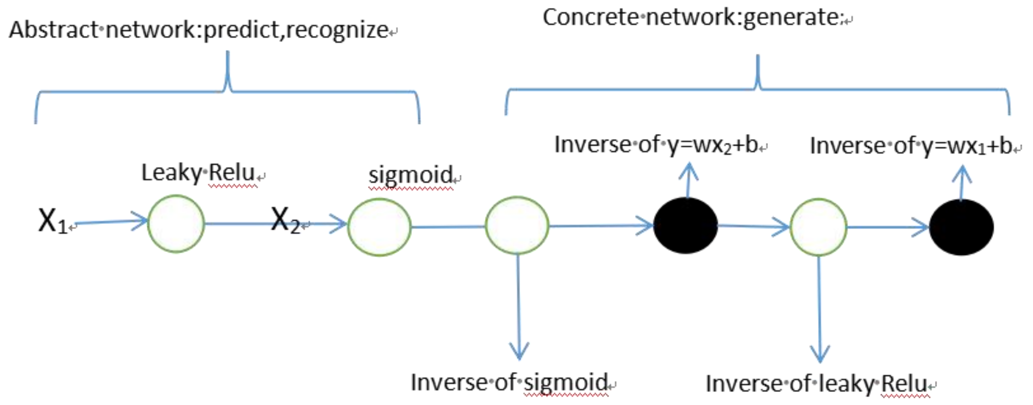

Fig5 The Functionally Separate Auto-encoder

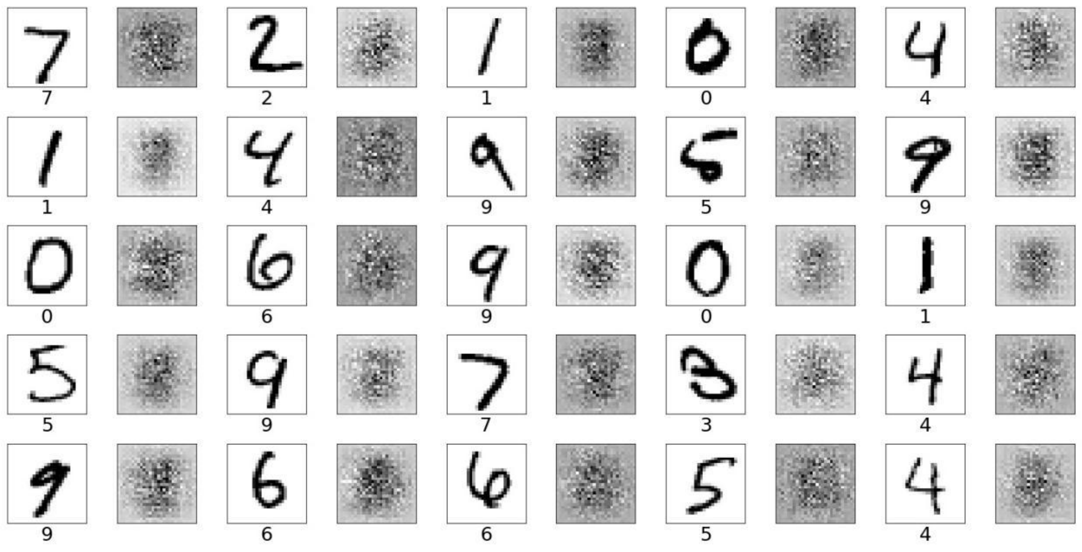

Fig6.1 Leaky Relu Function,Epoch=1

In figure 6.1, when epoch=1, the labels of input are all right. But the image of output is very blurred. Meanwhile, in figure 6.2, the labels of input are not all right. But the image of output is clear. One conclusion can easily draw. It is that the leaky relu function can do classification task well, while the absolute function can do generation task well. Because the classification task need more universality and generation task need more individualization. In figure 6.2 and 6.3, epoch=30, the result is almost the same between the two activation function. One different is the background color of absolute function is whiter than leaky relu function. So the absolute function can do generation task well.

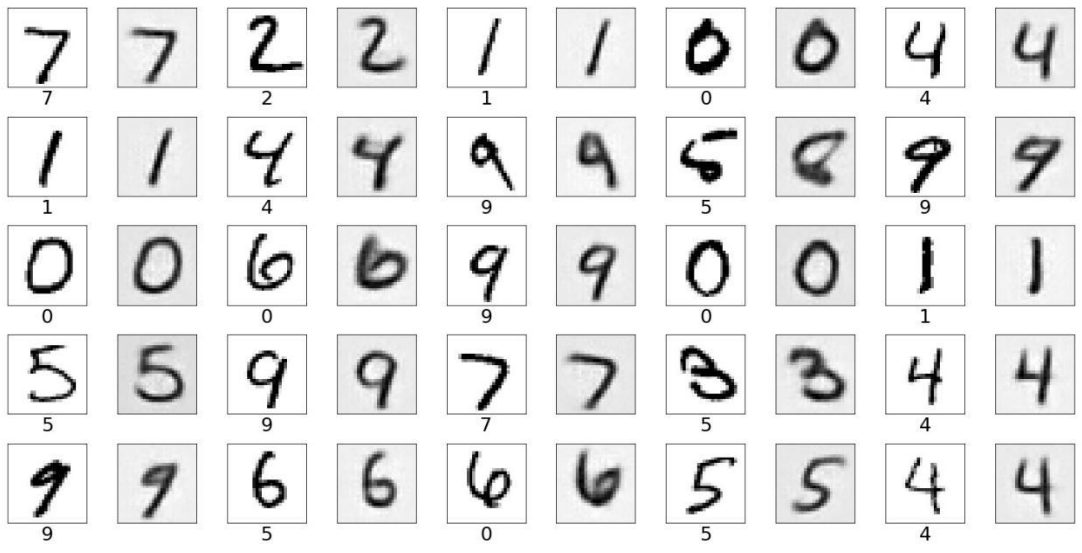

Fig6.2 Absolute Function,Epoch=1

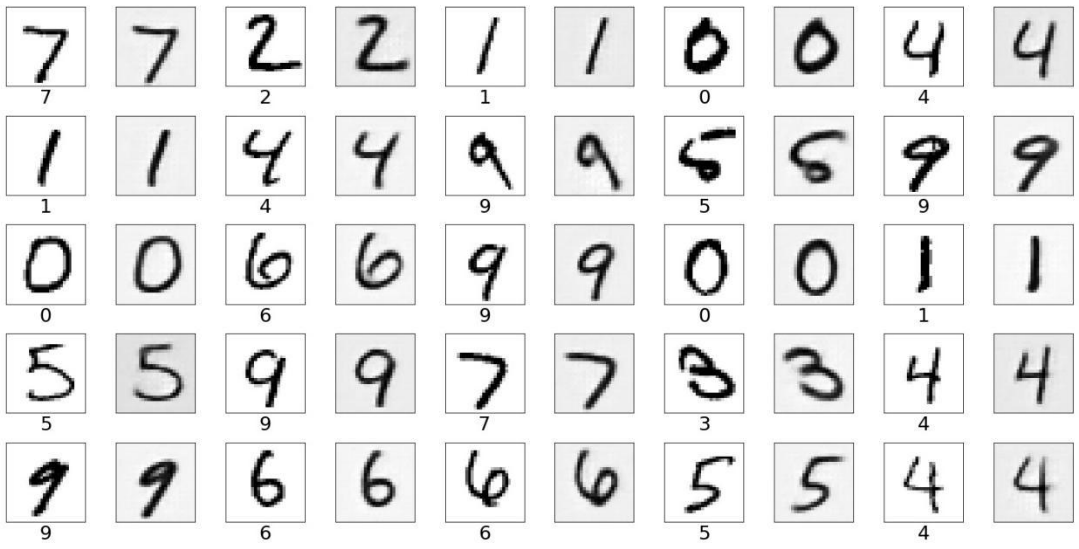

Fig6.3 Leaky Relu Function,Epoch=30

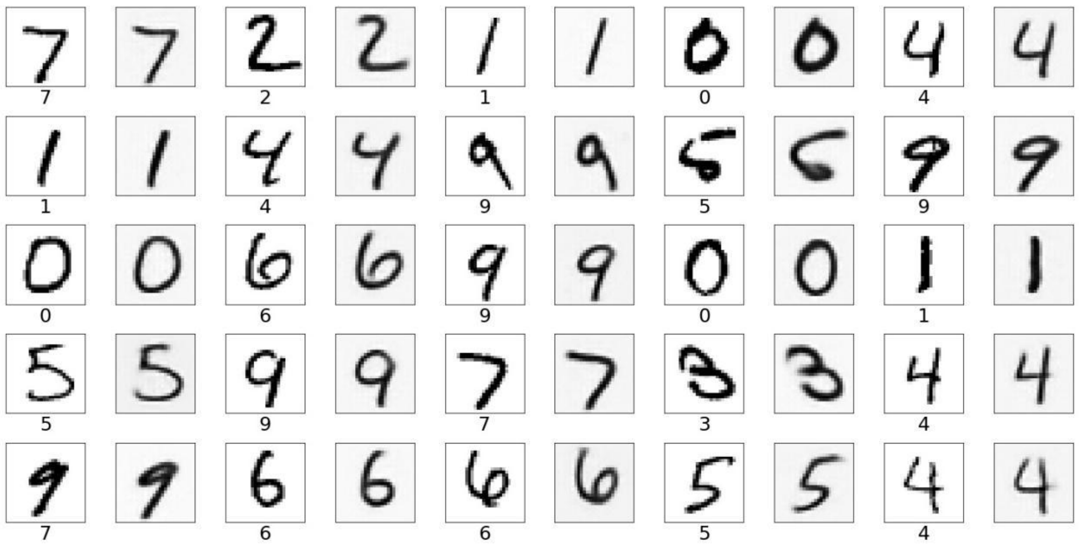

Fig6.4 Absolute Function,Epoch=30

## 4.Some Thinking

Should we keep the negative parts of the inputs? I think we can learn from other fields. The color has white and black, the temperature has hot and cool, the smell has fragrant and smelly, the taste has bitter and sweet. All information our body received are opposite mode. The pleasure irritation and painful irritation is not only the magnitude differences, but also the sign differences, so the negative parts should keep as a part. According to probability principle, nature word is normal distribution[4]. Stimulation which happens frequently is low value, it is showed around zero in figure 1 . Stimulation which happens accidentally is high value, it is showed far away from zero in figure 1. So the high value is the big stimulation, which is individualization.

## References

1. Ian Goodfellow, Yoshua Bengio, Aaron Courville, “Deep Learning”, MIT Press, 2016

2.Tensorflow Tutorials and Apis, https://tensorflow.google.cn/learn last accessed 2021/12/20

3..Jin-xin Wei, Qun-ying Ren, A Functionally Separate Autoencoder, Future Technology Conference, 2020/10

4.Sheldon M.Ross, “A First Course in Probability”,Pearson Prentice Hall, 2009/01/07

**5.**Aurélien Géron, ”Hands-On Machine Learning with Scikit-Learn and TensorFlow”, O'Reilly Media, 2017/01/15

6.François Chollet, ”Deep Learning with Python”, Manning Publications，2017/10

7.Tom Mitchell, “Machine Learning”, McGraw-Hill Science, 2008/03