Project Deep Learning II

Kevin XU
Qin WANG

10 May 2020

1 Introduction

The aim of this project is to build Deep Neural Networks pre-trained or not for the MNIST Handwritten Digit Classification task. We will compare the performances in terms of classification error rate of a pre-trained network and a randomly initialized network by varying the number of layers, the number of neurons and the number of training images.

2 Etude sur MNIST

For each experiment, we initialize first two identical networks. Then, we generatively pre-train one network as a deep belief network. Finally, we train both networks with the backpropagration algorithm separately and we compare their performances on the test set.

We set the learning rate to 0.1, the batch size to 64, the number of iterations in the Contrastive Divergence to 100 and the number of epochs in the backpropagation algorithm to 200. All the experiments were carried out on Google Colab.

2.1 Number of layers

First of all, we keep the number of neurons(200 neurons for each layer) and the number of training images(60000 training samples) unchanged, and set the number of layers to different values(See table 1).

Number of layers	**pre-training**	**no pre-training**
2	0.05249	0.09299
3	0.05449	0.09609
5	0.05420	0.18960
7	0.04949	0.88650

Table 1: Error rate according to number of layers

From table 1 and figure 1 we can see that, pre-trained network outperforms not pre-trained network for all the parameter settings. And when we increase the number of layers, the performance of pre-training network does not change too much but the performance of not pre-trained network degrades. As for not pre-trained network, when the accuracy is ideal with a small number of layers, we can not improve the performance significantly by increasing the number of layers. On the contrary, this will make it more difficult for training and cause vanishing gradient problem. Thus, the pre-training benefits when the we have many layers.

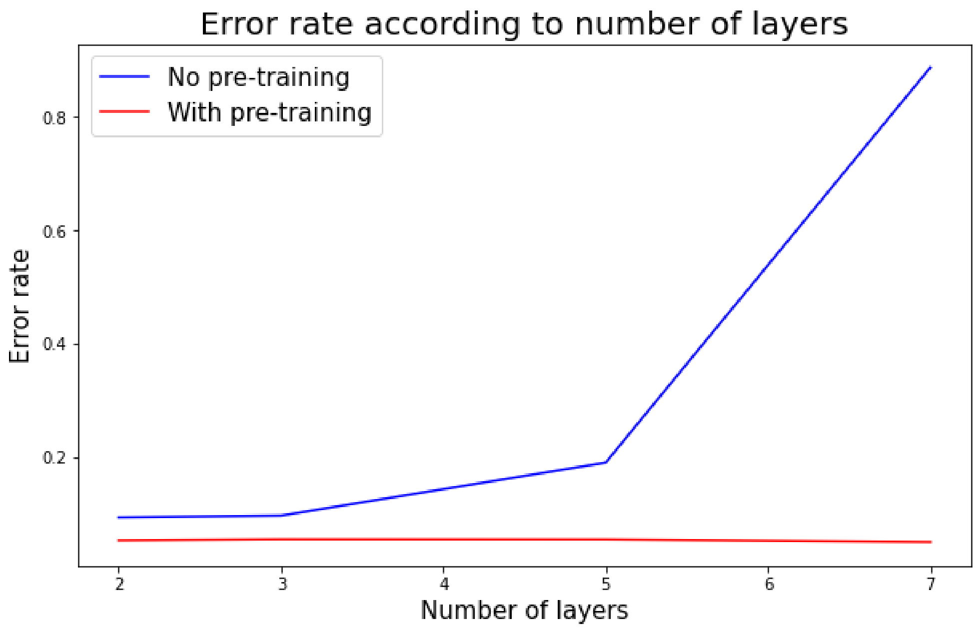

Figure 1: Error rates and number of layers

2.2 Number of neurons

In the next step, we keep the number of layers(2 layers) and the number of training images(60000 training samples) unchanged, and set the number of neurons to different values(See table 2).

Number of neurons	pre-training	no pre-training
100	0.07040	0.09509
300	0.05700	0.09130
500	0.05000	0.09880
700	0.04579	0.10019
1000	0.04849	0.09680

Table 2: Error rate according to number of neurons

From table 2 and figure 2, we can see that the error rate of pre-trained network is lower compared to the error of the network without pre-training. And we can conclude that increasing number of neurons does not increase accuracy endlessly. When we augment the number of neurons from 100 to 300, both networks improved their performance. But when we continue to increase the number, the effect is difficult to evaluate. For pre-trained network, the best number of neurons is 700, and for not pre-trained network is 300.

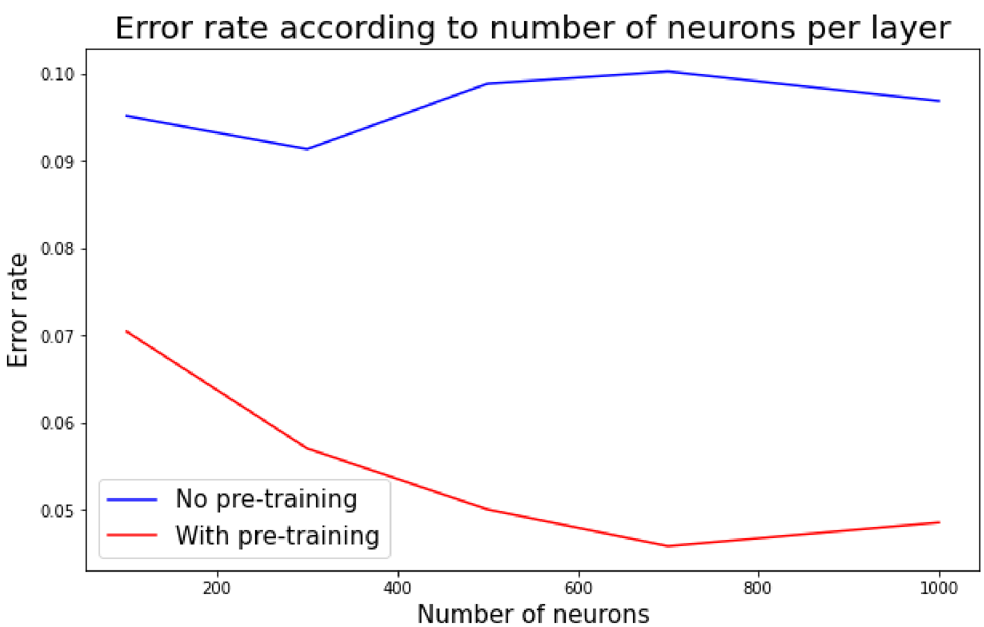

Figure 2: Error rates and number of neurons

2.3 Number of training images

Finally, we keep the number of layers(2 layers) and the number of neurons(200 neurons for each layer) unchanged, and set the number of neurons to different values(See table 3).

Number of training images	pre-training	no pre-training
1000	0.10319	0.14670
3000	0.06940	0.10660
7000	0.06120	0.09170
10000	0.05710	0.08899
30000	0.05630	0.09270
60000	0.05259	0.09099

Table 3: Error rate according to number of training images

From table 3 and figure 3, we can see that increasing the number of training images can lead to a better performance. But the effect is not obvious when accuracy is high. And a large data set make the computation more expensive. For both architectures, it seems like the error rates do not decrease anymore from a certain number of training samples.

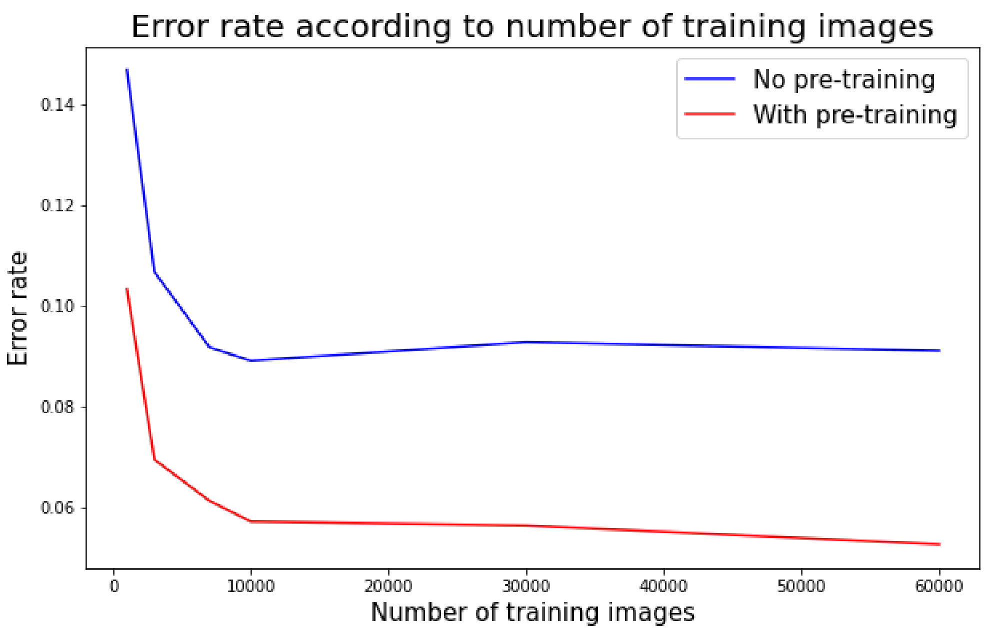

Figure 3: Error rates and number of samples

3 Conclusion

In conclusion, we saw that the performance of neural networks with a generative pre-training process are improved in most cases. However, the pre-training can require a significant amount of time, especially when the network is complex.

That's why, even though the performances are improved, we have to choose a trade-off between running time and accuracy.

For the classification task of MNIST data, the best configuration is a pre-trained network with 7 layers, 500 neurons in each layer and using about 20000 images for training.