Comparison of Deep Neural Network and Convolutional Neural Network for MNIST classification

Jerry

March 17, 2018

Abstract

With the help of Tensorflow, a deep neural network (DNN) and a convolutional neural network (CNN) were constructed for the hand-written digits classification problem. This report will introduce the architecture of these neural networks (NN) and implement them.

1 Architecture of DNN and CNN

In order to recognize the hand-written digits, we can take advantages of NN. NN is a standard learning algorithm and basically consists of three parts - INPUT LAYER, HIDDEN LAYER and OUTPUT LAYER. Each layer is made up of neurons (nodes) and is either fully (densely) or partially connected with the adjacent layers.

As for the hand-written digits problem, input is the image and output is the corresponding digit. The number of hidden layers decides whether the NN is deep or not. Normally, if the NN has more than 2 hidden layers, we name it DNN. In this report, a DNN with 3 hidden layers will be used. In terms of DNN, each layer is fully connected with the adjacent layers and this may hugely increase the number of parameters, especially when the size of input is big. It is not easy for the computer to learn and may lead to an overfitting issue.

To tackle this problem, CNN was designed and each layer of CNN is partially connected by a filter (kernel). The filter can scan the images in a three-dimensional space and extract some important features. Additionally, we can adopt pooling layers which help us reduce the parameters. In the final step, we normally need fully connected (FC) layers to realize classification. To summarize, the structures of DNN and CNN are:

DNN: input+n(hidden layer)+output

CNN: input+n(conv layer+pooling layer)+n(FC layer)+output

For both DNN and CNN, we MUST choose an activation function to add nonlinearity. Otherwise, a 100 hidden layers DNN performs the same as a 1 hidden layer NN, because they are linear. In this report, we select Relu.

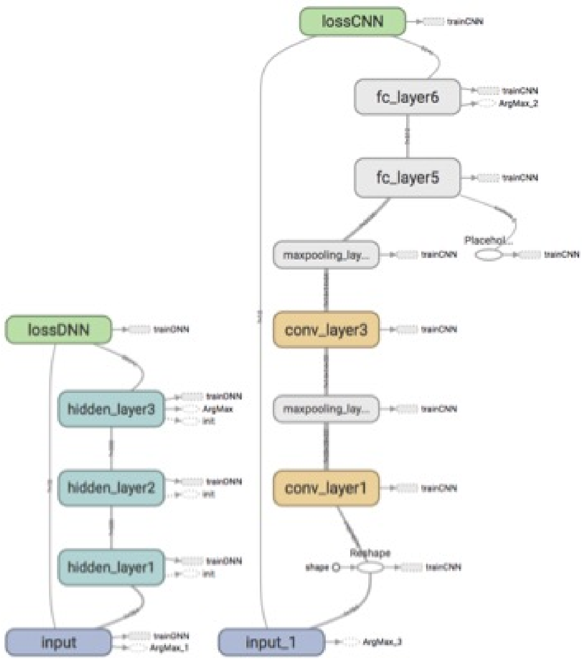

Figure 1: DNN (left) and CNN (right) architecture.

DNN	input	hidden layer1	hidden layer2	hidden layer3	output
	1x784	1x512	1x256	1x128	1x10

Table 1: DNN structure.

2 Tensorflow implementation

2.1 DNN and CNN setups

With regard to the Tensorflow implementation, the models are visualized in the Fig. 1. In DNN, we use three hidden layers and use the Relu as activation function. For CNN, the total number of layers is 6 and we use 2 convolutional layers combined with max pooling layers. Also, two FC layers are followed to realize classification. The sizes of each layer of DNN and CNN are listed in Tab. 1 and Tab. 2, respectively. In CNN, the first filter has a size 5x5 and its output has a depth 5. It shifts one unit every time and adopt zero padding so as to keep the size of image. As for the max pooling, it also uses zero padding and the size of image is halved by shifting two units at a time.

CNN	input	conv1	maxpool1	conv2	maxpool2	FC1	FC2	output
	28x28x1	5x5x32	2x2	5x5x64	2x2	1x3136	1x512	1x10

Table 2: CNN structure.

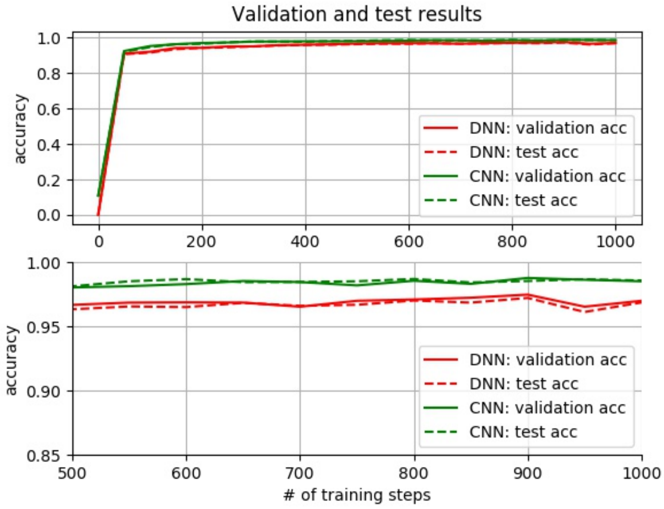

Figure 2: Validation and testing accuracies within 1000 rounds.

2.2 Cost function and training

Cross entropy is used for building the cost function and Adam is our optimizer. Since we know each image corresponds to one digit, we pick up the highest probability to form the cost function. Both the batch size (100) and learning rate (0.001) were selected by trial and error. The total number of training steps is 1000. To evaluate the learning performance, we use accuracy which is defined as:

$$acc = \frac{numCorrect}{numTotal} \tag{1}$$

2.3 Results and analysis

The MNIST dataset provides us training (55000), validation (5000) and testing (10000) dataset. We use the training dataset to train the NN and examine both the validation and testing accuracy every 50 iterations. For our case, I did not use the validation dataset to do early stopping. The revolutions of validation and testing accuracies are drawn in Fig. 2. It can be clearly seen that CNN (in green) outperforms DNN (in red). The final testing accuracy of CNN is 99.24%, which is higher than that of DNN (97.11%). In our implementation, the total number of parameters of DNN and CNN are 567434 and 1663370, respectively. Therefore, we applied the Dropout strategy for CNN so as to reduce the parameters. However, CNN (348s) is still more time consuming than DNN (12s) with 1000 training steps.

3 Conclusion

Overall, DNN and CNN were successfully applied to classification. According to the results, CNN did a better job.