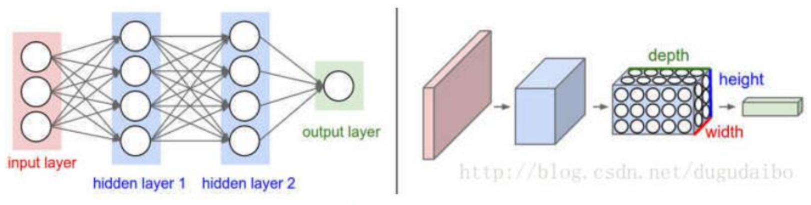
Q3 Handwritten Digit Classification

Author: Jiyao Wang 20797324

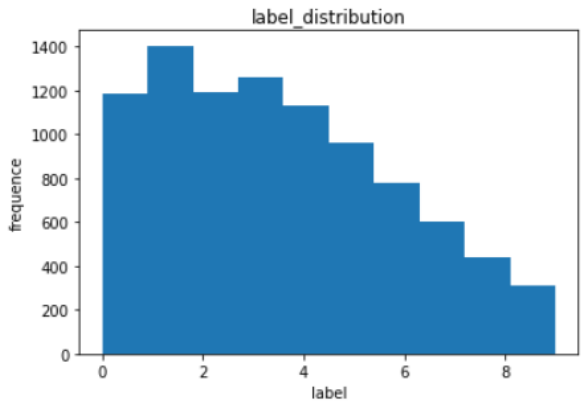
Date: 2021.12.13

In this task, I design a CNN network to classify MINIST images. Unlike conventional neural networks, neurons in each layer of a convolutional neural network are arranged in three dimensions: width, height, and depth. The width and height are well understood, because the convolution itself is a two-dimensional template, but the depth in the convolutional neural network refers to the third dimension of the activated data volume, not the depth of the entire network. The depth of the network refers to the number of layers of the network. Take an example to understand what width, height and depth are. If the image in CIFAR-10 is used as the input of the convolutional neural network, the dimension of the input data volume is 32x32x3 (width, height and depth). We will see that the neurons in the layer will only be connected to a small area in the previous layer, instead of being fully connected. For the convolutional network used to classify images in CIFAR-10, the dimension of the final output layer is 1x1x10, because the final part of the convolutional neural network structure will compress the full-size image into a vector containing the classification score , The vector is arranged in the depth direction.



For task1, I draw a histgraph to present the distribution of 10 label.

```
{0.0: 1183,
 1.0: 1405,
 2.0: 1194,
 3.0: 1262,
 4.0: 1135,
 5.0: 962,
 6.0: 778,
 7.0: 600,
 8.0: 441,
 9.0: 309}
```



I used batch normalization after the pooling in case of over-fitting. Batch normalization transfers the input of each unit of a neural network to a standard normal distribution.

Pros of batch normalization:

1. Shorten the training time, batch normalization makes the convergence faster by flattening the data distribution.
2. Lower the requirement of weight initialization because batch normalization can dynamically adjust the data distribution.
3. More loss function can be supported, and don't need to worry input falls into the dead zone of the activation.
4. Increase the generalization of model and makes it perform better on test.

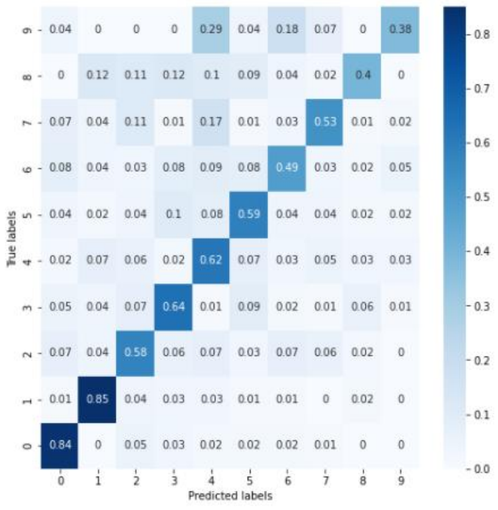
```
CNN(
  (conv1): Conv2d(1, 20, kernel_size=(5, 5), stride=(1, 1))
  (pool1): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
  (conv2): Conv2d(20, 50, kernel_size=(5, 5), stride=(1, 1))
  (norm): BatchNorm1d(400, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (L1): Linear(in_features=800, out_features=400, bias=True)
  (L2): Linear(in_features=800, out_features=400, bias=True)
  (L3): Linear(in_features=400, out_features=10, bias=True)
)
```

I used Adam optimizer and try to optimize cross entropy on 10 classes in this task. I split data into train and validation dataset to test model's performance. Following is the results:

```
              precision    recall  f1-score   support

           0       0.75      0.84      0.79       242
           1       0.79      0.85      0.82       274
           2       0.61      0.58      0.59       239
           3       0.68      0.64      0.66       256
           4       0.58      0.62      0.60       239
           5       0.62      0.59      0.60       200
           6       0.53      0.49      0.51       143
           7       0.57      0.53      0.55       124
           8       0.43      0.40      0.42        82
           9       0.45      0.38      0.41        55

    accuracy                           0.64      1854
   macro avg       0.60      0.59      0.60      1854
weighted avg       0.64      0.64      0.64      1854
```



Reference:

[1] https://zhuanlan.zhihu.com/p/47184529