Exercise: MNIST

1. Create a MNIST prediction model without convolutional computation and compare it with the CNN version.
2. Improve the accuracy of the CNN version.

Submission requirements:

1. source code (MNIST_noCNN.py, MNIST_CNN.py)
2. PDF documents
   Explaining your strategy of (2).
   Show the outputs.
3. Upload to e-learning before 4/12 14:10

Original Accuracy

```
def accuracy():
    temp = 0
    prediction = model2(sample_data).max(dim = 1)[1]
    for i in range(len(prediction)):
        if prediction[i] == sample_targets[i]:
            temp += 1

    return temp / len(prediction)
```

The original accuracy is 99.2%.

Modification

Remove one of the dropout layers.

```
class ConvNet(nn.Module):
    def __init__(self):
        super(ConvNet,self).__init__()
        self.cn1=nn.Conv2d(1, 16, 3, 1)
        self.cn2=nn.Conv2d(16, 32, 3, 1)
        self.dp1=nn.Dropout(0.10)
        self.dp2=nn.Dropout(0.25)
        self.fc1=nn.Linear(12*12*32, 64)
        self.fc2=nn.Linear(64,10)
```

The original structure

```
class ConvNet(nn.Module):
    def __init__(self):
        super(ConvNet,self).__init__()
        self.cn1=nn.Conv2d(1, 16, 3, 1)
        self.cn2=nn.Conv2d(16, 32, 3, 1)
        self.dp1=nn.Dropout(0.10)
        self.fc1=nn.Linear(12*12*32, 64)
        self.fc2=nn.Linear(64,10)
```

The modified structure

The original model structure contains two convolution layers, two dropout layers, and two fully connected layers.

Result

The accuracy increases from 99.2% to 99.4%.