Cifar10 Lab Report

M2 D&K   Xiaoxuan HEI   Yuru HE

The neural network:

In _init_ function:

self.conv1 = nn.Conv2d(3, 16, kernel_size=3, stride=1, padding=1)
self.conv2 = nn.Conv2d(16, 32, kernel_size=3, stride=1, padding=1)
self.conv3 = nn.Conv2d(32, 64, kernel_size=3, stride=1, padding=1)
self.conv4 = nn.Conv2d(64, 128, kernel_size=3, stride=1, padding=1)
self.pool = nn.MaxPool2d(kernel_size=2, stride=2, padding=0)

# Size of the output of the last convolution:
self.flattened_size = 128 * 8 * 8

In *forward* function:
# shape : 3x32x32 → 16x32x32
x = F.relu(self.conv1(x))
# 16x32x32 → 32x32x32
x = F.relu(self.conv2(x))
# 32x32x32 → 32x16x16
x = self.pool(x)
# 32x16x16 → 64x16x16
x = F.relu(self.conv3(x))
# 64x16x16 → 128x16x16
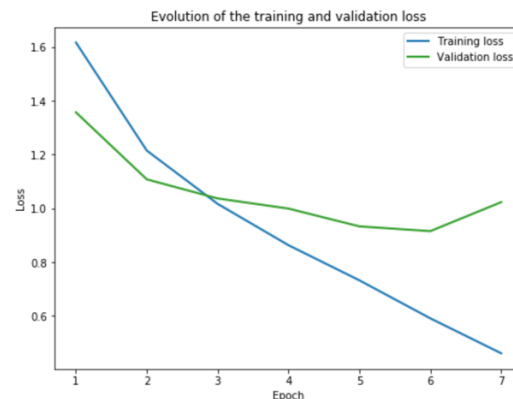x = F.relu(self.conv4(x))
# 128x16x16 → 128x8x8
x = self.pool(x)

Considering about accuracy and running time, after many attempts, we chose *batch_size=32, n_epochs=7*. From the results, the batch size doesn't have huge impact on the accuracy. The bigger the number of epochs is, the longer the running time is.

Batch_size	Number of epochs	Accuracy on the 5000 test images	Time(s)
32	6	68.12 %	42.48
32	7	69.86 %	49.27
32	10	67.72 %	68.67
64	7	68.84 %	38.99
64	8	67.60 %	44.31
64	10	66.96 %	55.93
64	20	67,00 %	113.51
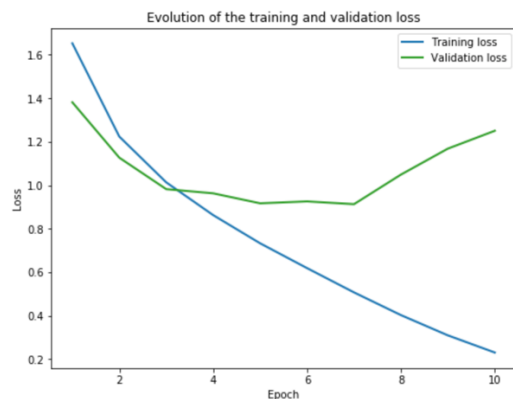
batch_size=32, n_epochs=10   *batch_size=32, n_epochs=7*

We can find some waves in the curve of validation loss in the left picture, which means that overfitting might occur. So we change the number of epochs from 10 to 7 to make an early stop. And the result is as shown in the right picture.

The accuracy before:

```
Computing accuracy...
Accuracy of the network on the 20000 train images: 75.10 %
Accuracy of the network on the 5000 validation images: 59.62 %
Accuracy of the network on the 5000 test images: 59.56 %
```

```
Class        Accuracy (%)
plane           68.85
car             74.06
bird            48.63
cat             39.03
deer            58.38
dog             46.72
frog            69.65
horse           63.43
ship            66.67
truck           60.23
```

The accuracy after our CNN:

```
Accuracy of the network on the 20000 train images: 85.98 %
Accuracy of the network on the 5000 validation images: 69.76 %
Accuracy of the network on the 5000 test images: 69.86 %
```

```
Class        Accuracy (%)
plane           74.18
car             85.15
bird            59.96
cat             44.47
deer            68.84
dog             52.25
frog            76.78
horse           78.18
ship            81.55
truck           76.80
```
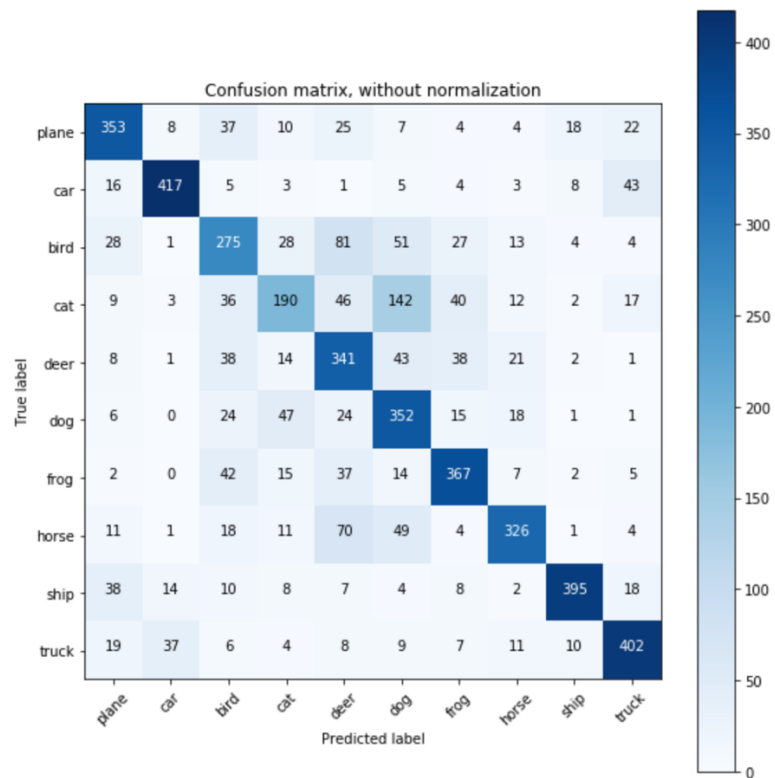
We can find that the accuracy has improved a lot, especially for horses and trucks.

The confusion matrix before:

Confusion matrix, without normalization
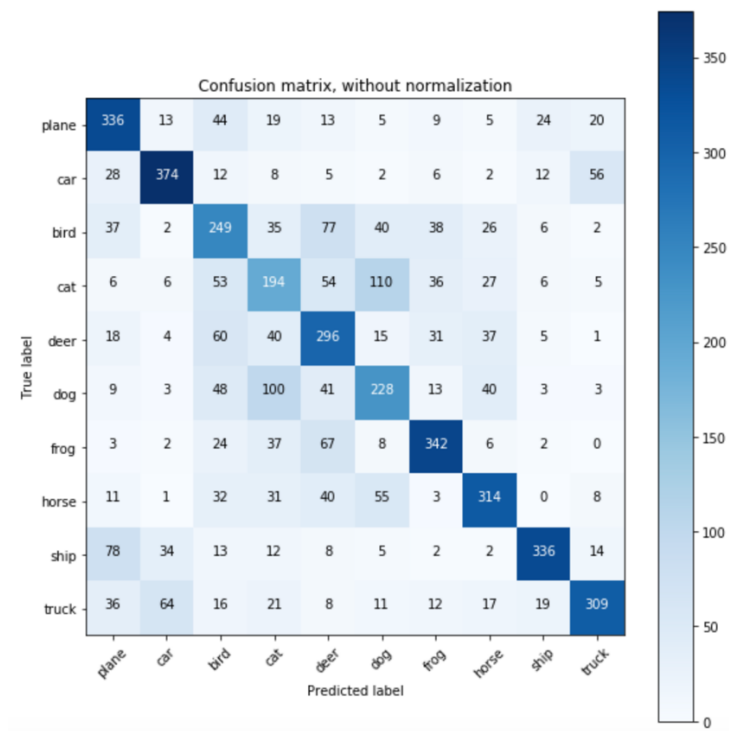
The confusion matrix after our CNN:

Confusion matrix, without normalization

We notice that our network is pretty good at classifying cars, ships and trucks, but there are still some difficulties to differentiate cats and dogs. It classifies a lot of cats as dogs. And it classifies a lot of birds as deers.