

cifar10_cnn

兰欣泽 2018012036 生医81

1、层的选择

在层的选择上，最初选择了两个卷积层和三个全连接层，选用ReLU，并采用MaxPool2d压缩图象

```
CnnNet(  
    (conv1): Conv2d(3, 6, kernel_size=(5, 5), stride=(1, 1))  
    (pool): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)  
    (conv2): Conv2d(6, 16, kernel_size=(5, 5), stride=(1, 1))  
    (fc1): Linear(in_features=400, out_features=120, bias=True)  
    (fc2): Linear(in_features=120, out_features=84, bias=True)  
    (fc3): Linear(in_features=84, out_features=10, bias=True)  
)
```

训练了25个epoch后发现正确率只有60%左右，且在加大epoch后出现了过拟合现象。

之后尝试了加大层的数量，尝试使用了5个卷积层和4个全连接层，发现过拟合现象过于严重，且准确率反而降低了，且层数增加导致训练速度很慢。

之后尝试了一些方法解决过拟合问题，因为上次作业的网络已经尝试过weight decay和early stopping，所以决定尝试新方法，使用了dropout，在训练过程中随机将部分神经元的权重置为0，缩减了参数量。

使用3个卷积层和3个全连接层。

```
CnnNet(  
    (conv): Sequential(  
      (0): Conv2d(3, 16, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))  
      (1): ReLU()  
      (2): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)  
      (3): Conv2d(16, 32, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))  
      (4): ReLU()  
      (5): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)  
      (6): Conv2d(32, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))  
      (7): ReLU()  
      (8): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)  
    )  
    (fc): Sequential(  
      (0): Linear(in_features=1024, out_features=512, bias=True)  
      (1): ReLU()  
      (2): Dropout(p=0.5, inplace=False)  
      (3): Linear(in_features=512, out_features=64, bias=True)  
      (4): ReLU()  
      (5): Dropout(p=0.5, inplace=False)  
      (6): Linear(in_features=64, out_features=10, bias=True)  
    )  
)
```

训练了50个epoch后，总的正确率在72%左右，且速度较快。

查阅资料发现想要继续提高正确率，可以使用VGG等模型，或对图象进行一定的处理，将在大作业中体现。

2、主要函数

Loss采用CrossEntropyLoss，optimizer选择SGD且尝试了几个不同的learning rate和momentum，最后选择了0.01和0.1。

训练过程中，注意需要使用Variable使得值可以不断改变，每次用zero_grad()将梯度归零，直接使用.backward()进行反向传播的梯度计算，再用.step()对optimizer的参数进行更新。

3、遇到的困难

由于采用的是最基本的CNN网络，整体并未遇到什么较大困难。

在选择模型与超参数时，每次训练的时间还是较长的，导致整体调试完成情况较慢。

4、结果与总结

最终使用最简单的CNN的准确率在72%左右，认为可以接受，更复杂的网络与方法将在大作业中体现。

对于每一种的准确率：

```
Accuracy of plane : 77 %  
Accuracy of   car : 87 %  
Accuracy of  bird : 61 %  
Accuracy of   cat : 54 %  
Accuracy of  deer : 66 %  
Accuracy of   dog : 67 %  
Accuracy of  frog : 80 %  
Accuracy of horse : 78 %  
Accuracy of  ship : 81 %  
Accuracy of truck : 81 %
```

可以发现对于动物分类的准确率很低，尤其是猫和鸟，可能是颜色对结果有一定影响，如果不改变结构，增加一个代表颜色的卷积层可能会对提升准确率有一定帮助。在大作业中可以尝试使用VGG或ResNet等模型提高准确率。