

ASSIGNMENT #2

KW-Vertically Integrated Project 2020-2

심동규 교수님

소프트웨어학부 3학년

2017203062 권나성

개요

CIFAR10 데이터셋 이미지를 분류하는 것이 이번 과제 목표입니다.

이미지는 airplane, automobile, bird, cat, deer, dog, frog, horse, ship, truck으로 총 10개 카테고리입니다.

구현 방법

Pytorch 라이브러리를 이용하여 구현하였습니다.

CNN model 구성은 4개의 Convolutional layer와 2개의 Fully Connected layer로 이루어져 있고, dropout 함수를 사용하여 overfitting을 방지할 수 있도록 설계하였습니다.

Hyper parameter같은 경우, Epoch은 30으로 하였고, batch size는 64로 하였습니다.

CIFAR 10의 클래스 수는 10개이므로, num_classes는 10으로 해주었고, learning rate는 0.001로 하였습니다.

layer를 깊게 쌓기 위해 Cifar10의 32x32 사이즈 input image를 64x64로 크기를 조정해주었습니다.

다.

Dataset은 torchvision.datasets.CIFAR10을 사용하여 다운받았습니다.

Loss function 같은 경우, cross entropy loss 를 사용하였고, optimizer는 Adam을 사용하였습니다.

구현한 네트워크 구조는 다음과 같습니다.

```
class ConvNet(nn.Module):
    def __init__(self, num_classes=10):
        super(ConvNet, self).__init__()
        self.layer1=nn.Sequential(
            nn.Conv2d(3,6,kernel_size=5, stride=1, padding=2),
            nn.ReLU(),
            nn.MaxPool2d(kernel_size=5, stride=2))
        self.layer2= nn.Sequential(
            nn.Conv2d(6,16,kernel_size=5, stride=1, padding=2),
            nn.ReLU(),
            nn.MaxPool2d(kernel_size=2, stride=2))
        self.layer3 = nn.Sequential(
            nn.Conv2d(16, 32, kernel_size=5, stride=1, padding=2),
            nn.ReLU(),
            nn.MaxPool2d(kernel_size=2, stride=2))
        self.layer4 = nn.Sequential(
            nn.Conv2d(32, 64, kernel_size=5, stride=1, padding=2),
            nn.ReLU(),
            nn.MaxPool2d(kernel_size=2, stride=2))

        self.fc_layer = nn.Sequential(
            nn.Linear(64*3*3, 120),
            nn.Linear(120, num_classes))

    def forward(self, x):
        out = self.layer1(x)
        out = self.layer2(out)
        out=self.layer3(out)
        out = self.layer4(out)
        out = F.dropout(out, training=self.training)
        out=out.reshape(out.size(0),-1)
        out=self.fc_layer(out)
        return out
```

결과 화면

최종 학습 모델에 대한 각 클래스 별 정확도입니다.

```
Accuracy of plane : 76 %  
Accuracy of   car : 80 %  
Accuracy of  bird : 69 %  
Accuracy of   cat : 53 %  
Accuracy of  deer : 54 %  
Accuracy of   dog : 47 %  
Accuracy of  frog : 78 %  
Accuracy of horse : 70 %  
Accuracy of  ship : 82 %  
Accuracy of truck : 76 %
```