

ASSIGNMENT #3

KW-Vertically Integrated Project 2020-2

심동규 교수님

소프트웨어학부 3학년

2017203062 권나성

개요

ImageNet 의 일부 데이터셋으로 제가 구성한 네트워크에 학습시켜 데이터셋 이미지를 분류하고 train과 validation의 accuracy와 loss를 출력하는 것이 이번 과제 목표입니다.

이미지는 ants와 bees 로 총 2개 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 10으로 하였고, batch size는 64로 하였습니다.

데이터셋 의 클래스 수는 2개이므로, num_classes는 2로 해주었고, learning rate는 0.001로 하였습니다.

layer를 깊게 쌓기 위해 32x32 사이즈 input image를 224x224로 크기를 조정해주었습니다.

Dataset은 ImageNet의 ants와 bees dataset 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=2):
        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*13*13, 4),
            nn.Linear(4, num_classes))
```

결과 화면

Train dataset과 Test dataset 에 대한 각각의 Accuracy와 Validation 입니다.

```
train Loss: 0.6517 Acc: 0.5615
val Loss: 0.6972 Acc: 0.6275
```