Assignment 3

2018707017 정용훈

●개요

주어진 데이터 셋을 활용해 CNN모델을 사용한다. 이번에 주어진 데이터는 CIFAR보다 적은 2개의 Classification이고, 데이터 양도 적다.

또한, 설계한 여러 모델들을 비교하면서 어느 모델이 정확도면에서 좋은 결과를 나타내는지 분석한다.

●구현 방법

```
class Net(torch.nn.Module):
    def __init__(self,num_classes=2):
        super(Net, self).__init__()

        self.layer1 = nn.Sequential(
            nn.Conv2d(3, 32,kernel_size=5, stride=1, padding=2),
            nn.ReLU(),
            nn.MaxPool2d(kernel_size=2,stride=2)
        )
        self.layer2 = nn.Sequential(
            nn.Conv2d(32, 64, kernel_size=5, stride=1, padding=2),
            nn.ReLU(),
            nn.MaxPool2d(kernel_size=2, stride=2)
        )
        self.layer3 = nn.Sequential(
            nn.Conv2d(64, 128, kernel_size=5, stride=1, padding=2),
            nn.ReLU(),
            nn.MaxPool2d(kernel_size=2, stride=2)
        )
        self.fc = nn.Linear(28* 28 * 128, num_classes)

    def forward(self, x):
        out = self.layer1(x)
        out = self.layer2(out)
        out = self.layer3(out)
        out = out.reshape(out.size(0), -1)
        out = self.fc(out)
        return out
```

<그림 1>

<그림 1>은 내가 설계한 Model 이다. Layer 1, Layer 2, Layer 3는 지난 Assignment #2와 동일하게 설정했다. 이제 각각의 Argument들을 설정한 과정을 살펴보자.

1. Layer 1

```
data_transforms = {
    'train': transforms.Compose([
        transforms.RandomResizedCrop(224),
        transforms.RandomHorizontalFlip(),
       # transforms.RandomVerticalFlip(),
        transforms.RandomVerticalFlip(),
        transforms.ToTensor(),
        transforms.Normalize([0.485, 0.456, 0.406],[0.229,0.224,0.225])
    ]),
    'val': transforms.Compose([
            transforms.Resize(256),
            transforms.CenterCrop(224),
            transforms.ToTensor(),
            transforms.Normalize([0.485, 0.456, 0.406],[0.229,0.224,0.225])
        ])
}
```

<그림 2>

<그림 2>에서 데이터를 불러오는 과정에서 데이터 size를 (224, 224)으로 설정했다.

따라서, Layer 1의 Output size = $\frac{224-5+2*2}{1}+1=224$이고, Max Pooling 을 거치면 절반인 112 x 112 이다.

2. Layer 2

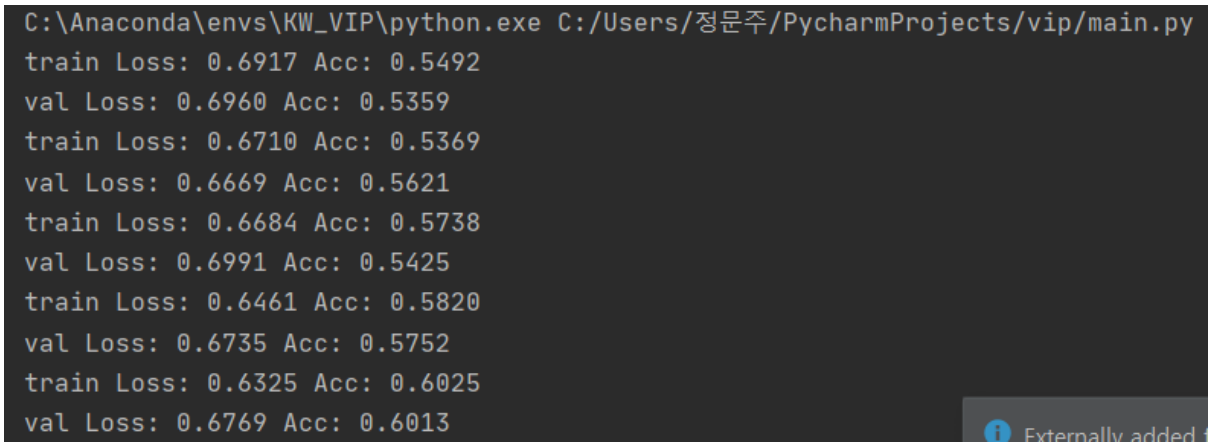
Layer 1의 Output size가 112 x 112 이므로, Layer 2의 Output size = $\frac{112-5+2*2}{1}+1=112$ 이고, Max Pooling을 거치면 절반인 56 x 56 이다.

3. Layer 3

Layer 2의 Output size가 56 x 56 이므로, Layer 3의 Output size = $\frac{56-5+2*2}{1}+1=56$ 이고, Max Pooling을 거치면 절반인 28 x 28 이다.

따라서, Layer 3의 Output size가 28 x 28 x 128 이므로 <그림 1>에서 self.fc의 Input size를 28 x 28 x 128로 설정했다.

●결과 화면



Train Loss	Validation Loss
0.6917	0.6960
0.6710	0.6669
0.6684	0.6991
0.6461	0.6735
0.6325	0.6769

다양한 Argument를 설정했지만, 최종적으로 가장 괜찮은 Training은 위 결과와 같다.

데이터 양이 적어서 Accuracy는 약 60%인 것을 확인할 수 있다.