

KW_VIP 과제2

2017741009 로봇학부 장우현

*과제 목표

CIFAR10 이미지 dataset을 불러와서, CNN을 통해 그 데이터들을 image classificaton(이미지 분류)하는 프로그램을 작성해보자.

*구현 방법

```
#-----Convolutional neural network-----
#-----two convolutional layers-----
class ConvNet(nn.Module):
    def __init__(self, num_classes = 10):
        super(ConvNet, self).__init__()
        self.layer1 = nn.Sequential(
            nn.Conv2d(3,8,kernel_size=5,padding=2),
            nn.ReLU(),
            nn.MaxPool2d(kernel_size=2, stride=2)
        )
        self.layer2 = nn.Sequential(
            nn.Conv2d(8,16,kernel_size=5,padding=2),
            nn.ReLU(),
            nn.MaxPool2d(kernel_size=2, stride=2)
        )
        self.fc = nn.Linear(8*8*16,num_classes)

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

CNN클래스 선언부이다. layer는 총 2개로 구성이 되어있다. 첫 nn.Conv2d의 input channel 값은 컬러사진이기 때문에 RGB값인 3 channel로 한다. 그리고 필터의 크기인 커널 사이즈는 5 * 5로, 각 구간의 픽셀에서 가장 큰 값을 가지는 Maxpooling의 사이즈는 2 * 2로 설정해주었다.

```
#-----Training-----
total_step = len(train_loader)
for epoch in range(num_epochs):
    for i, (images, labels) in enumerate(train_loader):
        images = images.to(device)
        labels = labels.to(device)

        outputs = model(images)
        loss = criterion(outputs, labels)

        optimizer.zero_grad()
        loss.backward()
        optimizer.step()
```

받은 cifar데이터시트를 트레이닝하는 부분이다. 에폭 파라미터값을 10으로 해줘서, 바깥 for문을 통해 각 샘플 데이터를 10번 돌며 학습하게 되어있다. criterion = nn.CrossEntropy()를 loss에 대입하면서 예측값과 정답의 차이를 계산하게 하고, 그런다음 그 차이를 점점 최소화시키기위한 gradient descent 직전에 zero_grad()로 초기화를 해준다. 그리고 loss에 대한 gradient를 구해주고, 이로 인한 값들을 step해서 model을 업데이트를 해준다.

```
#-----Test-----
className = ('plane', 'car', 'bird', 'cat', 'deer', 'dog', 'frog', 'horse', 'ship', 'truck')
model.eval()

with torch.no_grad():
    correct = list(0. for i in range(10))
    total = list(0. for i in range(10))
    for images, labels in test_loader:
        images = images.to(device)
        labels = labels.to(device)
        outputs = model(images)
        _, predicted = torch.max(outputs.data, 1)
        c = (predicted == labels).squeeze()
        for i in range(4):
            label = labels[i]
            total[label] += 1
            correct[label] += c[i].item()

    for i in range(10):
        print('Accuracy of %5s : %2d %%' % (
            className[i], 100 * correct[i] / total[i]))
```

10마리의 동물 이름배열과 correct, total 배열초기값을 설정해주고 for문을 통해, 각 동물에 해당하는 인덱스의 각 배열에 대입한다. 그리고 마지막 for문에서, 동물의 수만큼 반복시키면서 각 동물의 이름과, 트레이닝한 정보와 실제값의 비를 통해 정확도를 출력한다.

*과제 결과

```
D:\Anaconda\envs\KW_VIP\python.exe C:/Users/user/PycharmProjects/Assignment/Assignment2/assignment.py
Files already downloaded and verified
Files already downloaded and verified
Accuracy of plane : 66 %
Accuracy of car : 79 %
Accuracy of bird : 58 %
Accuracy of cat : 32 %
Accuracy of deer : 48 %
Accuracy of dog : 46 %
Accuracy of frog : 79 %
Accuracy of horse : 69 %
Accuracy of ship : 69 %
Accuracy of truck : 75 %

Process finished with exit code 0
```