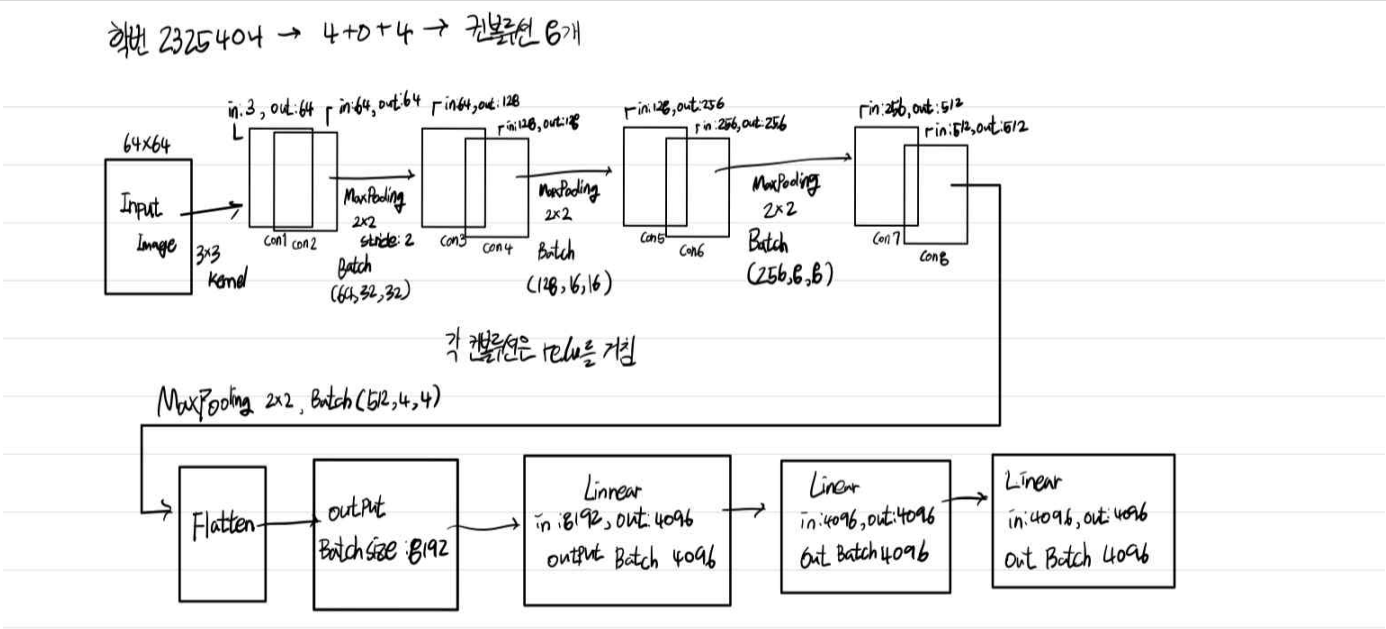
기계학습 과제

학과	학년	학번	이름
컴퓨터공학과	4	2325404	이인하



모자, 상의, 하의 이미지 분류를 위한 합성곱 신경망 모델로 구조는 총 8개의 합성곱 레이어와 3개의 완전 연결 레이어로 구성됩니다.
합성곱 레이어는 입력 이미지에 3개의 채널(RGB)을 받아들이고, 64개의 필터를 사용하여 출력을 생성합니다. 출력은 입력 이미지와 동일한 64x64 크기를 갖습니다.
나머지 합성곱 레이어는 출력 크기를 점차적으로 감소시키는 맥스 풀링 레이어와 함께 사용했습니다.
완전 연결 레이어 마지막 합성곱 레이어의 출력은 4x4 크기의 그리드로 이루어져 있습니다. 이를 1차원으로 Flatten을 한 후 3개의 완전 연결 레이어를 통해 최종 클래스 예측을 수행합니다.
모자, 상의, 하의 이미지 각각 30개를 넣었으며 총 500 에포크 동안 훈련되었습니다.

조건1 (컨볼루션층 수):

```
self.conv1 = nn.Conv2d(in_channels=3, out_channels=64, kernel_size=3, padding=1)
self.pool = nn.MaxPool2d( kernel_size: 2,  stride: 2)
self.conv2 = nn.Conv2d(in_channels=64, out_channels=64, kernel_size=3, padding=1)
self.conv3 = nn.Conv2d(in_channels=64, out_channels=128, kernel_size=3, padding=1)
self.conv4 = nn.Conv2d(in_channels=128, out_channels=128, kernel_size=3, padding=1)
self.conv5 = nn.Conv2d(in_channels=128, out_channels=256, kernel_size=3, padding=1)
self.conv6 = nn.Conv2d(in_channels=256, out_channels=256, kernel_size=3, padding=1)
self.conv7 = nn.Conv2d(in_channels=256, out_channels=512, kernel_size=3, padding=1)
self.conv8 = nn.Conv2d(in_channels=512, out_channels=512, kernel_size=3, padding=1)
```

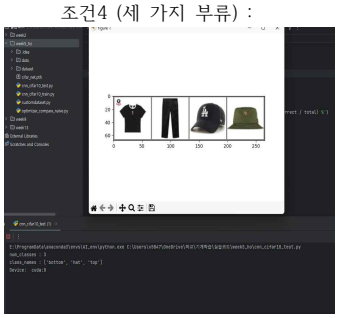
조건2 (하나 이상의 Relu함수 및 Maxpooling layer):

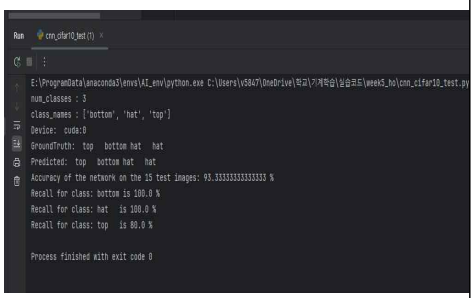
```
    x = F.relu(self.conv1(x))
    x = F.relu(self.conv2(x))
    x = self.pool(x)

    x = F.relu(self.conv3(x))
    x = F.relu(self.conv4(x))
    x = self.pool(x)

    x = F.relu(self.conv5(x))
    x = F.relu(self.conv6(x))
    x = self.pool(x)

    x = F.relu(self.conv7(x))
    x = F.relu(self.conv8(x))
    x = self.pool(x)

    x = torch.flatten(x, 1)
    x = F.relu(self.fc1(x))
    x = F.relu(self.fc2(x))
    x = self.fc3(x)
```

조건3 (완전연결층의 은닉층 2개 구성) :

```
x = torch.flatten(x, 1)
x = F.relu(self.fc1(x))
x = F.relu(self.fc2(x))
x = self.fc3(x)
return x
```

조건4 (세 가지 부류) :





모델의 정확도를 높이기 위해서는 더 많은 합성곱 레이어와 완전 연결 레이어를 추가하거나 훈련 데이터셋에 다양한 변형을 적용하여 데이터의 다양성을 높일 수 있습니다. 또한 학습률, 배치 크기, 옵티마이저의 모멘텀 등과 같은 하이퍼파라미터를 조정하는 등 여러 방법이 있습니다.