

Cnn 과제

YBIGTA 20기 이시영

1. Convolutional Neural Networks(이하 CNN)에 대한 설명으로 옳지 않은 것은?

1번 - Convolution 연산이란, 이미지 위에서 stride 값 만큼 filter(kernel)를 이동시키면서 겹쳐지는 부분의 각 원소의 값을 곱해서 모두 더한 값을 출력으로 하는 연산이다.

2번 - CNN은 Filter와 이미지의 Convolution으로 이미지의 Feature를 추출해내는 모델이다.

3. CNN은 parameter를 공유하여 전체 parameter 수를 줄여주기 때문에 overfitting이 일어날 가능성이 DNN보다 더 높다.

정답: 3번

2. CNN 모델을 구축하는 과정에서 다음과 같은 코드를 이용하여 필터(커널)를 만들어주었다.

```
1 conv = torch.nn.Conv2d(1,1,3)
```

다음에 대해 맞으면 True 틀리면 False 를 선택하시오.

"이 필터는 입력채널의 크기가 1, 출력채널의 크기가 1, 필터의 크기가 3*3인 필터이다."

정답: True

3. 다음과 같이 conv 의 이름으로 convolution layer 필터를 만들어 주고 inputs 를 넣어주었다.

```
1 conv = torch.nn.Conv2d(1,1,3)
2 inputs = ( A , B , C , D )
3 output = conv(inputs)
```

A, B, C, D 순서대로 쓰세요. 채널, Width, Height, 배치사이즈

정답 : 배치사이즈, 채널, height, width

4. 채널이 8인 63x63 input 이미지와 7x7의 16채널 필터를 "stride=1"로 convolution 연산을 하되, input과 같은 크기의 output 결과를 가져오도록 하려고 한다. 이 때, 얼마의 padding을 주어야 하는가?

정답 : 3

5. 다음 용어들에 대한 간단한 정의 혹은 설명을 쓰시오

Convolution 연산: 이미지의 형상을 무시하지 않고 이미지를 그대로 인공 신경망이 학습할 수 있게 해주는 수학 행렬 연산이다. 합성곱에서 원본 이미지는 학습해야 할 사진 데이터(행렬로 변환)이고, 필터(filter)는 원본 이미지에서 특징을 잡아내는데 사용되는 행렬이다. 특징을 잡아서 feature map을 생성해 원본이미지의 형태를 이해하는 것이다.

Padding: Convolution 레이어에서 Filter와 Stride의 작용으로 Feature Map의 크기는 입력 데이터보다 작다. Convolution 레이어의 출력 데이터가 줄어드는 것을 방지하는 방법이 Padding이다. Padding은 입력 데이터의 외각에 지정된 픽셀만큼 특정 값으로 채워 넣는 것을 의미한다. 보통 패딩 값으로 0으로 채워 넣는다(Zero-padding).

Channel: Convolution Layer에 유입되는 입력 데이터에는 한 개 이상의 필터가 적용된다. 1개 필터는 Feature Map의 채널이 된다. Convolution Layer에 n개의 필터가 적용된다면 출력 데이터는 n개의 채널을 갖게 된다.

Stride: 필터가 몇 칸씩 이동하는지 알려준다.

Filter: 이미지의 특징을 찾아내기 위한 정사각 행렬로 이루어지며 공유 파라미터이다. CNN에서 Filter와 Kernel은 같은 의미이다.

Pooling: convolution 레이어로부터 입력 받은 데이터의 크기를 줄이거나 특정 데이터를 강조하는 용도로 사용된다. Max pooling과 average pooling이 대표적으로 사용된다.

6. Conv 연산을 한 후 학습을 위해서는 nn.Linear()을 거쳐 1차원 벡터로 변경해야 한다.

정답: True

Tensorflow 과제

(죄송해요 구버전 tf1으로 작성된 파일을 수정했어요~)

과제는 구글드라이브에 올라가 있습니다!! 다운받아서 사용하세요.

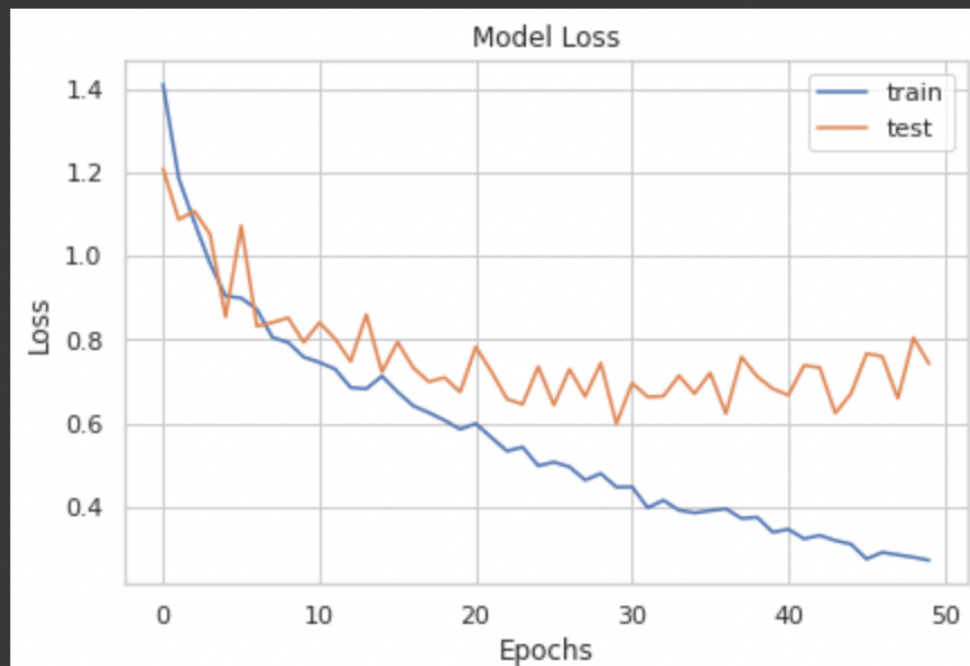
?를 채워서 모델을 돌린 후 결과화면 캡처 후 아래에 복붙!!

4) Evaluating the Model Performance

✓
0s

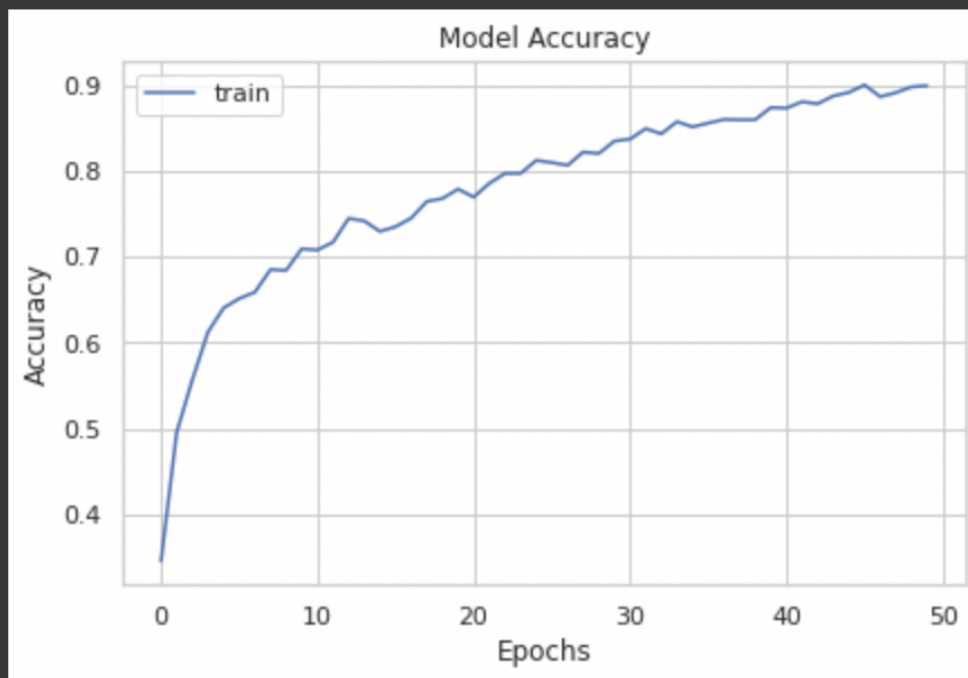


```
plt.plot(History.history['loss'])
plt.plot(History.history['val_loss'])
plt.title('Model Loss')
plt.ylabel('Loss')
plt.xlabel('Epochs')
plt.legend(['train', 'test'])
plt.show()
```



✓
0s

```
[25] plt.plot(History.history['accuracy'])  
      plt.title('Model Accuracy')  
      plt.ylabel('Accuracy')  
      plt.xlabel('Epochs')  
      plt.legend(['train', 'test'])  
      plt.show()
```



CORRECTLY CLASSIFIED FLOWER IMAGES