

# Cnn 과제

YBIGTA 20기 박슬빈

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 padding 을 주어야 하는가?

정답 : 3

5. 다음 용어들에 대한 간단한 정의 혹은 설명을 쓰시오

Convolution 연산: 신호를 커널을 이용해 국소적으로 증폭 또는 감소 스켜서 정보를 추출 또는 필터링 하는 것이다.

Padding: 아웃풋이 인풋 사이즈보다 작아지고 이미지의 가장자리가 덜 사용되는 걸 방지하기 위해 가장자리에 임의로 0의 값을 가진 픽셀을 추가하는 것이다.

Channel: 이미지의 깊이이다 흑백은 1 rgb는 3

Stride: 필터가 이동하는 단위

Filter: 정사각형 행렬이고 이미지를 스캔하면서 이미지의 특성을 찾아낸다

Pooling: 데이터의 크기를 줄이거나 특정 데이터를 강조한다. Max pooling과 average pooling이 있는데 max를 더 많이 쓴다.

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!

```
#물음표를 채워주세요.!\n#Hint Padding은 전부 대중적인 즉 stride가 10이라면 input과 output 사이즈를 같게 만들어주는 패딩\n#Hint 커널사이즈도 대중적인 커널사이즈\n#은닉층에 주로쓰는 Activation\nmodel = Sequential()\nmodel.add(Conv2D(filters = 32, kernel_size = (3,3),padding = 'same',activation = 'relu', input_shape = (150,150,3)))\nmodel.add(MaxPooling2D(pool_size=(2,2)))\n\nmodel.add(Conv2D(filters = 64, kernel_size = (3,3),padding = 'same',activation = 'relu'))\nmodel.add(MaxPooling2D(pool_size=(2,2), strides=(2,2)))\n\nmodel.add(Conv2D(filters =96, kernel_size = (3,3),padding = 'same',activation = 'relu'))\nmodel.add(MaxPooling2D(pool_size=(2,2), strides=(2,2)))\n\nmodel.add(Conv2D(filters = 96, kernel_size = (3,3),padding = 'same',activation = 'relu'))\nmodel.add(MaxPooling2D(pool_size=(2,2), strides=(2,2)))\n\nmodel.add(Flatten())\nmodel.add(Dense(512))\nmodel.add(Activation('relu'))\nmodel.add(Dense(5, activation = "softmax")) #우리가 몇개로 분류하나요?, 다중분류에 맞는 activation
```

```
Epoch 40/50\n25/25 [=====] - 18s 728ms/step - loss: 0.3728 - accuracy: 0.8643 - val_loss: 0.6884 - val_accuracy: 0.7796\nEpoch 41/50\n25/25 [=====] - 18s 727ms/step - loss: 0.3711 - accuracy: 0.8594 - val_loss: 0.6444 - val_accuracy: 0.8056\nEpoch 42/50\n25/25 [=====] - 18s 726ms/step - loss: 0.3476 - accuracy: 0.8707 - val_loss: 0.6020 - val_accuracy: 0.8056\nEpoch 43/50\n25/25 [=====] - 18s 731ms/step - loss: 0.3351 - accuracy: 0.8762 - val_loss: 0.6787 - val_accuracy: 0.7944\nEpoch 44/50\n25/25 [=====] - 18s 728ms/step - loss: 0.3354 - accuracy: 0.8749 - val_loss: 0.6485 - val_accuracy: 0.8009\nEpoch 45/50\n25/25 [=====] - 18s 729ms/step - loss: 0.3123 - accuracy: 0.8797 - val_loss: 0.6759 - val_accuracy: 0.7981\nEpoch 46/50\n25/25 [=====] - 19s 753ms/step - loss: 0.3086 - accuracy: 0.8881 - val_loss: 0.6376 - val_accuracy: 0.8093\nEpoch 47/50\n25/25 [=====] - 18s 732ms/step - loss: 0.2920 - accuracy: 0.8890 - val_loss: 0.6358 - val_accuracy: 0.7991\nEpoch 48/50\n25/25 [=====] - 18s 728ms/step - loss: 0.2979 - accuracy: 0.8910 - val_loss: 0.7247 - val_accuracy: 0.7935\nEpoch 49/50\n25/25 [=====] - 19s 741ms/step - loss: 0.2636 - accuracy: 0.9028 - val_loss: 0.6504 - val_accuracy: 0.8120\nEpoch 50/50\n25/25 [=====] - 18s 724ms/step - loss: 0.2501 - accuracy: 0.9032 - val_loss: 0.6626 - val_accuracy: 0.8111
```

