CNN 과제

YBIGTA 20기 박준하

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.

```
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."

(True)

`torch.nn.Conv2d(in_channels, out_channels, kernel_size, stride=1, padding=0, dilation=1, groups=1, bias=True, padding_mode='zeros', device=None, dtype=None)`

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

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

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

정답 : A – 배치사이즈, B – 채널, C – Width, D – Height

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

Output size = (input size – filter size + 2*padding)/stride + 1

63 = (63 – 7 + 2*x) + 1, x = 3

정답 : padding = 3

5. 다음 용어들에 대한 간단한 정의 혹은 설명을 쓰시오

Convolution 연산: 합성곱 연산, 크기가 같은 두 Matrix의 element-wise 곱을 전부 더한 값을 도출한다. 크기가 작은 Filter를 활용해 크기가 큰 Input Matrix에 Convolution 연산을 수행하면 window를 옮기면서 합성곱 연산을 순차적으로 수행, 각 위치에서의 합성곱 연산값을 entry로 하는 Output Matrix를 도출한다.

Padding: 합성곱 연산을 수행하면 일반적으로 가장자리의 정보들은 Filter와의 Convolution 연산 이후 사라지게 되고, Output Matrix의 크기가 더 작아지게 된다. 이를 방지하기 위해서 Padding 기법을 활용하는데, Input Matrix의 가장자리에 가상의 값(이를 테면 0)이 있다고 가정하고 확장된 Matrix에 대해서 합성곱 연산을 수행하게 된다. Padding은 주로 Edge 정보를 보존하거나, Input과 Output Matrix의 크기를 유지시킬 때 활용하게 되며 Zero Padding(0으로 채움), Reflection Padding(안쪽의 값을 대칭시켜서 덧붙임) 등의 방법이 있다.

Channel: 일반적인 이미지는 R,G,B 세 개의 채널로 구성되어 있고, 흑백 이미지의 경우 한 개의 채널로 구성되어 있다. 여기서 Channel은 처음 Input 이미지의 R,G,B 채널을 의미하기도 하지만

Convolution 연산 과정에서 10개의 필터를 적용하게 되면 각각의 필터는 이전 Output 이미지의 각 채널마다 Convolution 연산을 수행하여 연산값을 모두 더한 것을 최종 Output으로 도출하게 되고, 10개의 필터를 적용했으므로 총 10개의 Output 층이 생기게 되므로 10개의 채널로 구성되었다고 설명할 수 있다.

Stride: Convolution 연산을 수행할 때 window가 이동하는 폭을 stride라고 한다. 일반적으로는 stride = 1로 설정하여 한 칸씩 window가 slide하지만, stride=2,3 등으로 변형하여 활용할 수 있다. 그리고 (Input size – Filter size + 2* Padding)이 stride의 배수일 때만 Convolution 연산이 가능하다.

Filter: Convolution Layer에서 window를 이동시키면서 합성곱 연산을 하는 Matrix를 Filter라고 한다. 일반적으로 Filter size는 홀수인데, 왜냐하면 Filter의 center를 정의하고 Padding 과정에서의 대칭성을 확보하기 위해서이다.

Pooling: Pooling은 대표적인 dimensionality reduction 과정으로, 예를 들어 2X2 pooling이라고 하면 2X2 영역 내의 값들을 바탕으로 하나의 대표값을 추출하는 과정이다. Pooling에는 최대값을 추출하는 Max Pooling이 있고, 평균값을 사용하는 Average Pooling이 있다.

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

(True)

Tensorflow 과제

```
Epoch 40/50
25/25 [==============================] - 18s 734ms/step - loss: 0.3712 - accuracy: 0.8607 - val_loss: 0.7068 - val_accuracy: 0.7778
Epoch 41/50
25/25 [==============================] - 18s 733ms/step - loss: 0.3283 - accuracy: 0.8717 - val_loss: 0.6465 - val_accuracy: 0.8093
Epoch 42/50
25/25 [==============================] - 18s 732ms/step - loss: 0.3222 - accuracy: 0.8826 - val_loss: 0.6576 - val_accuracy: 0.8019
Epoch 43/50
25/25 [==============================] - 19s 734ms/step - loss: 0.2970 - accuracy: 0.8926 - val_loss: 0.6995 - val_accuracy: 0.7898
Epoch 44/50
25/25 [==============================] - 18s 733ms/step - loss: 0.3275 - accuracy: 0.8820 - val_loss: 0.6773 - val_accuracy: 0.8046
Epoch 45/50
25/25 [==============================] - 18s 727ms/step - loss: 0.2926 - accuracy: 0.8884 - val_loss: 0.7363 - val_accuracy: 0.7870
Epoch 46/50
25/25 [==============================] - 19s 736ms/step - loss: 0.2987 - accuracy: 0.8852 - val_loss: 0.7558 - val_accuracy: 0.7880
Epoch 47/50
25/25 [==============================] - 18s 715ms/step - loss: 0.2896 - accuracy: 0.8939 - val_loss: 0.6672 - val_accuracy: 0.8009
Epoch 48/50
25/25 [==============================] - 18s 755ms/step - loss: 0.2600 - accuracy: 0.9022 - val_loss: 0.7224 - val_accuracy: 0.7944
Epoch 49/50
25/25 [==============================] - 18s 732ms/step - loss: 0.2562 - accuracy: 0.9067 - val_loss: 0.7448 - val_accuracy: 0.7917
Epoch 50/50
25/25 [==============================] - 18s 731ms/step - loss: 0.2656 - accuracy: 0.9016 - val_loss: 0.7089 - val_accuracy: 0.8102
```