

파이토치로 배우는 딥러닝 기초

CNN 퀴즈 정답



Quiz 1 해설

Quiz

정답 Q 1 (단일선택, 10점)

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

1. Convolution 연산이란, 이미지 위에서 stride 값 만큼 filter(kernel)를 이동시키면서 겹쳐지는 부분의 각 원소의 값을 곱해서 모두 더한 값을 출력으로 하는 연산이다.
2. CNN은 Filter와 이미지의 Convolution으로 이미지의 Feature를 추출해내는 모델이다.
3. PyTorch에서 이미지 데이터의 차원은 (Channel, Height, Width) 순으로 구성되어 있다. (배치 Batch 를 제외한다.)
4. CNN은 parameter를 공유하여 전체 parameter 수를 줄여주기 때문에 overfitting이 일어날 가능성이 DNN보다 더 높다.

- ☐ 1
☐ 2
☐ 3
☒ 4

1

2

3

4

5

6

7

8

9

10

매개변수(parameter)의 갯수가 줄어들수록 overfitting을 줄여줍니다.

Quiz 2 해설

Quiz

정답 Q 2 (단일선택, 10점)

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

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

다음 문구의 진위 여부를 판별하세요.

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

☒ True

☐ False

1

2

3

4

5

6

7

8

9

10

파이토치에서 Convolutional Layer 를 사용하는 방법은 다음과 같습니다.

Conv2d(input_channels, output_channels, kernel_size, stride, padding)

- **input_channels** : 입력 채널 개수
- **output_channels** : 출력 채널 개수
- **kernel_size** : 커널(필터)의 크기
- **stride** : 스트라이드의 크기
- **padding** : 패딩의 크기

Quiz 3 해설

Quiz

정답 Q 3 (단일선택, 10점)

다음과 같이 conv 의 이름으로 convolution layer 필터를 만들어 주고 inputs를 넣어주었다.

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

위 코드에서 inputs의 형태로 보기 중 알맞은 것은?

	A	B	C	D
1	Batch_Size	Height	Width	Channel
2	Batch_Size	Channel	Height	Width
3	Channel	Batch_Size	Height	Width
4	Channel	Batch_Size	Width	Height
5	Channel	Height	Width	Batch_Size

- ☐ 1
☒ 2
☐ 3
☐ 4
☐ 5

1 2 3 4 5 6 7 8 9 10

파이토치에서 Convolutional Layer 의 입력의 형태는 다음과 같습니다.

(Batch_Size, Channel, Height, Width)

- **Batch_Size** : 미니배치 크기
- **Channel** : 채널 크기
- **Height** : 높이
- **Width** : 너비

Quiz 4 해설

Quiz

정답 Q 4 (단일선택, 10점)

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?

- ☐ 1
- ☐ 2
- ☒ 3
- ☐ 4

1 2 3 4 5 6 7 8 9 10

이미지의 입력 크기를 N , 필터(커널)의 크기를 F , 패딩 크기를 p , 스트라이드(stride) 크기를 s 라고 하였을 때, 출력 크기는 다음과 같이 계산됩니다.

$$\frac{N - F + 2p}{s} + 1 = \frac{63 - 7 + 2p}{1} + 1 = 63$$

따라서 입력 이미지와 같은 출력 이미지 크기를 형성하려면 $p=3$ 됩니다.

Quiz 5 해설

Quiz

정답 Q 5 (단일선택, 10점)

다음 Convolution 필터의 output 크기로 알맞은 것은?

```
1 inputs = torch.Tensor(1,1,32,32)
2 conv = torch.nn.Conv2d(1,1,(5,5),stride = 1, padding = 2)
3 output = conv(inputs)
4
5 print(output.size())
```

- ☐ torch.Size([1,1,28,28])
- ☐ torch.Size([1,1,29,29])
- ☐ torch.Size([1,1,30,30])
- ☐ torch.Size([1,1,31,31])
- ☒ torch.Size([1,1,32,32])

1 2 3 4 5 6 7 8 9 10

이미지의 입력 크기를 N , 필터(커널)의 크기를 F , 패딩 크기를 p , 스트라이드(stride) 크기를 s 라고 하였을 때, 출력 크기는 다음과 같이 계산됩니다.

$$\frac{N - F + 2p}{s} + 1 = \frac{32 - 5 + 2 \times 2}{1} + 1 = 32$$

Quiz 6 해설

Quiz

정답 Q 6 (단일선택, 10점)

채널이 16인 32x32 이미지를 filter(kernel) size 2로 max pooling을 적용하였을 때, output 크기로 알맞은 것은?

```
1 inputs = torch.Tensor(1,16,32,32)
2 pool = torch.nn.MaxPool2d(2)
3 output = pool(inputs)
4
5 print(output.size())
```

- ☐ torch.Size([1,8,16,16])
- ☒ torch.Size([1,16,16,16])
- ☐ torch.Size([1,8,32,32])
- ☐ torch.Size([1,16,15,15])

1 2 3 4 5 6 7 8 9 10

최대 풀링(Max Pooling) 도 Convolutional Layer 와 동일하게 계산할 수 있습니다. 단, [torch.nn.MaxPool2d](#)에서는 stride 를 명시하지 않을 경우 필터(커널) 크기와 같게 최대 풀링 연산을 진행합니다. 또한, 기본 패딩값은 0 입니다.

이미지의 입력 크기를 N, 필터(커널)의 크기를 F, 패딩 크기를 p, 스트라이드(stride) 크기를 s 라고 하였을 때, 출력 크기는 다음과 같이 계산됩니다.

$$\frac{N - F + 2p}{s} + 1 = \frac{32 - 2 + 2 \times 0}{2} + 1 = 16$$

조금 더 간단하게 계산하면 이미지 크기를 필터(커널) 크기로 나누어주어 16x16이 됩니다.

Quiz 7 해설

Quiz

정답 Q 7 (단일선택, 10점)

다음 코드는 2-layer CNN Model을 구현한 것이다.

```
1 # CNN Model (2 conv layers)
2 class CNN(torch.nn.Module):
3
4     def __init__(self):
5         super(CNN, self).__init__()
6
7         self.layer1 = torch.nn.Sequential(
8             torch.nn.Conv2d(1, 32, kernel_size=3, stride=1, padding=1),
9             torch.nn.ReLU(),
10            torch.nn.MaxPool2d(kernel_size=2, stride=2))
11
12        self.layer2 = torch.nn.Sequential(
13            torch.nn.Conv2d(32, 64, kernel_size=3, stride=1, padding=1),
14            torch.nn.ReLU(),
15            torch.nn.MaxPool2d(kernel_size=2, stride=2))
16
17        self.fc = torch.nn.Linear(7 * 7 * 64, 10, bias=True)
18        torch.nn.init.xavier_uniform_(self.fc.weight)
19
20    def forward(self, x):
21        out = self.layer1(x)
22        out = self.layer2(out)
23        out = out.view(out.size(0), -1)
24        out = self.fc(out)
25        return out
```

다음 문구의 진위 여부를 판별하세요.

"여기서 out = out.view(out.size(0), -1)의 역할은 다음 Fully Connected Layer에 넣어주기 위해 1줄로 Flatten 해주어 shape을 맞추어 주는 것이다."

- ☒ True
☐ False

1 2 3 4 5 6 7 8 9 10

파이토치 텐서 내장함수 view 에 대한 설명을 참고하세요!

<https://pytorch.org/docs/stable/tensors.html#torch.Tensor.view>

Quiz 8 해설

Quiz

정답 Q 8 (단일선택, 10점)

다음 문구의 진위 여부를 판별하세요.

"다음과 같이 학습해야 할 양이 많아 GPU를 사용해야할 때, 먼저 사용가능한 지 확인 후 모델에만 device를 적용시켜주면 된다."

```
1 device = 'cuda' if torch.cuda.is_available() else 'cpu'
2 model = CNN().to(device)
```

☐ True

☒ False

1 2 3 4 5 6 7 8 9 10

다른 네트워크를 통과하는 입력 데이터, 손실 함수에도 모두 적용해야 합니다.

Quiz 9 해설

Quiz

정답 Q 9 (단일선택, 10점)

다음 문구의 진위 여부를 판별하세요.

"VGGNet 은 Oxford Visual Geometry Group 에서 만든 모델의 깊이와 구조에 변화를 주며 만든 Network로, convolution layer에서 3x3 convolution 필터와 stride=1, padding=1만을 이용한 특징이 있다."

☒ True

☐ False

1 2 3 4 5 6 7 8 9 10

CNN에 대한 활용예시 중 하나인 VGG Net 은 Lab-10-5 Advance CNN(VGG) 혹은 [모두를 위한 딥러닝 시즌 1 강좌 영상](#)을 찾아보세요!

Quiz 10 해설

Quiz

정답 Q 10 (단일선택, 10점)

다음 문구의 진위 여부를 판별하세요.

"ResNet(Residual Network)은 'Shortcut connections' 방법을 통한 residual learning으로 Layer가 깊어질수록 gradient vanishing/exploding 문제가 발생하는 것을 해결하였다."

☒ True

☐ False

1 2 3 4 5 6 7 8 9 10

CNN에 대한 활용예시 중 하나인 ResNet 은 Lab-10-6-1,2 Advanced CNN(RESNET-1, 2) 혹은 [모두를 위한 딥러닝 시즌 1 강좌 영상](#)을 찾아보세요!

이제 프로젝트를 진행하세요!