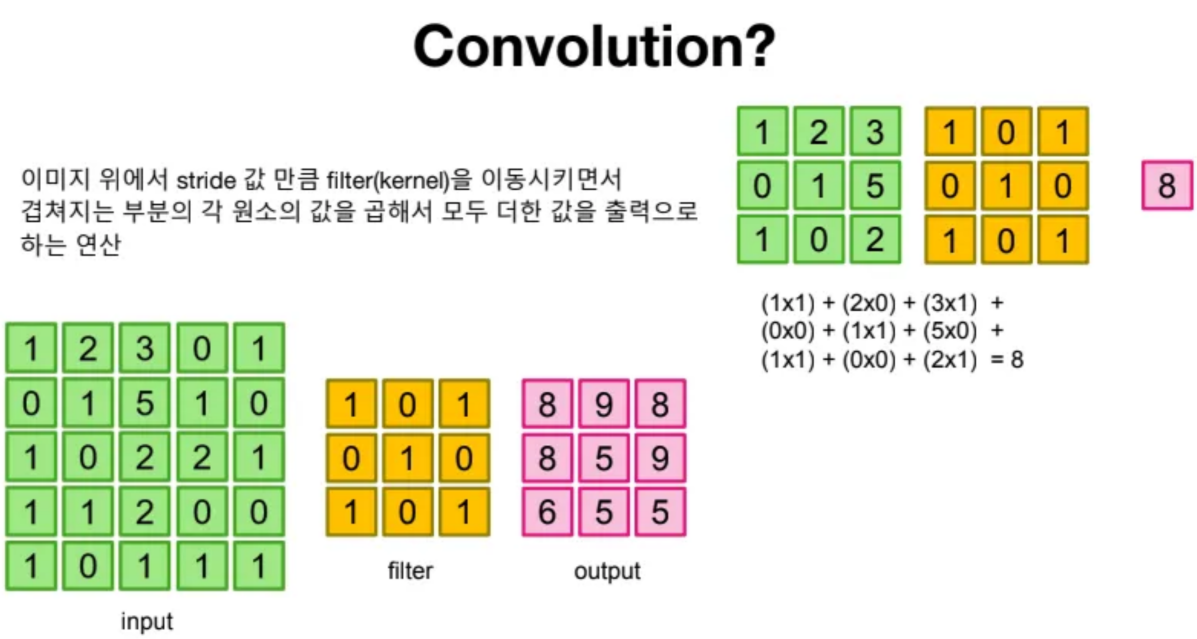
exam 개인학습

Lec 17

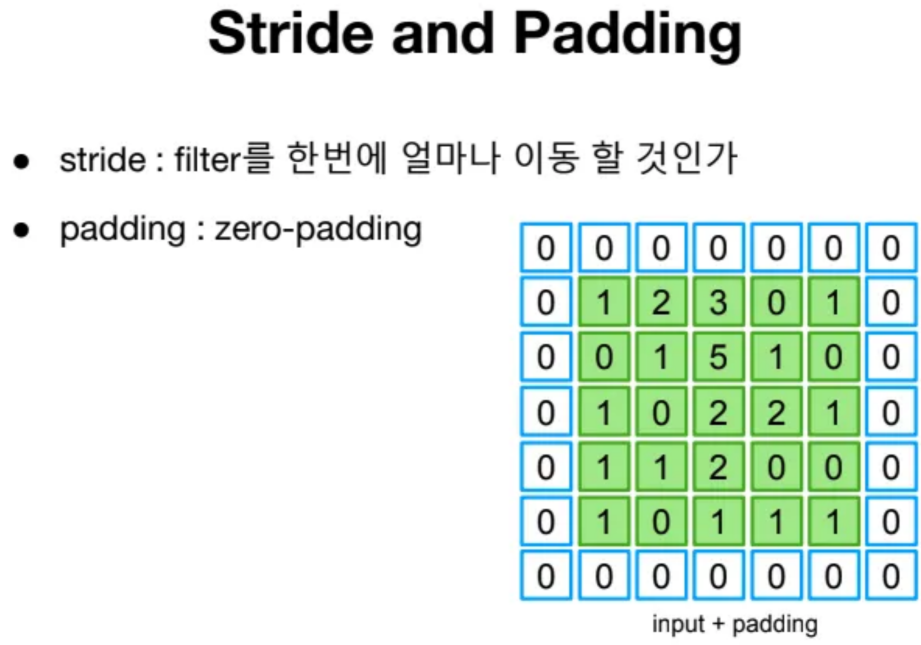
Type something...

01) Convolution

01) Convolution



- Convolution은 filter를 stride 값만큼 이동시켜 각 원소의 값을 곱해서 모두 더한 값을 만드는 연산이다.



- stride: filter를 얼마나 이동시킬 것인가, padding : 테두리에 zero를 얼만큼 넣을 것인가(이미지 손실을 방지하기 위함임)

Conv2d

CLASS 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*) [SOURCE]

Applies a 2D convolution over an input signal composed of several input planes.

In the simplest case, the output value of the layer with input size (N, C_{in}, H, W) and output $(N, C_{\text{out}}, H_{\text{out}}, W_{\text{out}})$ can be precisely described as:

$$\text{out}(N_i, C_{\text{out}_j}) = \text{bias}(C_{\text{out}_j}) + \sum_{k=0}^{C_{\text{in}}-1} \text{weight}(C_{\text{out}_j}, k) \star \text{input}(N_i, k)$$

- `torch.nn.Conv2d(in_channels, out_channels, kernel_size, stride, padding, .....)`

입력의 형태

- input type : torch.Tensor
- input shape : (N x C x H x W)

  (batch_size, channel, height, width)

- input type = `torch.Tensor`
- input shape = (N x C x H x W): (batch_size, channel, height, width)

Convolution의 output 크기

$$Output\ size = \frac{input\ size - filter\ size + (2 * padding)}{Stride} + 1$$

예제 1)
input image size : 227 x 227
filter size = 11x11
stride = 4
padding = 0
output image size = ?

예제 2)
input image size : 64 x 64
filter size = 7x7
stride = 2
padding = 0
output image size = ?

예제 3)
input image size : 32 x 32
filter size = 5x5
stride = 1
padding = 2
output image size = ?

예제 4)
input image size : 32 x 64
filter size = 5x5
stride = 1
padding = 0
output image size = ?

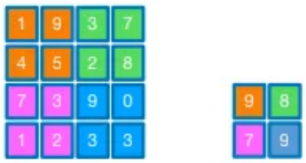
예제 5)
input image size : 64 x 32
filter size = 3x3
stride = 1
padding = 1
output image size = ?

$$output\ size = \frac{input_size - filter\ size + 2 * (padding)}{Stride} + 1$$

- 만약 나눗셈 과정에서 28.5 이런식으로 남았으면 버려주고 28만 사용한다. **(내림)**

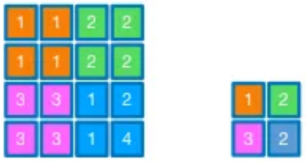
Pooling

- Max Pooling



max pooling

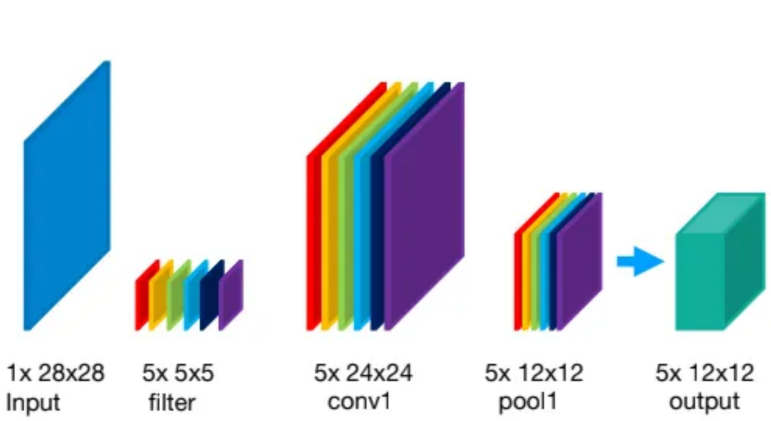
- Average Pooling



Average pooling

- pooling은 구역별로 값을 추출하는 과정으로 max pooling은 최대값을 추출, average pooling은 평균값을 추출한다.

CNN implementation




```
input = torch.Tensor(1,1,28,28)
conv1=nn.Conv2d(1,5,5)
pool = nn.MaxPool2d(2)
out = conv1(input)
out2 =pool(out)
out.size()
out2.size()
```

- batch_size를 제외하고 channel * height *stride만 고려한 필터이다.
- Input: 1 x 28 x 28
- filter: 5 x 5 x 5 `nn.Conv2d(1, 5, 5)` **:input_channel = 1, output_channel = 5, kernel_size = 5**
- conv1: 5 x 24 x 24
- `pool = nn.MaxPool2d(2)` : **5x12x12 output**