

[21차시 Convolution]

참조사이트: <https://dataplay.tistory.com/29>

Transposed Convolution, Dilated Convolution

1. Transposed Convolution

- Transposed Convolution은 이름 그대로 반대로 작동하는 Convolution
- 보통 Convolution은 padding이 없을시 feature를 뽑아내면서 이미지의 크기는 작아지게 된다. 하지만 Transposed Conv는 거꾸로 이미지의 크기를 더 커지게 만든다.

2. Dilated Convolution

- Dilated Convolution은 Convolution filter를 적용할 때, n칸 간격 만큼 건너 뛰며 filter를 적용하는 것

4. Transposed Convolution

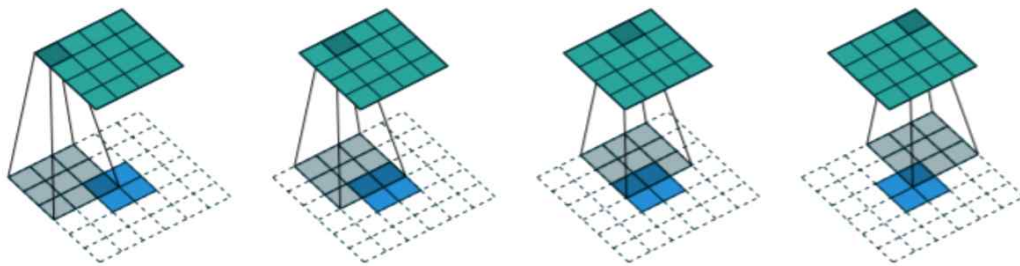


Figure 4.1: The transpose of convolving a 3×3 kernel over a 4×4 input using unit strides (i.e., $i = 4$, $k = 3$, $s = 1$ and $p = 0$). It is equivalent to convolving a 3×3 kernel over a 2×2 input padded with a 2×2 border of zeros using unit strides (i.e., $i' = 2$, $k' = k$, $s' = 1$ and $p' = 2$).

- Stride, Padding, filter size, dilations 등의 파라미터의 조정으로써 원하는 크기의 이미지를 만들어낼 수 있다.
- 위 사진은 2×2 이미지에 Transposed Conv를 적용시켜서 4×4 이미지를 만든 것/
 3×3 filter, 그리고 padding을 이용해서 만든 것/ Upsampling을 수행하는 Layer를 만든 것이다.
- 이러한 Transposed Convolution이 사용 되는 모델의 종류는 다양한데, 예를 들어 CNN을 사용한 Encoder-Decoder 구조의 Autoencoder에서 Encoder에서 Pooling 등을 통해 이미지를 축소시키면서 데이터를 압축했다면, 그 데이터를 다시 원래의 이미지로 복원하기 위해 이미지를 크게 만들어야 한다. 이럴 때 Transposed Convolution을 사용
- 이미지의 해상도를 높이하고자 하는 경우(Super resolution이라고도 불리는 기법)에는 input 이미지보다 output 이미지의 크기를 커지게 만듦으로써 가능한데, 이럴 때도 Transposed Convolution이 사용된다.

- 이미지 크기를 크게 하는 방법중 보간 이라는 방법이 있는데(보간이란: 데이터를 크게 만들 때 필요한 정보들을 수학적인 알고리즘을 바탕으로 채우는 것) Transposed Convolution은 보간 방식 비해 얻을 수 있는 장점이 더 있다 즉, 딥러닝 방식으로 파라미터들을 학습하는 것이다. 학습이 잘 된다면 기존 보간 방식에 비해 정확한 정보를 얻을 수 있습니다.
- Deconvolution도 보기에는 Transposed Conv와 같지않은데, Deconvolution은 Convolution의 역산으로 Convolution layer를 거치기 전의 input을 추정하는 과정이다.

5. Dilated Convolution

- Receptive Field의 개념

Receptive Field는 filter가 한 번에 보는 영역을 뜻함.

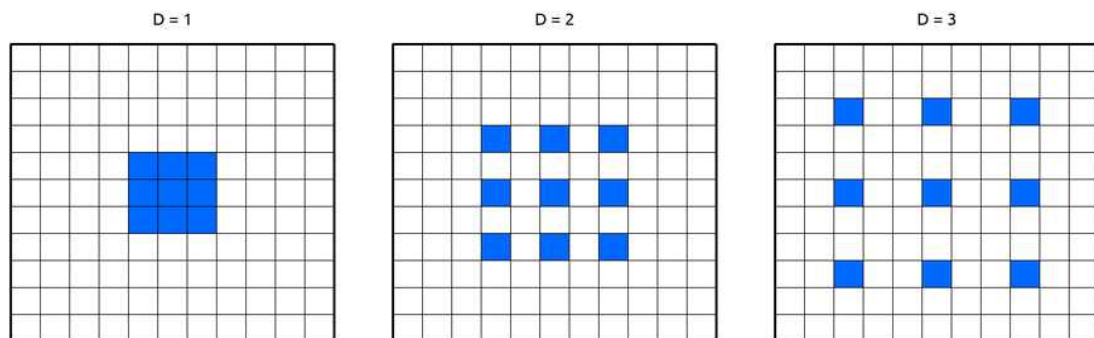
- 일반적인 3x3 filter size는 receptive field가 3x3 이미지 이다.

그리고 이러한 layer를 두개, 세개 쌓으면 receptive field가 5x5, 7x7 이런 식으로 늘어나게 된다. Receptive field가 늘어난다는 것은 output을 계산할 때 사용하는 정보의 양이 많다는 것이다.

- 정보의 양이 늘어나면, 성능이 좋아질 확률도 높아지지만, 학습해야할 양이 많아서 연산 량이 증가하게 되는 단점도 있다.

이 Receptive field를 높이기 위해서 filter의 크기를 키우거나, layer를 늘릴 수 있다. 또는 pooling 등을 사용하는 것도 receptive field를 높일 수 있다. Pooling의 경우 연산 량 까지 감소할 수 있지만 정보의 손실을 가져올 수도 있다.

- Dilated Convolution은 Receptive field를 크게 만들어서 커버하는 영역을 크게 만들면서, 연산 량의 증가는 가져오지 않는 효과적인 방법이다.



=> Dilation Factor를 1, 2, 3으로 각각 설정하고, Filter size를 3x3으로 설정.

- Dilation의 크기가 늘어날수록, filter가 추출해내는 input의 간격이 멀어지는 것을 볼 수 있다. D=1 일 때는 receptive field가 3x3 이지만, D=2 일 때는 receptive field가 5x5 이다. Receptive Field는 늘어났는데, 연산 량은 증가는 없다.

- filter가 sparse하게 feature를 추출함으로써 일어나는 손실도 있을 수 있겠으나, Receptive Field의 크기가 중요한 경우에는 아주 잘 작동한다고 알려져 있다.

예를 들어 Audio의 경우에 1초당 보통 16000~22500개의 sample이 존재하는데, Dilated Convolution 방법을 통해 몇 개의 layer를 통해서 많은 sample을 커버 할 수 있다.