Convolution과 Transposed Convolution: 어디서 온 이름인가?

2012년, 8개의 층을 가진 깊은 인공신경망으로 이미지 인식 대회를 제패한 AlexNet이 나오면서 딥러닝 붐이 시작되었다. AlexNet은 많은 층을 가진 convolutional neural network (CNN)를 성공적으로 구현한 첫 사례라고 볼 수 있고, 이후 이미지 처리를 포함한 매우 다양한 분야에서 CNN을 성공적으로 사용한 사례들이 많아지면서 CNN은 가장 많이 쓰이는 신경망 구조 중 하나가 되었다.

Convolution이라는 용어는 CNN 이전에도 이미지 처리 분야에서, 또 그 이전에도 수학, 특히 해석학 분야에서 널리 사용된 개념으로 이름의 유래에 대해 한 번쯤 살펴볼 필요가 있다. 다만, CNN의 convolution은 기존에 사용되던 같은 이름의 개념과 다소 다른 점이 있기 때문에, 이 둘의 관계를 면밀히 짚고 넘어가는게 좋다.

Convolution보다 더 큰 오해를 불러일으킬 수 있는 용어가 바로 transposed convolution이다. 이는 deconvolution, backwards convolution 등의 유의어를 갖고 있는데, 어떤 면에서는 오개념을 심어줄 수 있는 표현들이기 때문에 주의가 필요하다. 왜 수학 용어인 transpose가 붙게 되었는지 그 과정을 안다면 이러한 오해를 해결할 수 있을 것이다.

이 글에서는 convolution과 transposed convolution의 이름이 어디서 유래하였는지 살펴보고 원래의 뜻과 달라진 부분, 그리고 오개념을 살 수 있는 표현들에 대해 살펴본다. Convolution과 Transposed Convolution의 연산 과정에 대해서 독자들이 알고 있다고 가정하겠다.

Convolution

Convolution이란?

Convolution은 국소적인 정보를 가져오는 데에 효과적인 신경망 구조로, 평행이동에 대한 등변성(shift-equivariance)을 가지고 있어 이미지 처리를 비롯한 다양한 작업에서 사용된다. 데이터의 차원을 낮추는 convolution은 데이터의 정보를 압축하여 전달하는 데에 효과적이다. 이러한 점에서 높은 차원의 입력 데이터에서부터 낮은 차원의 출력 데이터를 뽑을 때 convolution이 쓰일 수 있다. 이미지 분류가 대표적인 예시가 될 것이다.

수학에서의 Convolution

18세기부터 수학자들이 연구하던 개념 중 convolution이라는 이름이 붙은 것이 있다. 두 함수 f와 g에 대하여,

$$(f * g)(x) = \int_{-\infty}^{\infty} f(t)g(x-t)\,dt$$

로 정의되는 함수 $f*g$를 f와 g의 convolution이라고 부른다. 해석학에서 출발한 이 개념은 이미지 처리, 신호 처리를 비롯해 다양한 과학적, 공학적 응용을 가지게 된다. 위의 수학적 정의가 이미지 처리의 convolution과 어떤 관계를 가지는지 알아보자.
먼저, 연속적인 convolution의 이산적인 버전은 무엇일지 생각해보자. 적분을 합으로 교체하면,

$$(f * g)(x) = \sum_{t=-\infty}^{\infty} f(t)g(x-t)$$

와 같은 식을 얻는다. 이를 discrete convolution이라고 부른다.
이제 모든 준비는 끝났다. 수학에서의 convolution과 CNN에서의 convolution이 어떻게 연결되는지 살펴보자. 수학에서는 convolution 정의식에서 볼 수 있듯이, $f(t)$와 곱해지는 $g(x-t)$와 같은 함수항이 있을 때 이를 kernel이라고 부른다. 이 용어가 그대로 CNN에서도 사용되는데, 문자를 바꿔 convolution되는 대상 함수를 input에서 따온 $I(x)$라 하고 kernel 함수를 $K(x)$라 하면,

$$(I * K)(x) = \sum_t I(t)K(x-t) = \sum_t I(x-t)K(t)$$

이 된다(정의로부터 $f*g = g*f$임을 확인할 수 있다). 이미지에서의 convolution 연산과 더 가까워졌으나, 한 가지 문제점이 있다. 부호가 뒤바뀌는(kernel flipping) 문제인데, 만약 부호를 바꾸어 $\sum_t I(x+t)K(t)$와 같이 쓰면 CNN에서 쓰이는 convolution이 된다. 따라서, CNN의 convolution은 원래 수학에서 정의된 convolution과는 다른 것이다. 이는 사실 cross-correlation이라는 개념과 일치하는데, 다음과 같이 정의되는 개념이다:

$$(f \star g)(x) = \int \overline{f(t)}g(x+t)\,dt$$

$\overline{f(t)}$는 켤레 복소수를 의미한다. 우리는 실함수를 다루고 있으므로 무시할 수 있다. 즉, CNN의 convolution은 엄밀히 말하자면 convolution이 아닌 cross-correlation인 셈이다!
cross-correlation $\star$을 이용하여 수식으로 나타내면,

$$(I \star K)(x) = \sum_t I(x+t)K(t)$$

이 CNN의 convolution 연산이 된다. 이는 1차원 convolution을 잘 표현하는 식이 되는데, 이미지의 경우 2차원 자료형이므로 다음과 같이 쓰면 더 자연스러울 것이다.

$$(I \star K)(x, y) = \sum_s \sum_t I(x + s, y + t)K(s, t)$$

이제 CNN의 convolution이 어디서 유래한 것인지, 그리고 원래 뜻과 달라진 점이 무엇인지 알게 되었다!

Transposed Convolution

Upsampling의 필요성

딥러닝을 사용할 수 있는 작업 중 upsampling layer가 필요한 경우가 있다. 입력 데이터보다 더 높은 차원의 데이터를 출력하는 해상도 업스케일링 작업이나, 입력 데이터와 동등한 차원의 데이터를 출력하는 오디오/이미지 노이즈 제거 작업 등이 대표적인 예시이다.
Transposed convolution은 데이터의 국소적 정보를 반영하면서 학습 가능한 매개변수를 이용하는 upsampling 방법이다. 이 방법이 사용된 대표적인 예시로는 2015년 발표된 U-Net이 있다. 반복되는 convolution으로 정보를 압축해 저차원의 latent vector에 저장하고 이를 다시 반복되는 transposed convolution으로 입력과 동일한 차원으로 복구시킨다.

Deconvolution, Fractionally strided convolution? 오해를 사는 이름들

Transposed convolution은 행렬 transpose와의 연관성이 밝혀지기 전에는 deconvolution, upconvolution, backwards strided convolution, fractionally strided convolution 등 다양한 이름으로 불렸다. 지금은 transposed convolution이라는 명칭이 제일 많이 쓰이는데, 이 연산의 특징을 가장 정확하게 설명하는 명칭이기 때문이다.
특히 deconvolution은 위에서 설명한 수학에서의 convolution의 역연산으로 가지는 의미가 있는 용어이므로 오해를 불러일으킬 수 있다. Deconvolution은 원래 어떤 의미를 가지는 용어인가? 일반적으로, 주어진 함수 h에 대하여 $f * g = h$를 만족하는 f를 찾아내는 작업을 deconvolution이라고 부른다. 이는 transposed convolution과는 완전히 다른 의미이므로, 사용에 주의가 필요하다.

1d convolution의 행렬 표현

그렇다면 왜 transpose라는 명칭이 적절한 것인가? 이를 위해서는 1d convolution의 행렬 표현을 이해해야 한다. 1d conv의 인풋을 $I \in \mathbb{R}^n$, kernel을 $K \in \mathbb{R}^p$라 하고 kernel size를 p라고 하자.

그러면

$$I * K = \begin{pmatrix} k_1 & k_2 & \cdots & k_p & 0 & \cdots & \cdots & 0 \\ 0 & k_1 & k_2 & \cdots & k_p & 0 & \cdots & 0 \\ \vdots & & & & & & & \vdots \\ 0 & 0 & 0 & \cdots & k_1 & k_2 & \cdots & k_p \end{pmatrix} \begin{pmatrix} i_1 \\ i_2 \\ \vdots \\ i_n \end{pmatrix}$$

임을 알 수 있다. 위 식의 $(n-p+1) \times n$ 행렬을 W라 하자. $X \in \mathbb{R}^{n-p+1}$에 대하여

$$W^\top X = \begin{pmatrix} k_1 & 0 & \cdots & 0 \\ k_2 & k_1 & \cdots & 0 \\ \vdots & & & \vdots \\ 0 & 0 & \cdots 0 & k_p \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n-p+1} \end{pmatrix} = \begin{pmatrix} k_1 x_1 \\ k_2 x_1 \\ \vdots \\ k_p x_1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ k_1 x_2 \\ \vdots \\ k_{p-1} x_2 \\ k_p x_2 \\ \vdots \\ 0 \end{pmatrix} + \cdots + \begin{pmatrix} 0 \\ 0 \\ \\ \vdots \\ \\ k_{p-1} x_{n-p+1} \\ k_p x_{n-p+1} \end{pmatrix}$$

은 정확히 1d transposed convolution이 된다.

연습문제: 2d convolution의 행렬 표현과 transpose

2d convolution 역시 1d와 마찬가지로 행렬 표현이 가능하며, 이를 transpose하면 transposed convolution을 얻을 수 있다. 간단한 예시로,

$$I = \begin{pmatrix} i_1 & i_2 & i_3 \\ i_4 & i_5 & i_6 \\ i_7 & i_8 & i_9 \end{pmatrix}, \quad K = \begin{pmatrix} k_1 & k_2 \\ k_3 & k_4 \end{pmatrix}$$

에 대하여 $I * K = W(i_1 \dots i_9)^\top$를 만족하는 행렬 $W \in \mathbb{R}^{4 \times 9}$의 각 원소를 k_j $(1 \le j \le 4)$로 나타내어 보자. 그리고 $X \in \mathbb{R}^4$에 대하여 $W^\top X$을 구해보면 2d transposed convolution이 됨을 확인할 수 있다!

마치며

딥러닝 분야에서 널리 쓰이는 convolution과 transposed convolution의 명칭 유래와 흔한 오개념에 대해 알아보았다. 딥러닝 분야에서는 혼용되는 용어가 많아서 소통할 때 주의가 필요하다. CNN의 Convolution은 수학에서의 convolution 및 cross-correlation과 어떤 관계인지, transposed convolution은 deconvolution 및 행렬 transpose와 어떤 관계인지 알 필요가 있다. 용어에 담긴 의미를 살펴보면서 수학이 딥러닝에 어떻게 쓰이는지 이해해보자.