

Discussion: 이번 단원에서는 CNN, Convolutional Neural Network에 대해 소개하고, 기존까지 배웠던 MLP방식과 이를 비교하여 이미지 데이터 처리에서 CNN이 왜 더 적합한지, 그리고 CNN이 이미지를 어떻게 처리하는지에 대한 설명을 진행하였다. 지금까지 배웠던 MLP방식은 Table Data를 다루기 적합했지만, 고해상도의 이미지와 같이 고차원적인 Data에 대해서는 사용하기 적합하지 않다는 단점이 있었다. 아무리 hidden layer의 차원을 줄이려 해도, 엄청난 양의 parameter를 학습해야 하므로 현재의 컴퓨팅 자원으로는 매우 비효율적인 방법이다. 다행히, 이미지가 가지고 있는 구조적 특성 덕에, 다른 방법을 생각해 낼 수 있었는데 그것이 바로 합성곱을 이용한 CNN방식이다.

CNN은 이미지가 가진 구조적 특성 - translation invariance / locality를 활용하여 효율적으로 학습할 수 있도록 한다. (사실 나의 배경 지식 및 현재 상태의 이해력의 한계로 완전히 이해하지 못하지만, 내가 이해한 수준에서 설명하면) Translation invariance는, 이미지 내의 물체가 다른 위치에 위치하더라도, 동일한 물체는 동일하게 인식되는 특성이다. 예를 들어, 다양한 사진 속 "비행기"를 찾는 상황에서, 비행기는 이미지 속 상단에 위치할 수도, 하단에 위치할 수도, 좌/우에 위치할 수도 있지만, 비행기의 특징 자체는 동일하다는 것이다. 이는 CNN이 동일한 patch의 이미지를 다른 위치에서 발견해도 동일하게 반응할 수 있도록 한다. Locality는, 이미지 내에서 정보를 처리할 때 국소적인 영역에 집중하는 것을 뜻한다. 처음 단계에서 이미지의 작은 부분만을 분석하고(local filter 영역), 이후 단계에서 정보를 종합하여 전체 이미지를 분석하는 것이다.

그렇다면 CNN은 어떻게 이러한 것을 가능하게 할까? -> 바로 수업시간에도 배웠던 convolution, 합성곱 연산을 통해 파라미터 수를 줄이며 학습 능력을 유지하도록 도와주었다. 합성곱이란 한 함수를 뒤집고, 다른 함수와 겹쳐서 측정하는 것이다. 수학적으로는,  $(f*g)(x) = \int f(z)g(x-z)dz$ 으로 될 수 있고, 뭐 이제 실제 컴퓨터에서는 이산적이므로 시그마로 바꾸거나 2차원 텐서를 위해 변수가 여러개인 함수를 한다던가 그런 식이다. 또한 CNN은 channel이란 것을 활용하는데, 이는 단순한 흑백 픽셀 값을 넘어 RGB 값도 인식에 반영될 수 있게 하는 것이다.

이번 단원에는 Code가 없으므로 exercise 문제들 중 하나를 진행해 보려고 한다.

6. 합성곱  $f*g$ 가 symmetric, 즉  $f*g = g*f$ 임을 증명. 수식 기호가 없어 일단 이렇게 진행한다.  $(f*g)(t) = \int f(t-r)g(r)dr$   $u = t-r$ 로 치환하면,  $du = -dr$   $(f*g)(t) = \int f(t-u)g(u)du = \int g(u)f(t-u)du = (g*f)(t)$  간단한 치환적분을 통해 증명하였다.

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js