Lecture 4

경사하강법: 저번 시간에 얘기 했던

수치적 경사 Numerical Gradient(어림값, 작성 쉽다), 분석적 경사 Analytic Gradient(정확함, 실수하기 쉽다).

- ➔ 분석적 경사(계산 그래프 Computational Graph)에 대한 이야기

Convolution network, AlexNet:

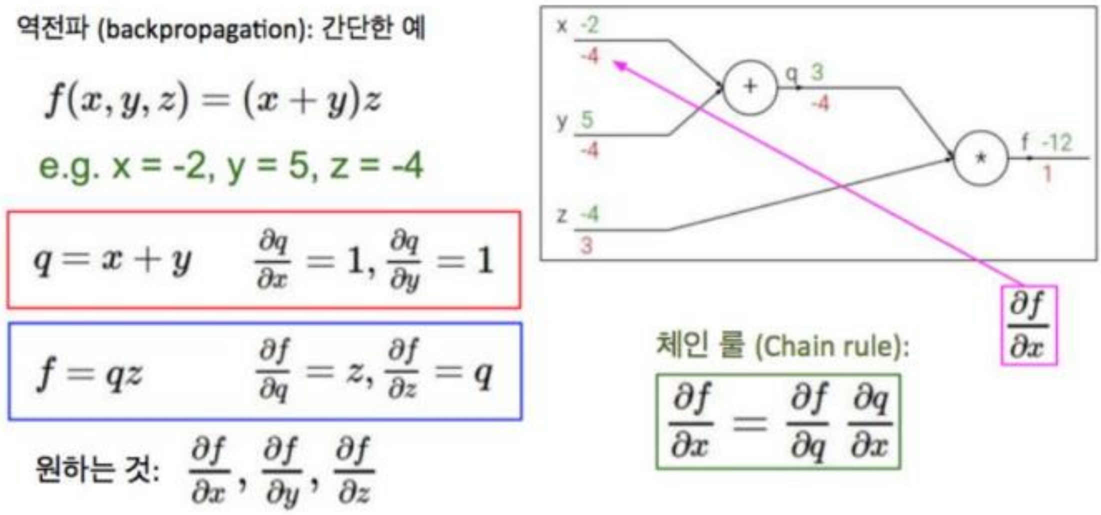
image(input) → weights(가중치) → loss(손실)

Neural Tuning Machine:

이미지 → 손실(규모가 엄청남) → 딥러닝 모델

Forward Pass(FP): input이 마지막 단계에 어느 영향을 끼치는 가.

backpropagation(역전파): FP 값을 구하기 위해, 역순으로 계산하는 것.



chain rule: dq/ dx 는 원래 구하고자 하는 gradient를 위한 값이므로, 이를 local gradient

df/dq는 global gradient라고 함.

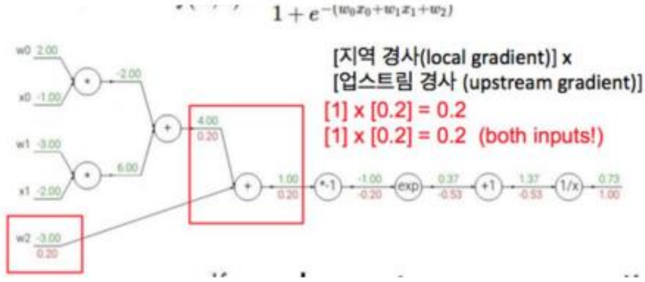
➔ Forward Pass(FP) 시에는 local gradient 값을 바로 구해서 메모리 저장 가능.

Backpropagation(역전파) 시에는 local gradient와 global gradient값을 곱해줘서 gradient를 구할 수 있다.

한마디로 역전파시에는, chain rule이 발생한다.

input이 더 많을 때:

$$f(w,x)=\frac{1}{1+e^{-(w_0x_0+w_1x_1+w_2)}}$$



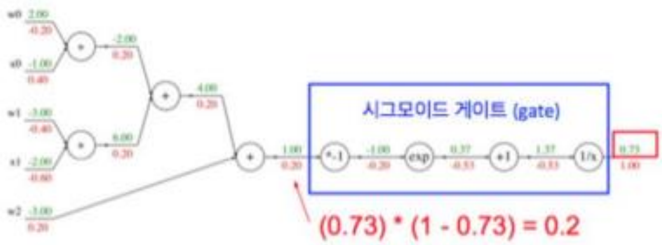
+ 연산은, gradient distributor. 그냥 동일한 연산을 전파해준다.

위에 식은 시그모이드 함수랑 형태가 비슷한데,

$$f(w,x)=\frac{1}{1+e^{-(w_0x_0+w_1x_1+w_2)}} \qquad \boxed{\sigma(x)=\frac{1}{1+e^{-x}}}$$

시그모이드 함수
(sigmoid function)

$$\frac{d\sigma(x)}{dx}=\frac{e^{-x}}{(1+e^{-x})^2}=\left(\frac{1+e^{-x}-1}{1+e^{-x}}\right)\left(\frac{1}{1+e^{-x}}\right)=(1-\sigma(x))\sigma(x)$$



이런식으로 간단하게도 계산이 가능하다.

Patterns in Backpropagation:

add gate: gradient distributor(경사 분배기)

max gate: gradient router(경사 라우터)

mul gate: gradient switcher(경사 스위처)

Vectorized code(벡터화된 코드)경사

$$\frac{\partial L}{\partial x} = \boxed{\frac{\partial f}{\partial x}} \frac{\partial L}{\partial f}$$

➔ 가운데 파란색이 Jacobian matrix

만약 4096 차원의 입력벡터로 넣고, 4096이 출력 된다면, 이 중간 matrix값은 [4096 x 4096] 크기가 됨.

신경망:

기존에는, f = Wx라고 간단히,

이제는 2-layer nn을 다룰 것이기 때문에, f = W2max(0, W1x)로

3-layer nn은 f = W3max(0, w2max(0,W1x))

activation function 중에서, sigmoid function을 가장 많이 사용한 이유가, 0~1 사이의 값을 갖기 때문.

활성화 함수: