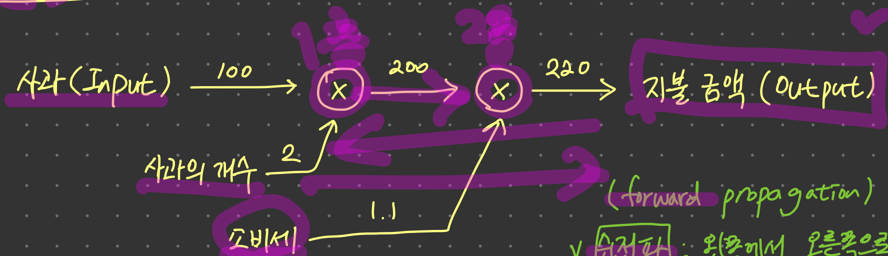
Chapter 5. 오차역전파법

수치미분 < 단순, 구현 easy
계산시간 ↑ ⇒ 가중치 매개변수(w) 기울기 효율적 계산하기 위해 "오차역전파법"(backpropagation) 사용

1. 계산 그래프

· 계산 그래프의 예시.

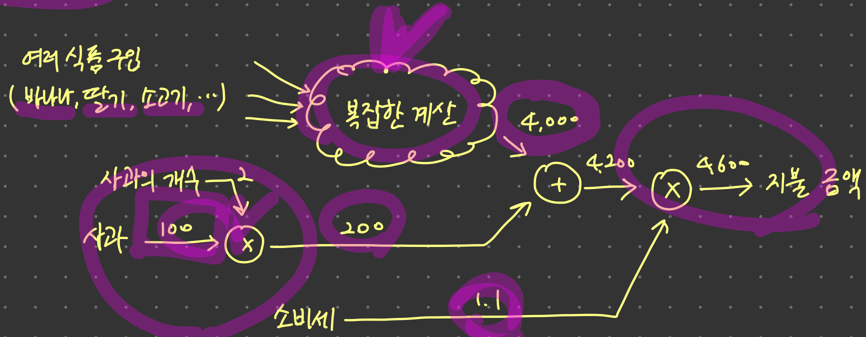
"현빈 군은 슈퍼에서 1개에 100원인 사과를 2개 샀습니다. 이 때 지불 금액을 구하시오. 단, 소비세가 10% 부여됩니다."



(forward propagation)
√ 순전파: 왼쪽에서 오른쪽으로 진행
√ 역전파: 반대로 진행
(back propagation)

* 계산 그래프의 중요한 특징 : "국소적 계산"

ex) 사과 2개를 포함해 여러 식품을 구입하는 경우



√ 전체에서 어떤 일이 벌어지든 상관없이 자신과 관계된 정보만으로 결과를 출력할 수 있다.

√ 중간 계산 결과를 보관할 수 있다.

✓ 역전파를 통해 미분을 효과적으로 구할 수 있다.

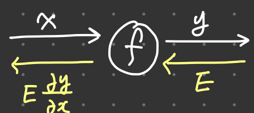
ex) 역전파에 의한 미분값의 전달

⇒ '사과' 가격에 대한 '지불 금액'의 미분 값은 2.2

⇒ 중간에 구한 미분 결과 공유할 수 있어서 다수의 미분 효율적 계산 가능

2. 연쇄 법칙

• 계산 그래프의 역전파 : 순방향과는 반대 방향으로 국소적 미분을 곱한다.

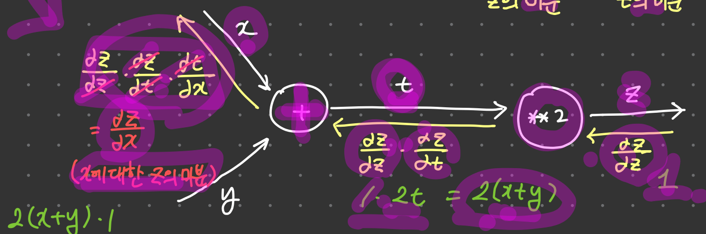


• 합성함수의 미분 ⇐ 합성함수를 구성하는 각 함수의 미분의 곱 (연쇄법칙)

ex) $\begin{cases} z = t^2 \\ t = x + y \end{cases}$

$$\frac{\partial z}{\partial x} = \frac{\partial z}{\partial t} \cdot \frac{\partial t}{\partial x} = 2t \cdot 1 = 2(x+y)$$

t에 대한 z의 미분 / x에 대한 t의 미분 / 두 미분의 곱

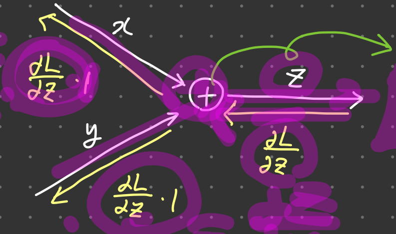


⇒ 역전파가 하는 일 = 연쇄 법칙의 원리

3. 역전파

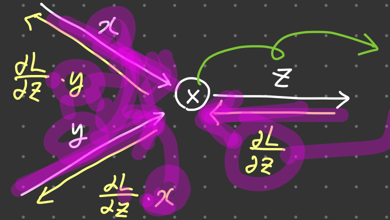
<덧셈노드의 역전파> $z = x + y$ $\begin{cases} \frac{\partial z}{\partial x} = 1 \\ \frac{\partial z}{\partial y} = 1 \end{cases}$

$z = 2x + y = 2 \cdot x + y$



덧셈노드의 역전파는 입력 신호를
그대로 다음노드에 출력한다.

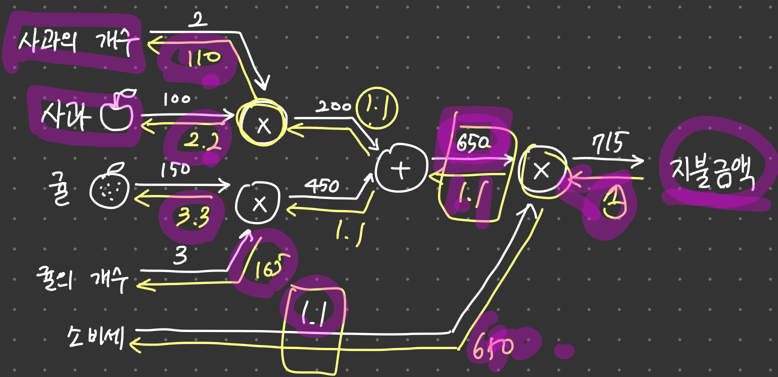
<곱셈노드의 역전파> $z = xy$ $\begin{cases} \frac{\partial z}{\partial x} = y \\ \frac{\partial z}{\partial y} = x \end{cases}$



다음 노드 = 이전노드 X 순전파 때의 입력값들을 '서로 바꾼 값'

⇒ 곱셈 노드를 구현할 때에는
순전파의 입력 신호를 변수에 저장해둔다.

· 사과와 귤 쇼핑의 역전파 예.



⇒ 4. 계층 구현하기

- 곱셈노드 'Mul Layer'
- 덧셈노드 'Add Layer'

5. 활성화함수 계층 구현하기

☆ 계산 그래프를 신경망에 적용하기!

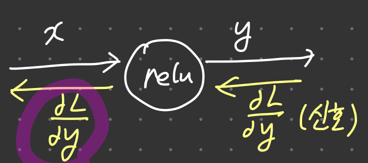
1) ReLU 계층

- 활성화함수 ReLU : $y = \begin{cases} x & (x>0) \\ 0 & (x \le 0) \end{cases} \rightarrow \frac{\partial y}{\partial x} = \begin{cases} 1 & (x>0) \\ 0 & (x \le 0) \end{cases}$

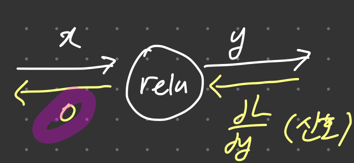
∴ 순전파 때 입력값 x가 0보다 크면 신호값 그대로 흘려보냄,

" 0보다 작거나 같으면 신호를 보내지 않음.

$\boxed{x>0}$



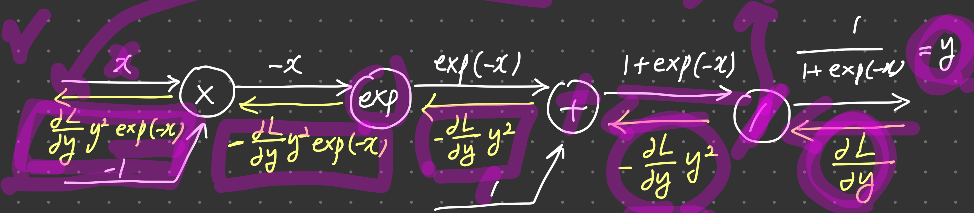
$\boxed{x \le 0}$

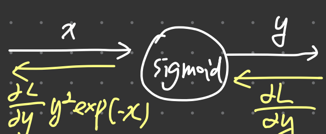


"like 스위치"

2) Sigmoid 계층

- 시그모이드 함수 : $y = \frac{1}{1+\exp(-x)}$





$$* \ \frac{\partial L}{\partial y} y^2 \exp(-x) = \frac{\partial L}{\partial y} \frac{1}{(1+\exp(-x))^2} \exp(-x)$$

$$= \frac{\partial L}{\partial y} \frac{1}{1+\exp(-x)} \frac{\exp(-x)}{1+\exp(-x)}$$

$$= \boxed{\frac{\partial L}{\partial y} y(1-y)}$$

✓ 순전파의 출력(y)만으로 계산 가능!

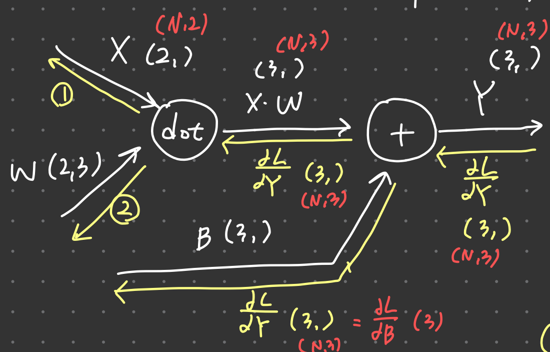
6. Affine / Softmax 계층 구현하기

· Affine 계층 (어파인 계층) : 신경망의 순전파 때 수행하는 행렬의 곱

ex) $\underset{(2,\ )}{X} \cdot \underset{(2,3)}{W} = \underset{(3,\ )}{O}$

(대응하는 차원의 원소 수를 일치시킨다.)

· Affine 계층의 계산그래프   np.dot(X, W) + B



∨ 스칼라값 대신 행렬이 흐르고 있음.

① $\underset{\substack{(2,\ )\\(N,2)}}{\frac{\partial L}{\partial X}} = \underset{\substack{(3,\ )\\(N,3)}}{\frac{\partial L}{\partial Y}} \cdot \underset{\substack{(3,2)\\(3,2)}}{W^T}$

↳ W의 전치행렬

② $\underset{\substack{(2,3)\\(2,3)}}{\frac{\partial L}{\partial W}} = \underset{\substack{(2,1)\\(2,N)}}{X^T} \underset{\substack{(1,3)\\(N,3)}}{\frac{\partial L}{\partial Y}}$

배치용 Affine 계층에서는 X의 형상이 (N,2)가 된다.

∨ 전치행렬의 곱은 dot 노드의 역전파 계산에 의한 것이기도 하지만,
행렬의 대응하는 차원의 원소 수가 일치하도록 곱을 조립하는 것이기도 하다.

· Softmax 계층 (출력층) : 입력값을 정규화하여 출력

ex)

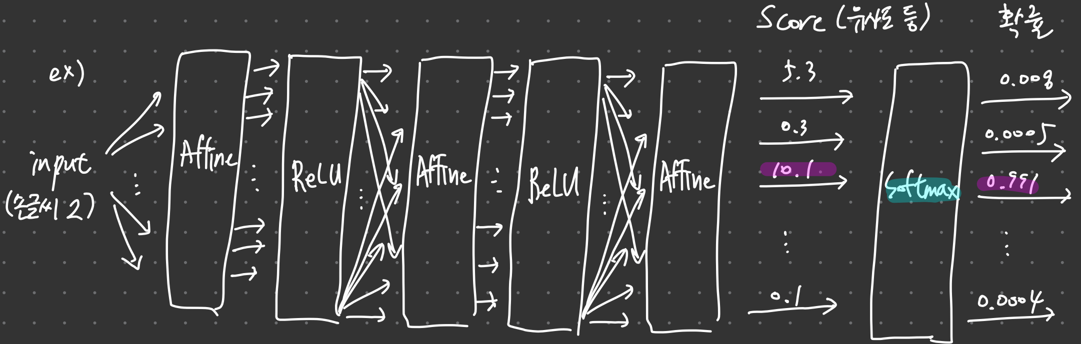
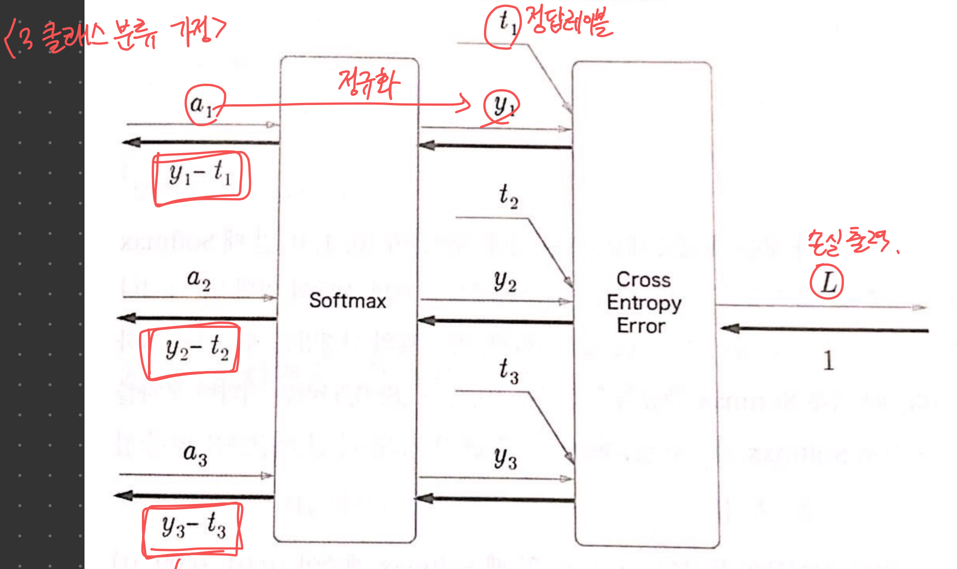


그림 5-30 '간소화한' Softmax-with-Loss 계층의 계산 그래프



"Softmax 계층의 출력과 정답 레이블의 차분."

교차 엔트로피 오차 함수의 설계 상 역전파가 말끔하게 떨어진다.

(회귀 출력층에서 사용하는 '항등함수'의 손실함수로 '평균제곱오차(MSE)'를 사용하는 것도 같은 이유)