

#0308_2

2021년 2월 8일 월요일 오전 9:28

■ 5장. 오차역전파법

□ 5.6 Affine / Softmax 계층 구현하기

● 5.6.1 Affine 계층

어파인 변환 : Affine transformation ---> 신경망의 순전파 시 수행하는 행렬의 곱
책에서는 어파인 변환을 수행하는 처리를 '어파인 계층' 이라고 표현하고 있음

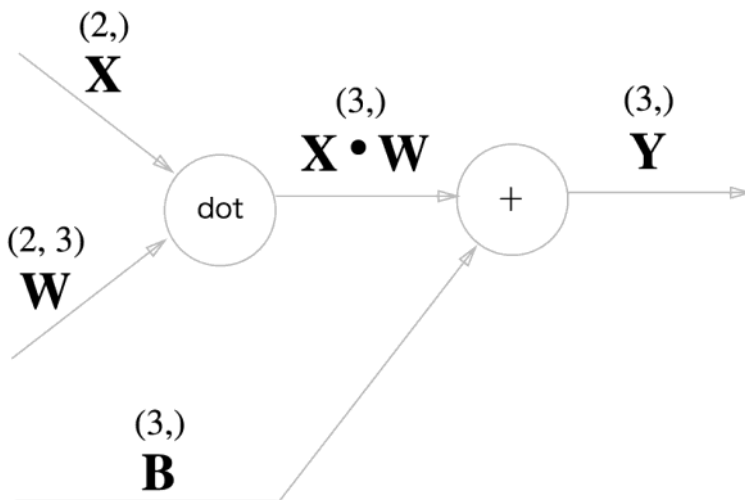
행렬 내적 연산

$$\begin{matrix} \mathbf{X} & \cdot & \mathbf{W} & = & \mathbf{O} \\ (2,) & & (2, 3) & & (3,) \end{matrix}$$

일치

- # 행렬의 곱에서는 대응하는 차원의 원소 수를 일치시켜야 한다
- # 위 연산의 결과로서, 1차원 dot(내적) 2차원 ---> 1차원 결과값 반환 임을 알고 있으면 좋다
- # 1차원 ndarray 의 경우 곧장 적용된다

[그림 5-24]



- # 순전파 과정
- # 순전파에서는 이전에는 스칼라 값이었으나, 지금은 다차원 배열인 것이 차이점
- # 행렬이기 때문에 형상에 주의해서 신경망을 구성하여야 한다
- # $X \cdot W + B = Y$

[식 5.13]

$$\frac{\partial L}{\partial \mathbf{X}} = \frac{\partial L}{\partial \mathbf{Y}} \cdot \mathbf{W}^T$$

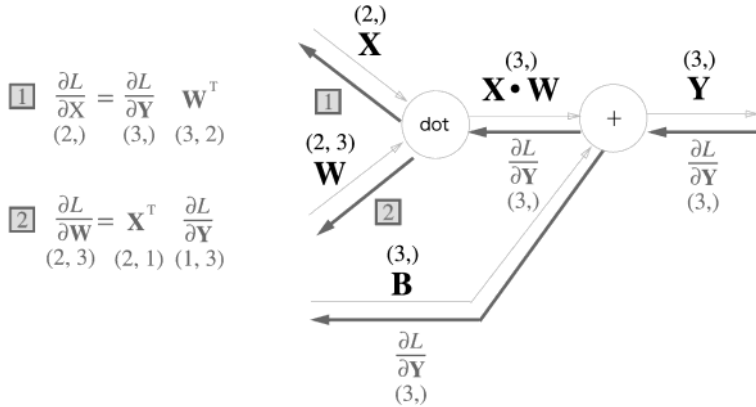
$$\frac{\partial L}{\partial \mathbf{W}} = \mathbf{X}^T \cdot \frac{\partial L}{\partial \mathbf{Y}}$$

- # 역전파 식
- # t는 전치행렬 : 원 행렬의 (i, j) 위치의 원소를 (j, i) 로 변경하는 것

어떤 행렬 "A.shape #(2,3)" 전치행렬을 시켜주면 "A_t.shape #(3,2)" 로 형상이 재설정된다

위 식과 같이 되는 것은 + 는 역전파시 *1 을 해주고, dot(내적) 연산은 서로 바꾸어 곱하기 때문에 5.13 식이 유도된다.

[그림 5-25] 역전파



X_t (X의 전치행렬) 와 W_t 값이 서로 위치가 다른 것은 두 행렬의 내적이 가능하도록 조정하기 위해서라고 이해

B 는 + 연산이므로 *1 만 수행하여 역전파 값이 흘러간다

[식 5-15]

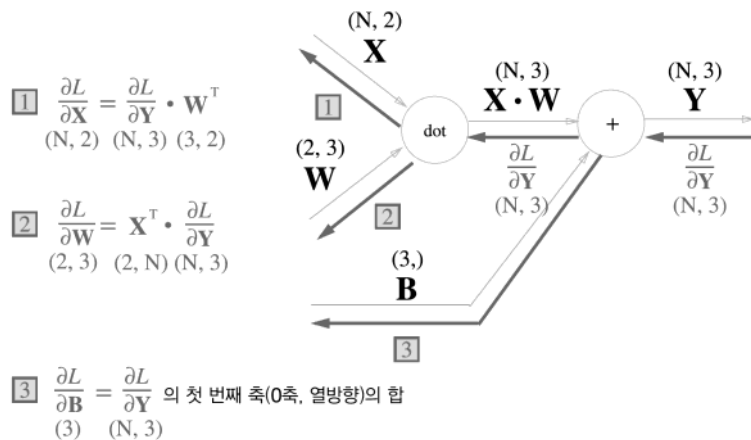
$$\mathbf{X} = (x_0, x_1, \dots, x_n)$$

$$\frac{\partial L}{\partial \mathbf{X}} = \left(\frac{\partial L}{\partial x_0}, \frac{\partial L}{\partial x_1}, \dots, \frac{\partial L}{\partial x_n} \right)$$

X 와 편미분X의 형상이 동일

그렇다면 왜 이 값이 일치해야 할까?

●5.6.2 배치용 Affine 계층



X 데이터에 N의 배치값이 설정된 것을 볼 수 있다.

N 배치만큼 신경망의 형상이 조정되는 것 외에는 앞과 큰 차이점이 없다.

편향 역전파시 순전파시 broadcasting 되었던 값들을 컬럼별로 더해서 합쳐서 역전파를 수행해야 한다.

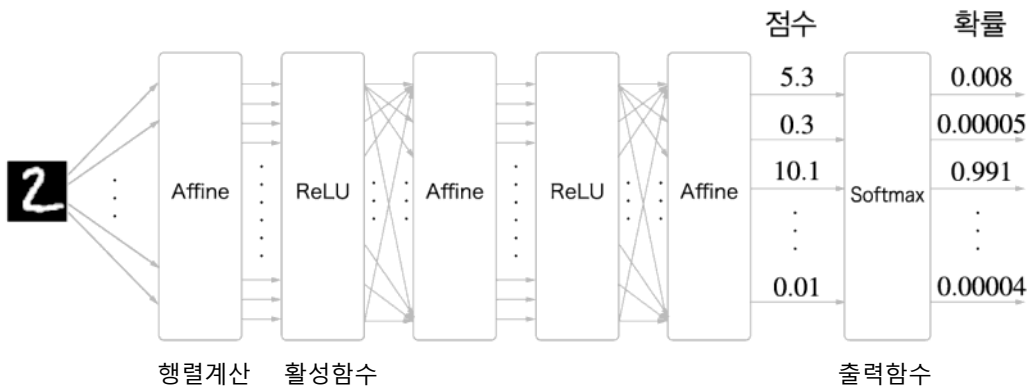
p.175 [code]

`dB = np.sum(dY, axis=0)` 부분이 해당 #axis=0 은 row(행)별 column 값들을 더하는 것

이후 코드는 한번 속 보고 지나가도록 합시다 (실제 구현하면 좋겠지만 사용하는 ANN, DNN 모델들은 더 복잡합니다)

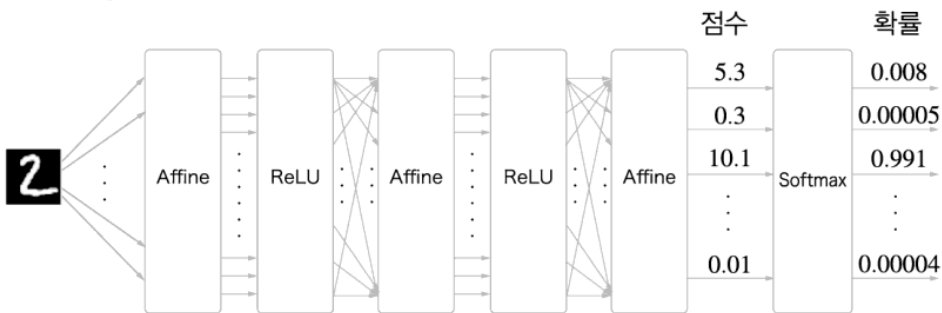
●softmax-with-loss 계층

[그림 5-28]

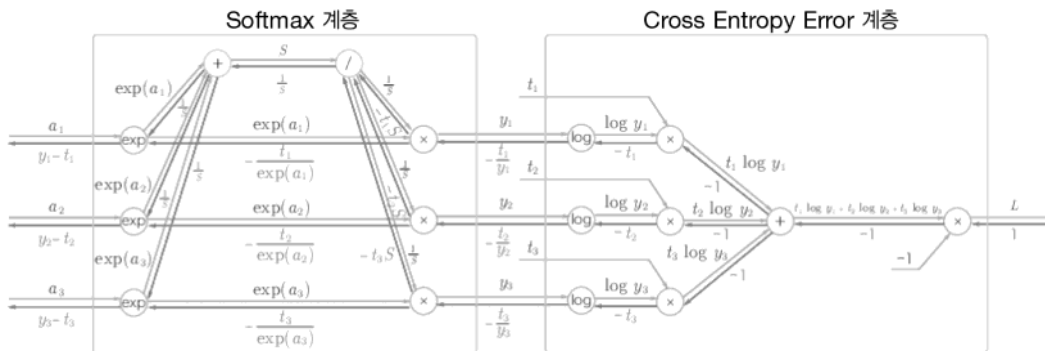


- # 신경망에서 수행하는 작업은 학습과 추론 2가지
- # Softmax 앞의 Affine 계층의 출력 = 점수(score)
- # 신경망 추론에서 답을 1개만 찾을 경우에는 softmax 출력함수를 적용할 필요 없이 최대값을 찾으면 된다
- # 신경망 학습시에는 softmax 계층이 필요하다
- # softmax 함수는 입력값들의 총합이 1인 probability(확률)의 개념으로 접근

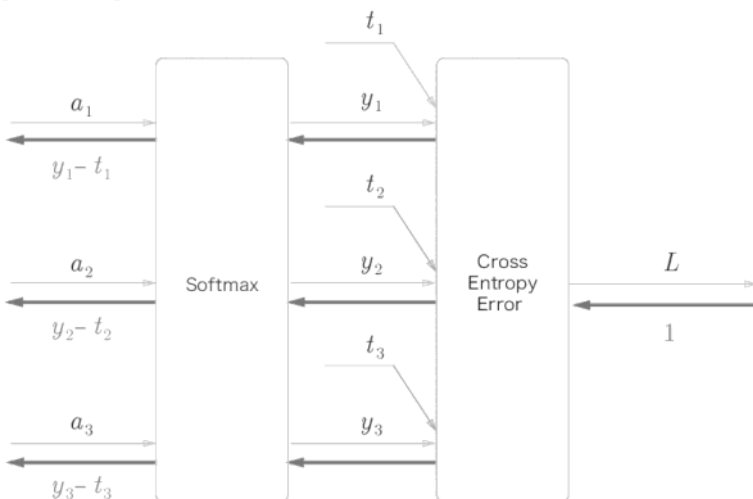
[그림 5-28]



[그림 5-29] Softmax-with-Loss



[그림 5-30]



손실함수로 교차 엔트로피 오차를 사용 : Cross Entropy Error

손실함수로 교차 엔트로피 오차를 사용하면 역전파가 $y_1 - t_1$ 형식으로 깔끔하게 떨어진다. ($y_1 - t_1$)=error

위와 같은 형식으로 역전파가 이루어지는 것은 교차 엔트로피 오차 함수의 설계목적이 이에 부합하기 때문

□ 5.7 오차역전파법 구현하기

●5.7.1 신경망 학습의 전체 그림

전체 : 신경망에는 weight, bias 가 있고, 이를 훈련 데이터에 적응하도록 조정하는 과정을 '학습' 이라고 한다

1단계 - 미니배치

훈련 데이터 중 일부를 무작위로 sampling, 선별된 데이터를 미니배치라 하며, 이 미니배치의 손실함수 값을 줄이는 것이 학습목표

2단계 - 기울기 산출

1단계의 목적 달성을 위해 각 가중치 매개변수의 기울기를 구한다. 기울기는 손실함수값을 가장 작게하는 방향을 제시

3단계 - 매개변수 갱신

가중치 매개변수를 기울기 방향으로 조금씩 갱신

4단계 - 반복

기울기가 0, 손실함수값이 더이상 감소하지 않을때까지 1~3단계 반복

●5.7.2 오차역전파법을 적용한 신경망 구현하기

[코드부분은 생략]

코드전체를 이해하려는 노력보다는 핵심 개념을 어떻게 코드로 구현했는지 살펴보기

Class 와 def method를 이용한 코드 구현의 큰 그림을 파악하는 정도로만 공부

#EndLine=====