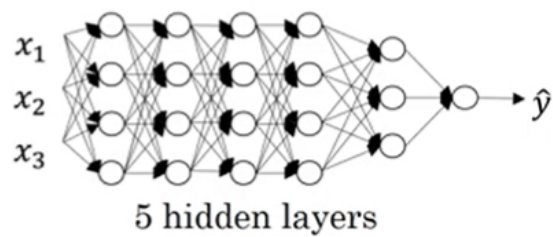
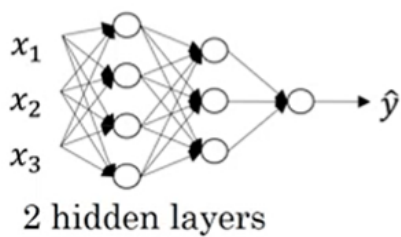
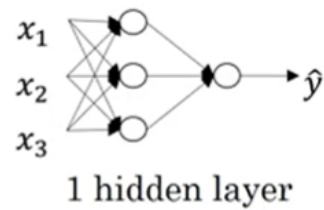
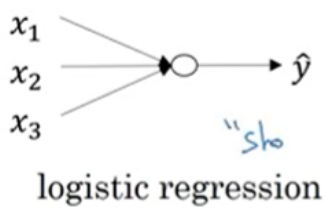


[Week5]_문가을

5. 심층 신경망 네트워크

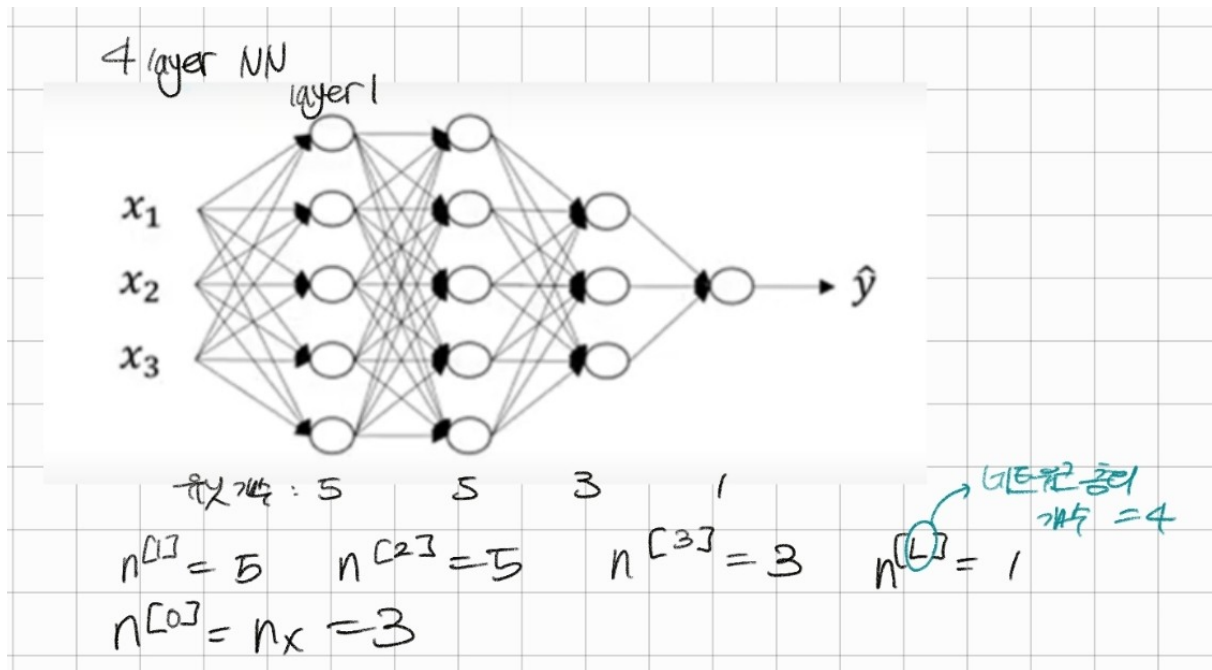
더 많은 층의 심층 신경망



*신경망의 층을 셀 때는 입력층은 세지 않음.

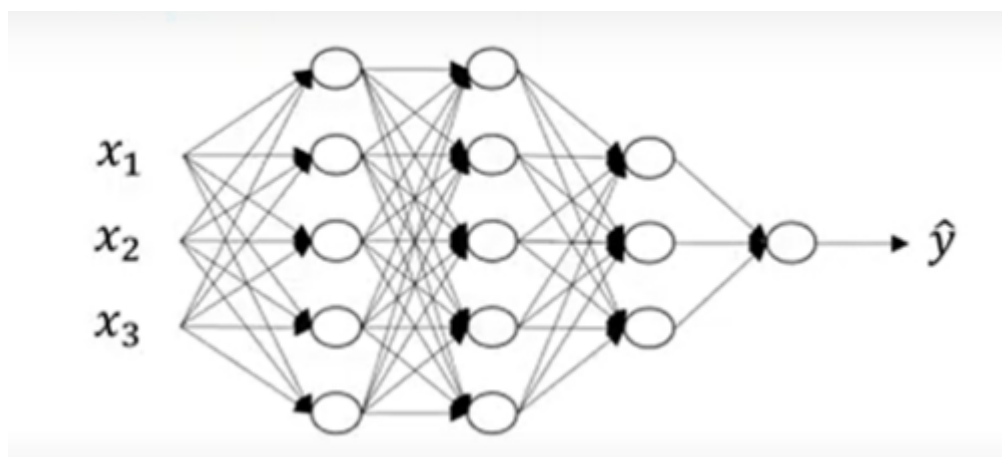
얼마나 깊은 신경망을 사용해야 하는지 미리 예측하기 어려움

→ 은닉층의 개수를 하이퍼파라미터로 두고, 다양한 값을 시도해서 평가해야 함.



- L : 네트워크 층의 수
- $n^{[l]}$: l 층에 있는 유닛 개수
- $a^{[l]}$: l 층에서의 활성화값
- $a^{[0]}$: 입력 특징 (x)
- $a^{[L]}$: 예측된 출력값 (\hat{y})

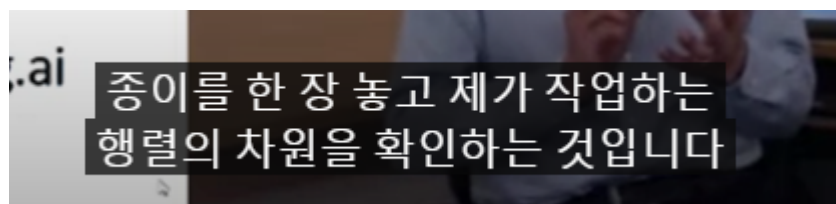
심층 신경망에서의 정방향전파



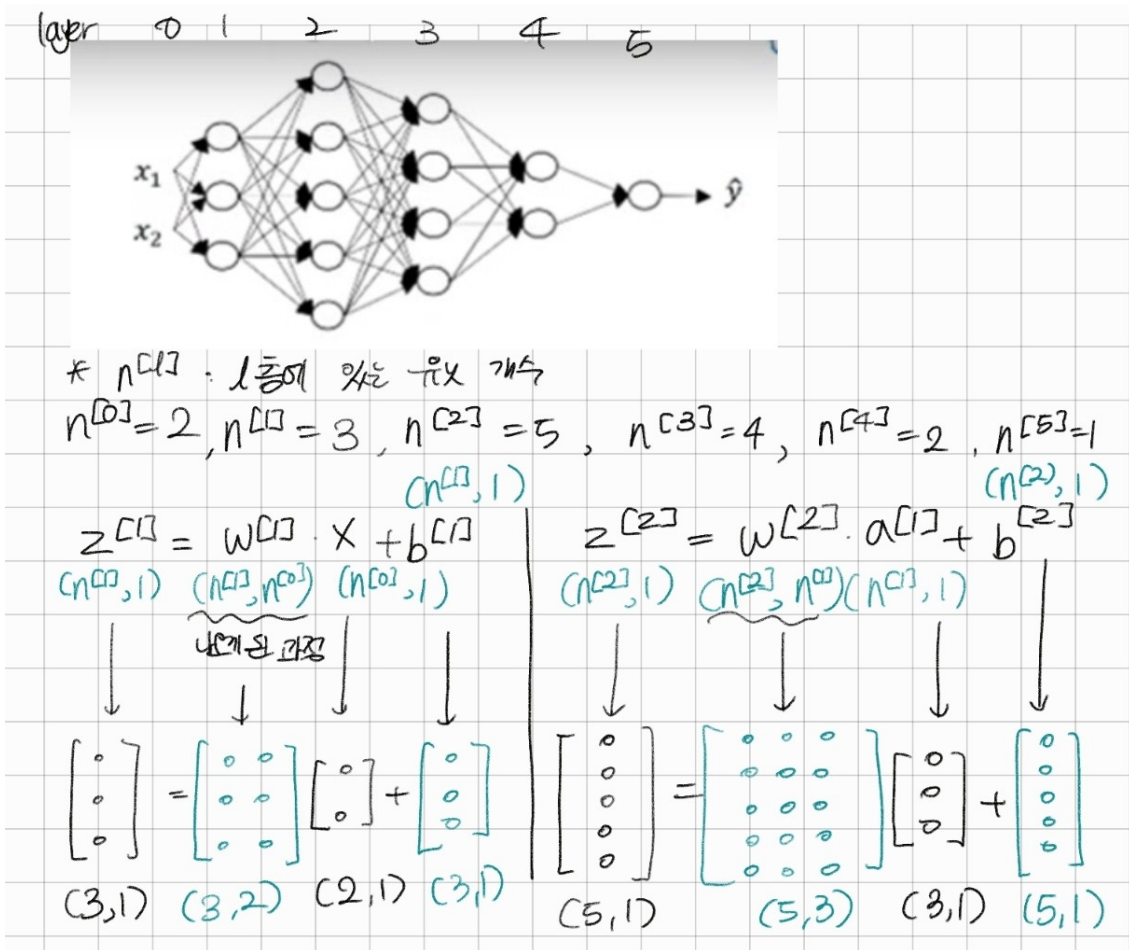
<p>layer 1</p> $Z^{[1]} = W^{[1]} X + b^{[1]}$ $a^{[1]} = g^{[1]}(Z^{[1]})$ <p>layer 2</p> $Z^{[2]} = W^{[2]} a^{[1]} + b^{[2]}$ $a^{[2]} = g^{[2]}(Z^{[2]})$ <p>⋮</p> <p>layer 4</p> $Z^{[4]} = W^{[4]} a^{[3]} + b^{[4]}$ $a^{[4]} = g^{[4]}(Z^{[4]})$ $Z^{[L]} = W^{[L]} a^{[L-1]} + b^{[L]}$ $a^{[L]} = g^{[L]}(Z^{[L]})$	<p>vectorizer for $l=1 \dots 4$</p> $Z^{[1]} = W^{[1]} A^{[0]} + b^{[1]}$ $A^{[1]} = g^{[1]}(Z^{[1]})$ $Z^{[2]} = W^{[2]} A^{[1]} + b^{[2]}$ $A^{[2]} = g^{[2]}(Z^{[2]})$ <p>⋮</p> $\hat{y} = g(Z^{[4]}) = A^{[4]}$ $Z^{[L]} = W^{[L]} A^{[L-1]} + b^{[L]}$ $A^{[L]} = g^{[L]}(Z^{[L]})$
--	---

$$* Z^{[L]} = \begin{bmatrix} Z^{[L](1)} & Z^{[L](2)} & \dots & Z^{[L](i)} & \dots & Z^{[L](m)} \end{bmatrix}$$

행렬의 차원을 알맞게 만들기



- 벡터화 X, 하나의 데이터에 대해 차원 확인



$w^{[l]} : (n^{[l]}, n^{[l-1]})$ $b^{[l]} : (n^{[l]}, 1)$	$z^{[l]} = g^{[l]}(a^{[l]})$ <div style="border: 1px solid black; padding: 2px; display: inline-block;"> $a^{[l]}$ </div> (활성화 함수)
$dw^{[l]} : (n^{[l]}, n^{[l-1]})$ $db^{[l]} : (n^{[l]}, 1)$	$a^{[l]} = z^{[l]}$

- 벡터화 O, m개의 데이터에 대한 차원 확인

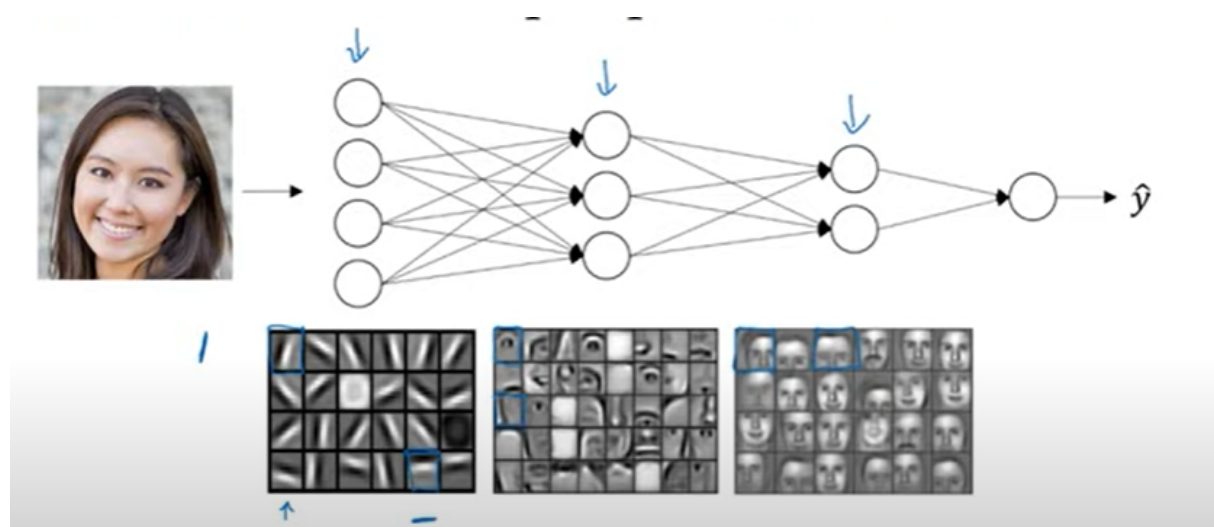
$$\begin{aligned}
 & \left[z^{[1]}_1 \ z^{[1]}_2 \ \dots \ z^{[1]}_m \right] \\
 & \quad \quad \quad (n^{[1]}, 1) \text{ broadcasting} \\
 & z^{[l]} = W^{[l]} \cdot X + b^{[l]} \\
 & \quad \quad \quad (n^{[l]}, m) \quad (n^{[l]}, n^{[l-1]}) \quad (n^{[l]}, m) \\
 & z^{[L]}, A^{[L]} = (n^{[L]}, m) \\
 & dz^{[L]}, dA^{[L]} = (n^{[L]}, m)
 \end{aligned}$$

왜 심층 신경망이 더 많은 특징을 잡아낼 수 있을까요?

직관 1: 네트워크가 더 깊어질수록, 더 많은 특징을 잡아낼 수 있다.

→ 낮은 층에서는 간단한 특징을 찾아내고, 깊은 층에서는 탐지된 간단한 것들을 함께 모아 복잡한 특징을 찾아낼 수 있습니다.

시각화 예시 (합성곱 신경망)



- 1번째 층

각 방향에서의 모서리가 어디에 있는지 알아냄.

- 2번째 층

모서리와 그룹화된 모서리를 받아서 얼굴의 일부 형성

- 3번째 층

얼굴의 일부를 최종적으로 모아서 서로 다른 종류의 얼굴을 감지함.

음성 인식 예

낮은 단계의 음성 파형 특징 → 소리의 기본 단위인 음소를 찾기

→ 단어 → 문장

직관 2 : 충분한 은닉층이 없다면, 계산에 기하급수적으로 많은 은닉 유닛이 필요하다.

→ 순환이론에 따라 상대적으로 은닉층의 개수가 작지만, 깊은 신경망에서 계산할 수 있는 함수가 있다. 그러나 얇은 네트워크로 같은 함수를 계산하려고 하면, 많은 은닉 유닛이 계산에 필요하다.

→ 얇은 네트워크보다 깊은 네트워크에서 더 계산하기 쉬운 수학적 함수가 있다.

- 순환 이론 : 로직 게이트의 서로 다른 게이트에서 어떤 종류의 함수를 계산할 수 있을지에 관한 것이다.

논리 회로(**영어**: logic gate)는 **불 대수**를 물리적 장치에 구현한 것으로, 하나 이상의 논리적 입력값에 대해 **논리 연산**을 수행하여 하나의 논리적 출력값을 얻는 **전자회로**를 말한다. **AND, OR, NOT**의 기본 불 대수를 수행하며, 이 기본 불 대수들의 결합으로 복합적인 논리 기능을 수행한다.

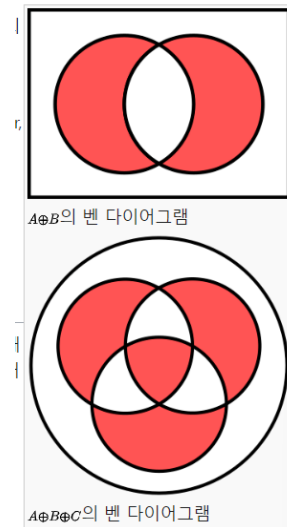
예시 : 배타적 논리합

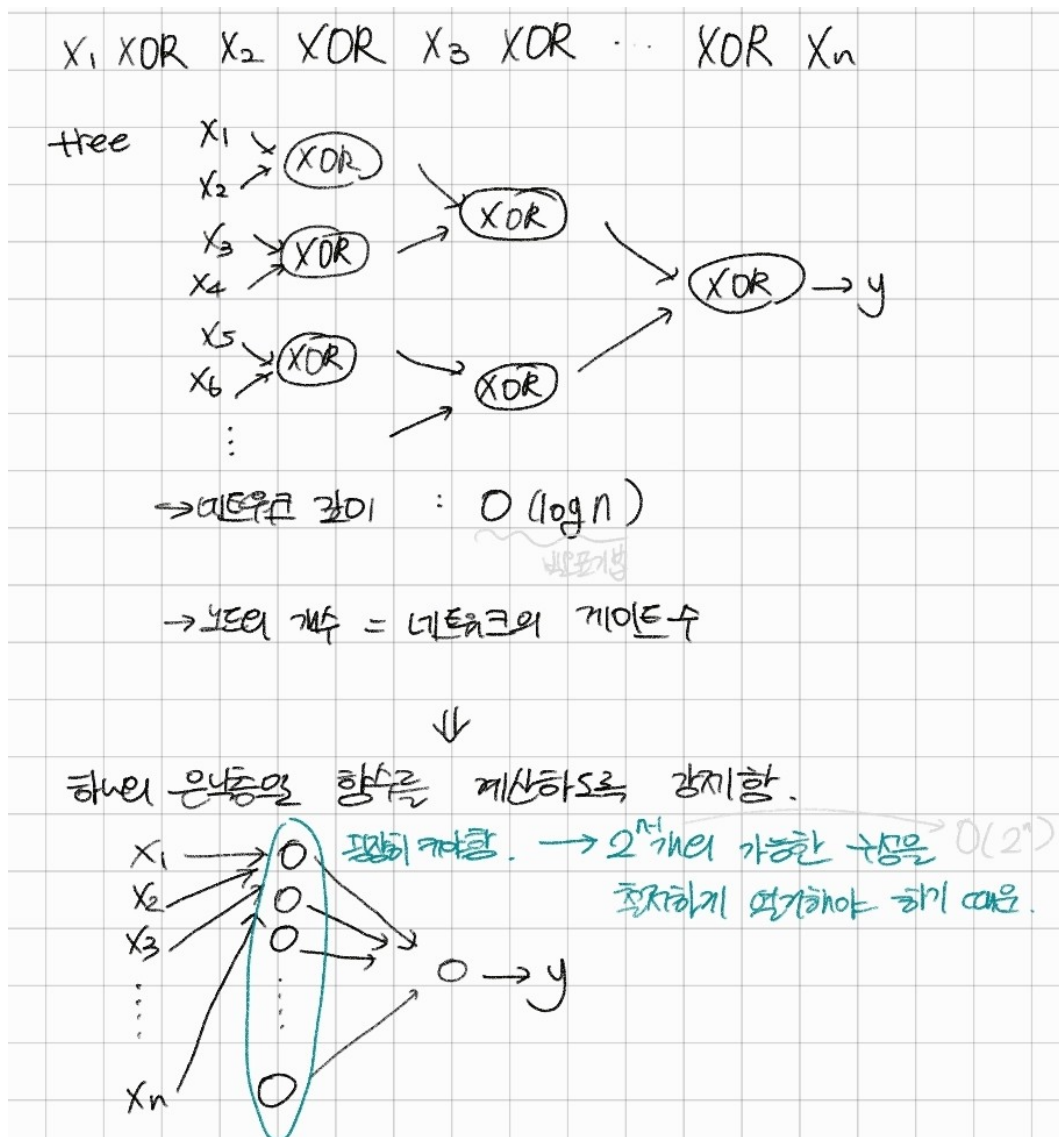
배타적 논리합(排他的論理合, exclusive or)은 **수리 논리학**에서 주어진 2개의 명제 가운데 1개만 참일 경우를 판단하는 **논리 연산**이다. 약칭으로 **XOR**, **EOR**, **EXOR**라고도 쓴다.

예시 [편집]

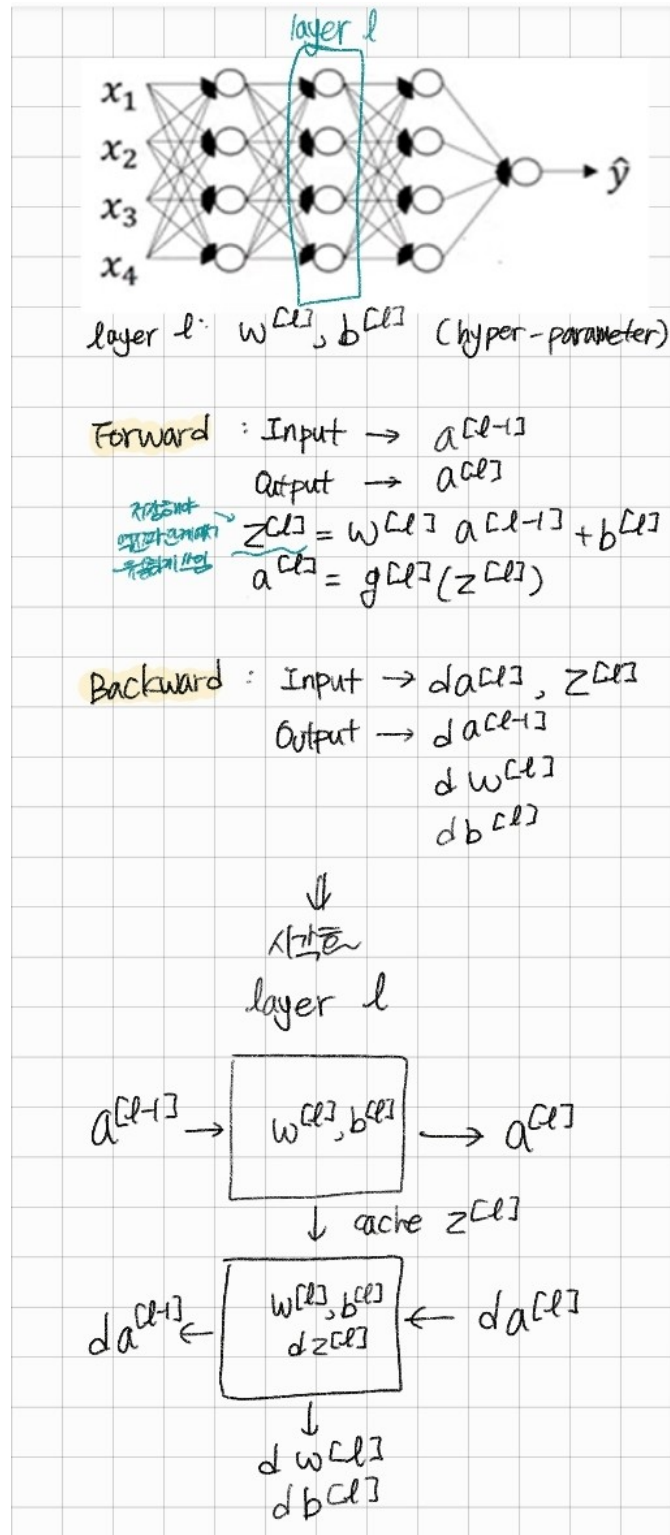
“내 키는 160cm 이상이다.”와 “내 몸무게는 60kg 미만이다.” 이 두 명제의 배타적 논리합은, “내 키는 160cm 이상이고 몸무게는 60kg 이상이다. 혹은, 내 키는 160cm 미만이고 몸무게는 60kg 미만이다.”가 된다.

추가로, 두 명제 A, B 에 대한 **교집합** ($A \wedge B$)가 **공집합**이면, 배타적 논리합은 **논리합**과 같게 된다. 예를 들어, A = “내 키는 160cm이다.”와 B = “내 키는 170cm이다.”는 동시에 성립할 수 없기 때문에(교집합이 없음), $(A \text{ xor } B)$ 와 $(A \vee B)$ 는 동일하게 “내 키는 160cm와 170cm 중 하나다.”가 된다.

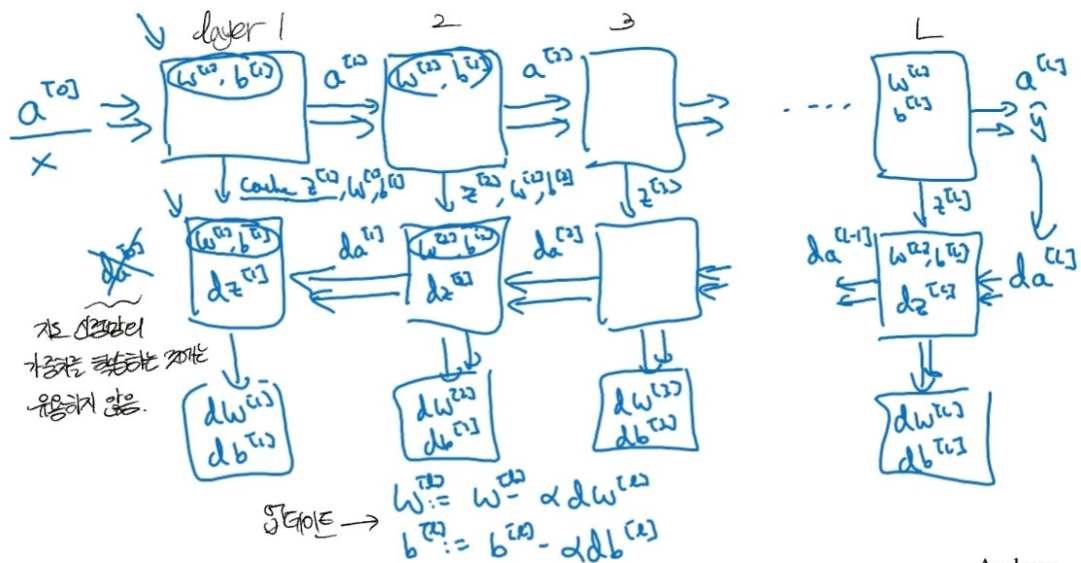




심층 신경망 네트워크 구성하기



- 모든 층에서 전방향, 역방향 계산 시각화



정방향전파와 역방향전파

- forward propagation for layer l

Input $a^{[l-1]}$

Output $a^{[l]}$, cache $(z^{[l]})$

$z^{[l]} = w^{[l]} \cdot a^{[l-1]} + b^{[l]}$ $a^{[l]} = g^{[l]}(z^{[l]})$	<p>[Vectorization]</p> $Z^{[l]} = W^{[l]} \cdot A^{[l-1]} + b^{[l]}$ $A^{[l]} = g^{[l]}(Z^{[l]})$
--	---

- back propagation for layer l

Input $da^{[l]}$

Output $da^{[l-1]}$, $dW^{[l]}$, $db^{[l]}$

$$\begin{aligned}
dZ^{[L]} &= dA^{[L]} * g^{[L]'}(Z^{[L]}) \\
dW^{[L]} &= dZ^{[L]} \cdot A^{[L-1]} \\
db^{[L]} &= dZ^{[L]} \\
dA^{[L-1]} &= W^{[L]T} \cdot dZ^{[L]} \\
dZ^{[L-1]} &= W^{[L+1]T} dZ^{[L]} * g^{[L]'}(Z^{[L]})
\end{aligned}$$

벡터화

$$\begin{aligned}
dZ^{[L]} &= dA^{[L]} * g^{[L]'}(Z^{[L]}) \\
dW^{[L]} &= \frac{1}{m} dZ^{[L]} \cdot A^{[L-1]T} \\
db^{[L]} &= \frac{1}{m} \text{np.sum}(dZ^{[L]}, \text{axis}=1, \text{keepdims}=\text{True}) \\
dA^{[L-1]} &= W^{[L]T} \cdot dZ^{[L]}
\end{aligned}$$

Parameters vs Hyperparameters

Parameters: $W^{[1]}, b^{[1]}, W^{[2]}, b^{[2]}, W^{[3]}, b^{[3]} \dots$

Hyper parameters

Learning rate : α
number of iteration
number of hiddenlayer : L
number of units : $n^{[1]}, n^{[2]} \dots$
choise of activationfunction
 ...

hyper parameter는 parameter를 통제함.

Applied deep learning is a very empirical process!

새로운 애플리케이션을 시작할 때 hyper parameter의 가장 적합한 값을 정확히 미리 아는 것은 어려움.

→ 다양한 값을 시도하고, experiment의 결과를 관찰하고 다른 값으로 적용하고... 반복

인간의 뇌와 어떤 연관이 있을까요?

뇌의 신경세포인 뉴런은 다른 뉴런으로부터 전기적 신호를 받는다.

→ 계산을 한 후, 전기 신호를 다른 뉴런에게 보냄.

딥러닝 분야와 뇌의 비유는 점점 무너져 가고 있음.