

5.심층 신경망 네트워크

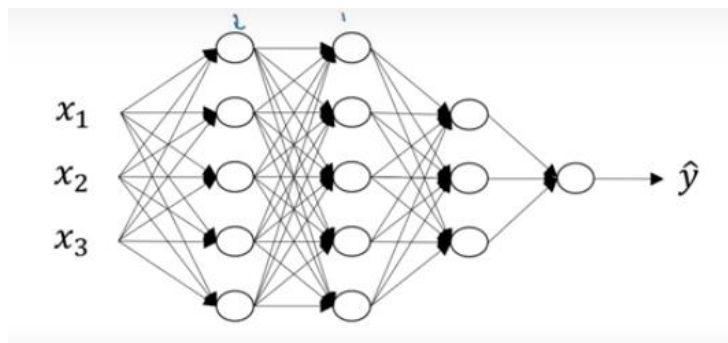
<더 많은 층의 심층 신경망>

로지스틱 회귀 모형(예시로 층이 2개) : 얕은 모형

은닉층이 5개인 신경망 : 깊은 모델

★ 층 셀 때 입력 층 안센다

<표기법>



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

- 4개의 층, 은닉층 3개
- 각 은닉층의 유닛 개수 5개, 5개, 3개
- $L=5$
- $n[1]=5, n[3]=3$
- 출력층을 layer 0 라 하면 차례대로 0,1,2,3,4

L: layer 수

$n^{[l]}$: 소문자 l층의 단위(unit)의 개수

$a^{[l]}$: l층에서의 활성화값

$a^{[l]}=g(z^{[l]}) \Rightarrow g^{[l]}$ 는 활성화 함수

$W^{[l]}$: $z^{[l]}$ 값을 계산하기 위한 가중치

X : 입력 특징 (x는 층 0의 활성화값 $a^{[0]}=\hat{y}$)

<정방향전파와 역방향전파>

★ 정방향 전파

⇒ l번째 층에서 이전 층의 활성화 값 $a^{[l-1]}$ 를 입력으로 받아 $a^{[l]}=g^{[l]}(z^{[l]})$ 를 출력함

⇒ $z^{[l]}=W^{[l]}*a^{[l-1]}+b^{[l]}$, $W^{[l]}$, $b^{[l]}$ 는 캐시로 저장함

벡터화 ⇒

$$Z^{[l]}=W^{[l]}A^{[l-1]}+b^{[l]}$$

$$A^{[l]}=g^{[l]}(Z^{[l]})$$

⇒ $a^{[0]}$: **한 번에 하나씩** 할 경우의 학습 데이터에 대한 입력 특성

⇒ $A^{[0]}$: **전체 학습 집합**을 진행 할 때의 입력 특성

★ 역전파 (도함수 계산)

l번째 층에서 $da^{[l]}$ 을 입력으로 받아서 $da^{[l-1]}$ 출력함, 업데이트를 위한 $dW^{[l]}$ 과 $db^{[l]}$ 도 출력함.

벡터화 ⇒

$$dz^{[l]}=dA^{[l]}*g^{[l]'}(Z^{[l]}) : \text{요소별 곱셈}$$

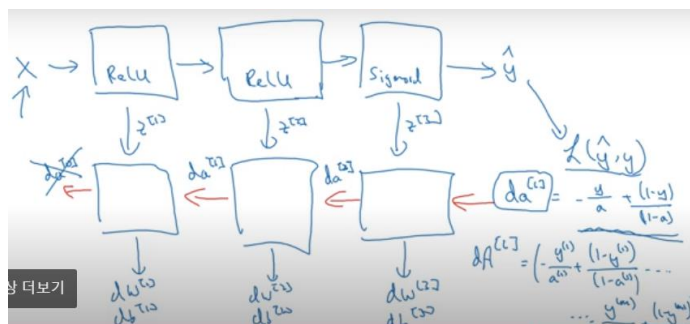
$$dW^{[l]}=1/m \times dz^{[l]} \times A^{[l-1]}$$

$$db^{[l]}=1/m \times \text{np.sum}(dz^{[l]}, \text{axis}=1, \text{keepdims}=\text{True})$$

$$dA^{[l-1]}=W^{[l]T} \times dz^{[l]}$$

○ 정방향 반복은 입력데이터 X로 초기화한다.

○ 역방향 반복은 $da[l]$ or dA (벡터화된 구현일 때)



심층 신경망에서의 정방향 전파

$z[l] = W[l]a[l-1] + b[l]$ (l층의 편향 벡터 더한 것)

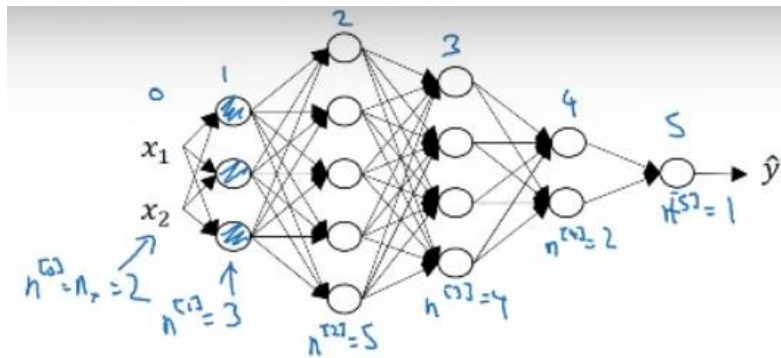
$a[l] = g^{[l]}(z[l])$

벡터화 \Rightarrow

$Z[l] = W[l]A[l-1] + b[l]$ (z벡터를 열로 쌓아서 만든게 Z)

$A[l] = g^{[l]}(Z[l])$

행렬의 차원을 알맞게 만들기



$n[1]=3, n[2]=5$

5개의 층 = 4개 은닉층 + 1개의 출력층

$z[l] = w[l]a[l-1] + b[l]$

z는 첫번째 은닉층에 대한 활성화 벡터 \Rightarrow (3,1) 3차원 벡터, $(n[1],1)$

\Rightarrow x는 $(n[0],1)$ 니까 $w[1]$ 는 (3,2) 여야함 $(n[1],n[0])$

★ $W[l] \Rightarrow (n[l], n[l-1])$ 차원

★ $b[l] \Rightarrow (n[l], 1)$ 차원

★ $dW^{[l]}$ 차원은 $W^{[l]}$ 차원과 같아야함

★ $z^{[l]} = g^{[l]}(a[l]) \Rightarrow$ z와 a의 차원은 같아야함

벡터화 \Rightarrow

★ $Z[1] = W[1]X + b[1]$

★ $Z \Rightarrow (n[1], m)$ 차원 (m은 training set 의 크기)

★ $W \Rightarrow (n[1], n[0])$ 차원

★ $X \Rightarrow (n[0], m)$ 차원 (모든 학습 데이터가 수평으로 저장되어 있기 때문)

\Rightarrow 결과적으로 $Z[1]$ 을 계산하면 $(n[1], m)$ 가 나옴

$z[l], a[l] \Rightarrow (n[l], 1)$

$Z[l], A[l] \Rightarrow (n[l], m)$

$l=0$ 일때 $\Rightarrow A[0] = X = (n[0], m)$

$dZ \Rightarrow (n[l], m)$

★ $b=> (n[1],1)$ (이후 브로드캐스팅으로 더해짐)

왜 심층 신경망이 더 많은 특징을 잡아낼 수 있을까요?

<입력 값이 얼굴 사진일 때>

1. 첫번째 층: 특성 탐지거나 모서리 탐지기. (수평방향의 모서리가 어디에 있는지)
모서리를 형성하기 위해 픽셀을 그룹화
2. 많은 모서리를 모아 얼굴의 일부를 감지
3. 다른 얼굴의 일부를 모아서 서로 다른 종류의 얼굴 감지

★ 간단한 함수를 감지하고 층에서 이것을 구성해서 복잡한 함수 학습하도록 함.

모서리 탐지기는 이미지에서 작은 영역을 보고 얼굴 탐지기는 이미지의 넓은 영역을 봄

★ 합성곱 신경망

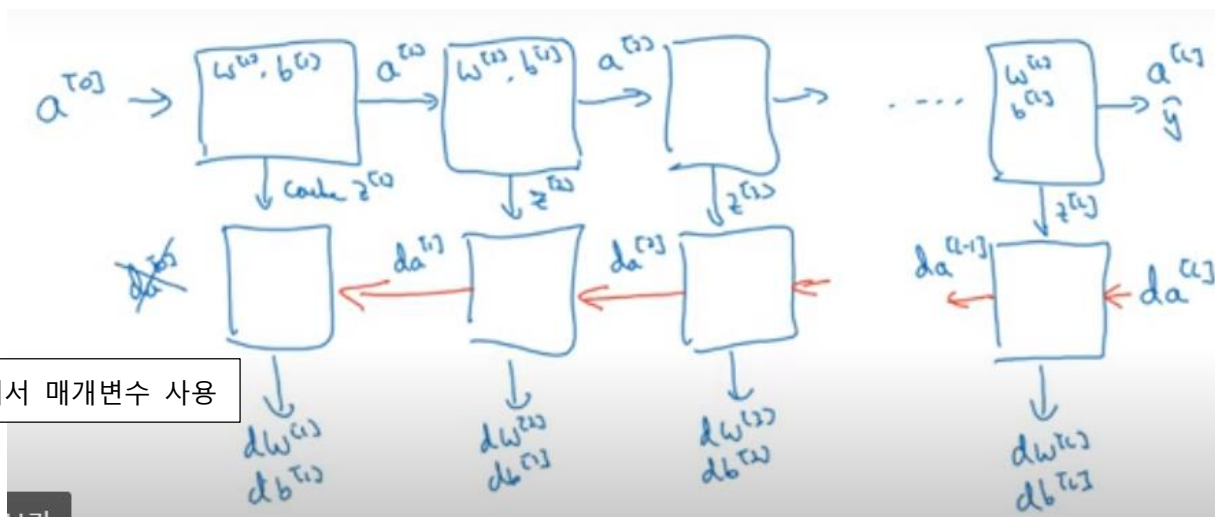
<음성인식 시스템일 때>

1. 첫번째 층: 음성을 입력으로 주면 낮은 단계의 음성파형 특징 탐지
2. 소리의 기본인 음소 학습하고 그것을 구성해서 단어 인식을 학습하고 문장 인식

★ 회로이론은 로직 게이트의 서로 다른 게이트에서 어떤 종류의 함수를 계산할 수 있을지 봄

⇒ 얇은 네트워크는 은닉 유닛이 훨씬 많이 필요하다!

심층 신경망 네트워크 구성하기



<정방향 함수> ⇒ 결과적으로 \hat{y} 출력

- $a[l-1]$ (이전 층의 활성화 값)를 입력으로 받아 $a[l]$ 값 출력
- 선형결합된 $z[l]$ 과 $W[l], b[l]$ 캐시로 저장

<역전파> ⇒ 역방향으로 가면서 경사 계산

- $da[l]$ 을 (+캐시 $z[l]$) 입력으로 받아 $da[l-1]$ (그래디언트) 출력
- 이전층의 활성화에 대한 도함수
- $W[l], b[l]$ 역방향 함수는 $dW[l], db[l]$
- 특성 $a[0]$ 가져와서 $a[1]$ 활성화 구하고 ($W[1], b[1]$) $Z[1]$ 캐시에 저장
- $da[l]$ $da[l-1]$... $da[1]$ 순서로 구함
- $dW[l], db[l]$ $dW[1], db[1]$ 순서로 구함
- $W[l]$ 값은 학습률을 곱해 $W[l]*adW[l]$ 로 업데이트 함

변수 vs 하이퍼파라미터

<하이퍼파라미터>

- 알파 learning rate
- 반복횟수
- 은닉층 개수 (L)
- 은닉 유닛 개수
- 활성화 함수 (Relu, tanh, sigmoid)
- 모멘텀
- 미니배치 사이즈

⇒이 매개변수들은 W랑 b 결정