

# 4주차

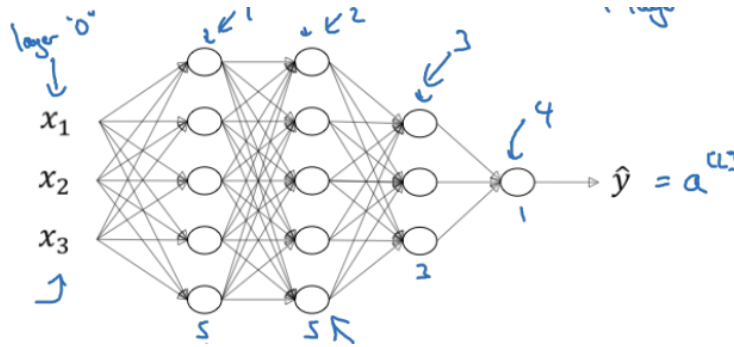
## 5. 심층 신경망 네트워크

### 1. 더 많은 층의 심층 신경망

#### 심층신경망이란?

- 로지스틱 회귀 : 매우 얇은 모델, 한 층의 신경망
- 심층 신경망 : 더 깊은 모델, 여러 개의 은닉층+출력층을 가진 신경망


#### 심층신경망을 설명하기 위한 표기법




- $L$  : 네트워크 층의 개수
  - $L=4$
- $n^{[l]}$  :  $l$ 층에 있는 유닛 개수
  - $n^{[1]}=5, n^{[2]}=5, n^{[3]}=3, n^{[4]}=n^{[L]}=1$
  - $n^{[0]}=n_{\{x\}}=3$
- $a^{[l]}$  :  $l$ 층에서의 활성화값
- $w^{[l]}$  :  $z^{[l]}$ 의 값을 계산하기 위한 가중치
- $b^{[l]}$
- $a^{[0]}=X$  : 입력 특징
  - $g^{[l]}(z^{[l]})$
- $a^{[L]}=\hat{y}$  : 예측된 출력값

## 2. 심층 신경망에서의 정방향전파

### 단일 학습 데이터 정방향 전파


$$z^{[l]} = w^{[l]}a^{[l-1]} + b^{[l]}$$
$$a^{[l]} = g^l(z^{[l]})$$

### 전체 학습 세트 벡터화



$$Z^{[l]} = w^{[l]}A^{[l-1]} + b^{[l]}$$
$$A^{[l]} = g^l(Z^{[l]})$$

- 층 1부터 L까지 각각에 대한 활성화를 계산하는 반복문을 사용하는 것은 괜찮음
  - 명시적 반복문 외에 구현 방법이 x


## 3. 행렬의 차원을 알맞게 만들기

$$z^{[l]} = W^{[l]}x + b^{[l]}$$

- 차원은 각각  $(n^{[1]}, 1)(n^{[1]}, n^{[0]})(n^{[0]}, 1)(n^{[1]}, 1)$


$$W^{[l]} : (n^{[l]}, n^{[l-1]})$$
$$b^{[l]} : (n^{[l]}, 1)$$


- W와 b 벡터가 올바른 차원을 가지는지 이중으로 확인
- 역전파 구현 경우
  - W는 dW, b는 db와 같은 차원을 가짐


$$dW^{[l]} : (n^{[l]}, n^{[l-1]})$$
$$db^{[l]} : (n^{[l]}, 1)$$

## 벡터화된 구현

$$Z^{[l]} = W^{[l]} X + b^{[l]}$$


- 차원은 각각  $(n^{[1]}, m)(n^{[1]}, n^{[0]})(n^{[0]}, m)(n^{[1]}, 1)$ 
  - $b^{[l]}$ 은  $(n^{[1]}, m)$ 으로 broadcasting


$$Z^{[l]}, A^{[l]} : (n^{[l]}, m)$$
$$dZ^{[l]}, dA^{[l]} : (n^{[l]}, m)$$

- $l=0$  일 때  $A^{[0]} = X = (n^{[0]}, m)$
- $z^{[l]}, a^{[l]} : (n^{[l]}, 1)$
- $Z/A$ 와  $dZ, dA$ 는 같은 차원을 가짐

## 4. 왜 심층 신경망이 더 많은 특징을 잡아 낼 수 있을까?

### 직관 1



#### 직관 1

- 네트워크가 더 깊어질수록, 더 많은 특징을 잡아낼 수 있음
- 낮은 층에서는 간단한 특징을 학습
- 깊은 층에서는 탐지된 간단한 것들을 함께 모아 더 복잡한 것들을 탐지

### 얼굴인식

- 입력값: 얼굴사진
- 20개의 은닉층 : 작은 네모상자
  1. 첫번째 층 : 특성 탐지기, 모서리 탐지기
    - 사진을 보고 모서리가 어디에 있는지 파악
  2. 두번째 층 : 모서리를 형성하기 위해 픽셀을 그룹화
    - 감지된 모서리와 그룹화된 모서리를 받아서 얼굴의 일부 형성
    - 즉, 많은 모서리를 한데 모아서 얼굴의 일부 감지 가능

### 3. 세번째 층

- 최종적으로 서로 다른 얼굴의 일부를 모아서 서로 다른 종류의 얼굴 감지 가능
- 모서리처럼 간단한 것을 찾고 같이 모아서 더 복잡한 것을 찾고 또 모아서 더 복잡한 것을 찾는다

### 음성인식시스템

- 입력 : 음성
1. 첫번째 층 : 낮은 단계의 음성 파형 특징을 탐지
  2. 두번째 층 : 소리의 기본 단위 탐지를 학습 (음소)
  3. 세번째 층 : 단어
  4. 네번째 층 : 문장/구

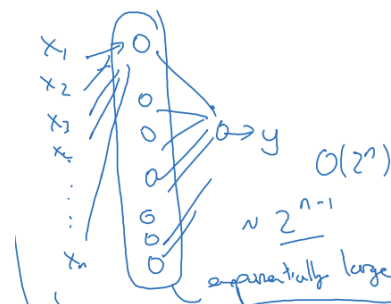
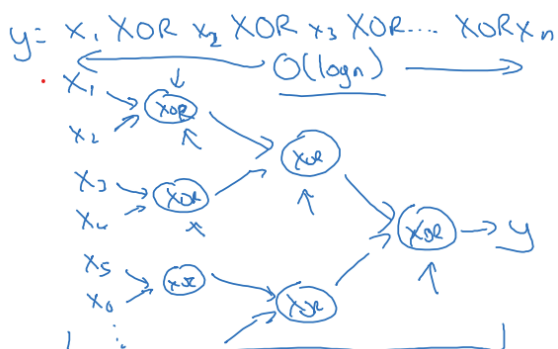
## 직관 2



### 직관 2

- 순환 이론에서 따르면, 상대적으로 은닉층의 개수가 작지만 깊은 심층 신경망에서 계산할 수 있는 함수가 있음
- 얇은 네트워크로 같은 함수를 계산하려고 하면, 즉 충분한 은닉층이 없다면 기하급수적으로 많은 은닉 유닛이 계산에 필요

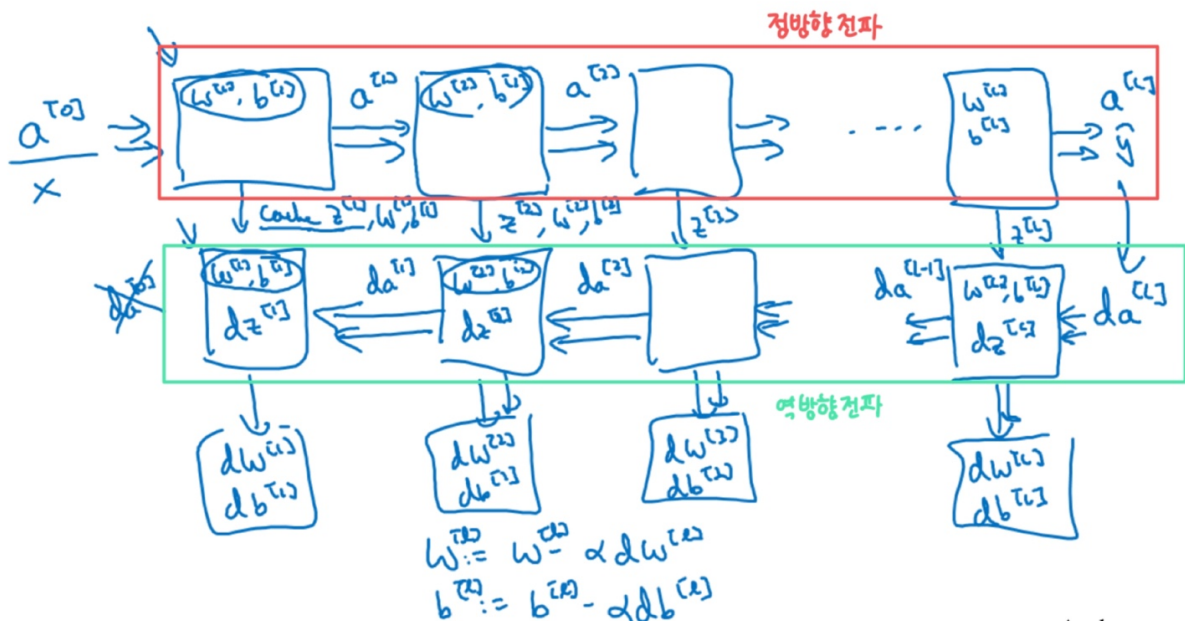
### 배타적논리합



- 여러개의 은닉층이 있는 네트워크
- $O(\log n)$
- 매우 큰 은닉층이 필요해짐
- $O(2^n)$

## 5. 심층 신경망 네트워크 구성하기

- Layer  $l$ :  $W^{[l]}, b^{[l]}$
- 정방향 전파
  - Input  $a^{[l-1]}$
  - Output  $a^{[l]}$
  - $z^{[l]} = W^{[l]}a^{[l-1]} + b^{[l]}$
  - $a^{[l]} = g^{[l]}(z^{[l]})$
  - 나중에 사용할 경우를 대비해 cache에  $z^{[l]}$  값을 저장
- 역방향 전파
  - Input  $da^{[l]}$ /cache  $z^{[l]}$  사용
  - Output  $da^{[l-1]}$



Andrew

- $da^{[0]}$ 는 지도 신경망의 가중치를 학습하는 것에는 유용하지 않으므로 계산하지 않는다

## 6. 정방향 전파와 역방향 전파

### 정방향 전파

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

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

## 구현

- $z^{[l]} = W^{[l]}a^{[l-1]} + b^{[l]}$
- $a^{[l]} = g^{[l]}(z^{[l]})$

## 벡터화된 구현

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

## 역방향 전파

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

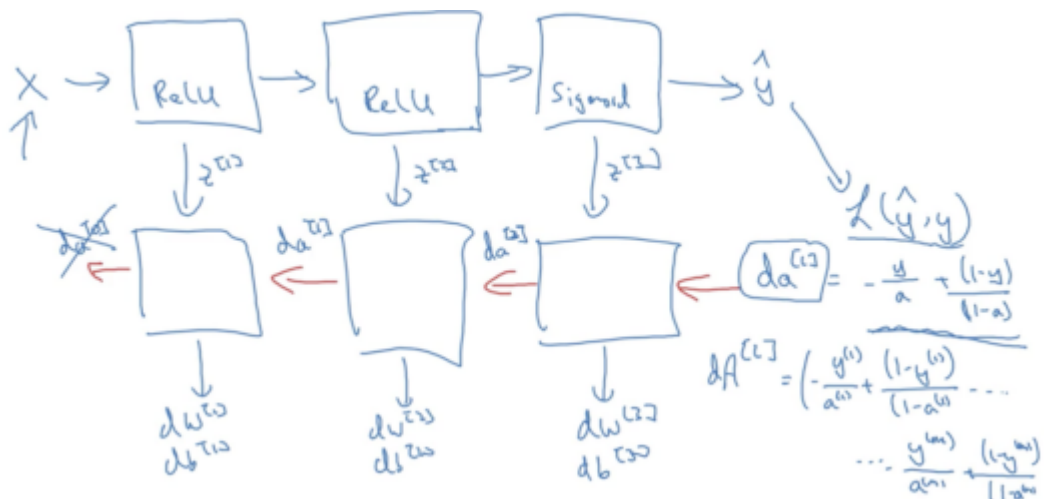
## 구현

$$\begin{aligned} dz^{[l]} &= da^{[l]} * g^{[l]'}(z^{[l]}) \\ dW^{[l]} &= dz^{[l]} \cdot a^{[l-1]} \\ db^{[l]} &= dz^{[l]} \\ da^{[l-1]} &= W^{[l]T} \cdot dz^{[l]} \\ dz^{[l-1]} &= W^{[l+1]T} dz^{[l]} * g^{[l+1]'}(z^{[l-1]}) \end{aligned}$$

## 벡터화된 구현

$$\begin{aligned} dz^{[l]} &= dA^{[l]} * g^{[l]'}(z^{[l]}) \\ dW^{[l]} &= \frac{1}{n} dz^{[l]} \cdot A^{[l-1]T} \\ db^{[l]} &= \frac{1}{n} np.sum(dz^{[l]}, axis=1, keepdims=True) \\ dA^{[l-1]} &= W^{[l]T} \cdot dz^{[l]} \end{aligned}$$

## 요약



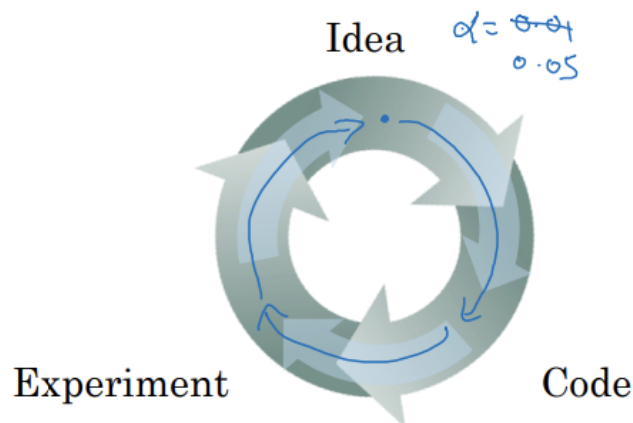
- 역방향 반복에 대한 초기화

- 일반 구현 : A에 대해 손실 함수 L미분 ( $da^{[l]}$ )
- 벡터화된 구현 : 층 L에 대한 dA ( $dA^{[l]}$ )

## 7. 변수 vs 하이퍼파라미터

- 변수 : 신경망에서 학습 가능한 W,b
- 하이퍼 파라미터 : 변수 W,b를 통제하는 매개변수
  - 학습률 (learning rate,  $\alpha$ )
  - 반복횟수 (numbers of iteration)
  - 은닉층의 갯수 (numbers of hidden layer, L)
  - 은닉유닛의 갯수 (numbers of hidden units)
  - 활성화 함수의 선택 (choice of activation function)
  - Later
    - 모멘텀항 (momentum term)
    - 미니배치 크기(mini batch size)

## 딥러닝을 적용하는 것은 매우 경험적인 과정



- 경험적인 과정 = 많은 것을 시도하고 작동되는지를 확인
  - ex) 학습률  $\alpha$  값으로 다양한 값을 시도
    - 학습을 빠르게 하고 더 낮은 비용함수 J로 수렴하면 해당 값 사용

## 8. 인간의 뇌와 어떤 연관이 있을까요?



신경망과 인간의 뇌 간의 연관은 그렇게 크지 않다.

그렇다면 사람들이 왜 딥러닝과 인간의 뇌를 비유하는 것일까?

- 신경망의 복잡한 과정은 직관적으로 전달 어렵
- 단순화해서 뇌세포의 프로세스로 비유하게 되면, 사람들에게 조금 더 직관적/효과적으로 전달 가능
- 그러나 최근에는 이런 비유가 점점 무너져 가고 있음
  - 오늘날 신경 과학자들조차도 하나의 뉴런이 무엇을 하는지 거의 모름
  - 하나의 뉴런은 훨씬 더 복잡하고 알기 어려움
  - 뉴런이 신경망 처럼 역전파를 통해서 학습 하는지도 의문이기 때문