3주차

파이썬의 브로드 캐스팅

```
import numy as np
A=np.array([[56.0, 0.0, 4.4, 68.0],[1.2,104.0, 52.0, 8.0],[1.

cal=A.sum(axis=0)
percentage=100*A/cal.reshape(1,4)
```

- axis=0: 파이썬에게 세로로 더하라고 알려주는 것.
- 파이썬의 행렬이 확실하지 않을 때, reshape 함수를 사용하여 필요한 차원으로 확실하게
- (m,n) +,-,*,/ (1,n)

  → (1,n) 행렬을 m번 복사해서 (m,n)과 모양 맞춰주기
- 행렬이 아니어도 복사해서 연산할 수 있는 모양으로 맞춰줌.

넘파이 패키지 주의 사항

→ 명확하게 차원을 정의하지 않고 사용하는 것을 피하자

```
a=np.random.randn(5) #사용하지 말 것

a=np.random.randn(5,1) #a.shape=(5,1)
a=np.random.randn(1,5) #a.shape=(1,5)
```

- 코드의 차원이 분명하지 않을 때, assert 함수를 사용하여 확인 후 reshape 사용

```
assert(a.shape==(5,1))
```

- 문제가 생기는 이유

```
print(a)
```

```
[ 0.50290632 -0.29691149  0.95429604 -0.82126861 -1.46269164]
```

→ 그냥 `np.random.randn(5)` 라고 하면 랭크가 1인 배열이 나옴.

→ 전치를 한 것을 내적하면 그냥 값 하나로 나옴.

```
a = np.random.randn(5,1)
print(a)
```

```
[[-0.0967311 ]
 [-2.38617377]
 [-0.3243588 ]
 [-0.96216349]
 [ 0.54410384]]
```

→ `np.random.randn(5,1)` 로 작성할 경우 랭크가 2가 나옴.

→ 전치를 한 것을 내적하면 행렬로 나오게 됨.

로지스틱 회귀의 비용함수 설명

- y값이 1 이 될 확률 : $P(y=1|x) = \hat{y}$
- y값이 0 이 된 확률 : $P(y=0|x) = 1-\hat{y}$

→ 줄여서 한번에 $P(y|x) = \hat{y}^{y}(1-\hat{y})^{(1-y)}$

- 로그함수는 강한 단조증가함수이기에 $logp(y|x)$ 를 최대화→ $p(y|x)$를 최대
- 손실함수를 최소화→ 확률의 로그값 $logP(y|x)$을 최대화→ -1 을 곱한 확률 $-logP(y \mid x)$ 를 최소화
- 손실함수의 정의 :

  → $L(\hat{y}, y) = -\log P(y|x) = -log(\hat{y}^{y}(1-\hat{y})^{(1-y)})$
- 비용함수(m개의 examples에 대한)

  → 훈련샘플들이 독립동일분포라고 가정할 때,

  → 전체 샘플의 확률은 각 샘플의 확률의 곱으로 정의
  - 최대 우도 추정: Maximum Likelihood Estimation, MLE

- $P(\text{labels in training set}) = \prod_{i=1}^{m} P(y^{(i)}|x^{(i)})$
- $\log P(\text{labels in training set}) = \log \prod_{i=1}^{m} P(y^{(i)}|x^{(i)}) = -\sum_{i=1}^{m} L(\hat{y}^{(i)}, y^{(i)})$

$J(w,b) = -\log P(\text{labels in training set}) = \frac{1}{m}\sum_{i=1}^{m} L(\hat{y}^{(i)}, y^{(i)})$

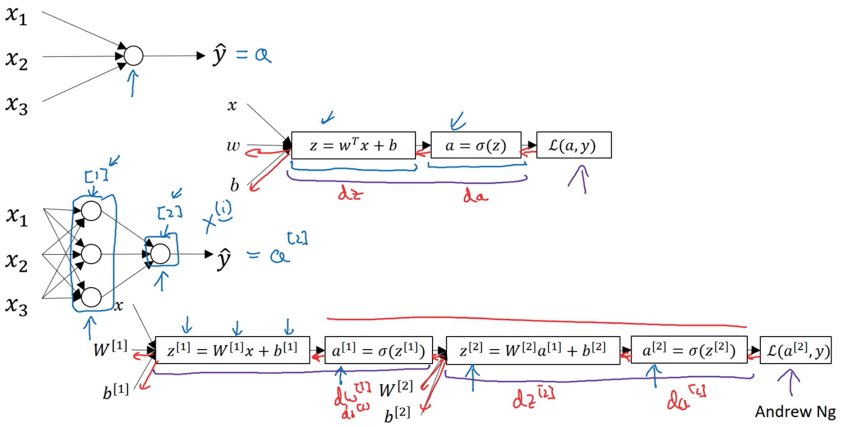
→ 스케일을 맞추기 위해서 m으로 나눠주기

얕은 신경망 네트워크

신경망 네트워크 개요

로지스틱 회귀 vs 신경망

→ z와 a를 한번 계산 vs 여러번 계산

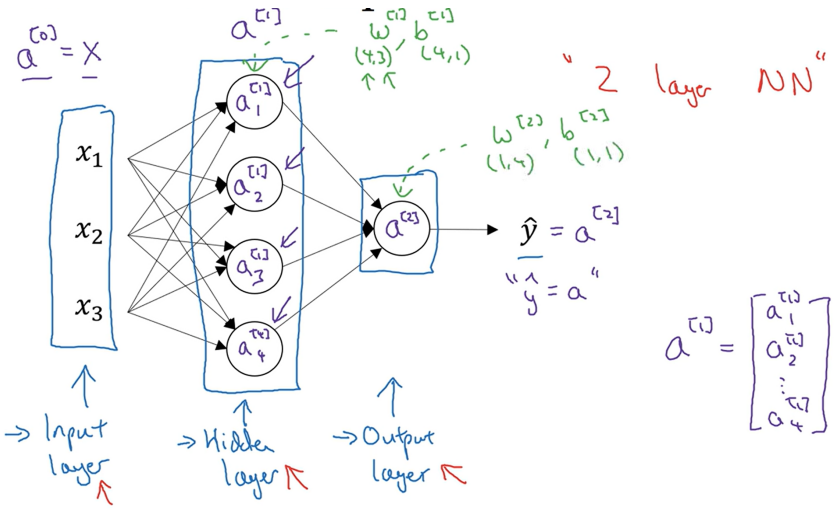


신경망 네트워크의 구성 알아보기

→ 신경망의 층을 셀 때 입력층은 세지 않음.

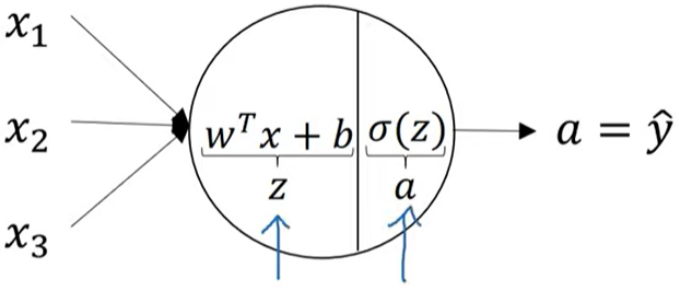
→ 입력층, 은닉층, 출력층(입력층과 다르게 은닉층과 출력층은 연관된 매개변수가 있음.)

→ 매개변수의 차원: 행렬로 봤을 때(a,b): a는 현재 층 노드의 개수, b는 이전 층 노드의 개수



신경망 네트워크 출력의 계산

- 노드에서의 두 단계 계산



$$z = w^T x + b$$

$$a = \sigma(z)$$

- $a_i^{[l]}$: l : 몇 번째 층인지 의미, i 해당 층에서 몇 번째 노드인지 의미
- 열벡터: 열을 늘리는 방향으로 벡터 형성

많은 샘플에 대한 벡터, 벡터화 구현에 대한 설명

idea) 샘플이 여러 개일 경우, for문으로 계산을 하게 되면 계산량이 많아질 수 있다.→ 벡터화를 통해 해결 가능

```
for i = 1 to m:
```

$$z^{[1](i)} = W^{[1]}x^{(i)} + b^{[1]}$$

$$a^{[1](i)} = \sigma(z^{[1](i)})$$

$$z^{[2](i)} = W^{[2]}a^{[1](i)} + b^{[2]}$$

$$a^{[2](i)} = \sigma(z^{[2](i)})$$

→ $x^{(i)}$를 벡터로 만들자!