

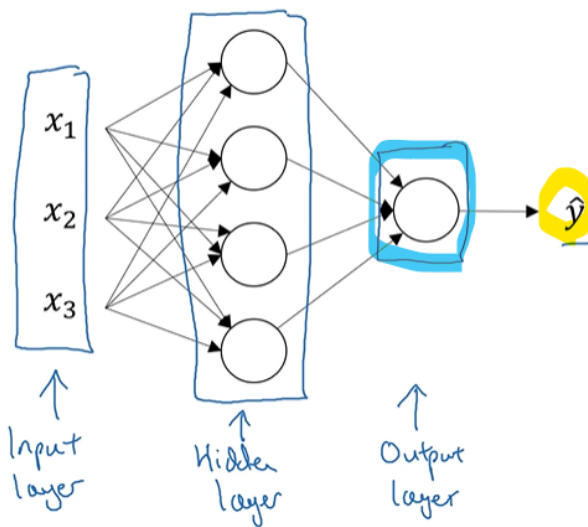
[Week3]_이원주

numpy 쓸 때 오류 줄이는 Tip!

- reshape을 생활화하자.
 - shape을 확실히 아는 게 아니면 그냥 한 번 더 써주자. 이런 건 확실히 하는 게 좋음.
 - 게다가 비용도 적은데 왜 안 써? $\text{cost} = O(n)$
- shape이 헛갈리면 `assert(a.shape == (1,5))` 함수로 찍어보자.
 - 확인은 자주자주!
- rank가 1인 배열은 아예 사용하지 말자.
 - shape이 (4,)처럼 (n,) 꼴인 배열.
 - 행 벡터도, 열 벡터도 아님. ⇒ 직관적이지 않은 결과가 나올 수 있음.

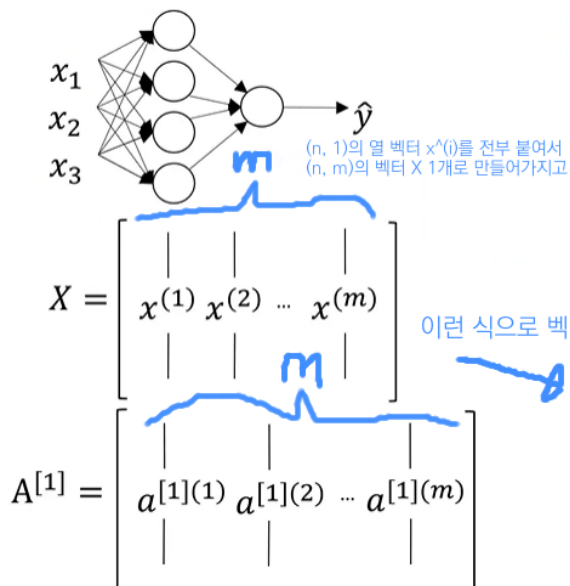
얕은 신경망 네트워크

신경망 네트워크



- Output Layer
 - 노란 형광펜 (X)
 - 파란 형광펜 (O)
- Layer 수 셀 때
 - Input Layer는 빼고 셈.
 - Input Layer만 node로 표현하지 않고, w, b도 없음.

벡터화



for i = 1 to m ← 원래 이걸 거를

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

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

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

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

$$Z^{[1]} = W^{[1]}X + b^{[1]}$$

$$A^{[1]} = \sigma(Z^{[1]})$$

$$Z^{[2]} = W^{[2]}A^{[1]} + b^{[2]}$$

$$A^{[2]} = \sigma(Z^{[2]})$$

X = $A^{[0]}$ 이라서
Layer1, 2만 다르고
사실상 코드가 같음.