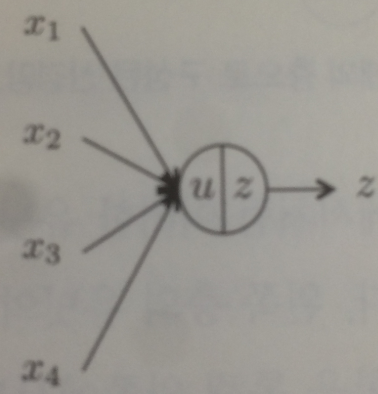
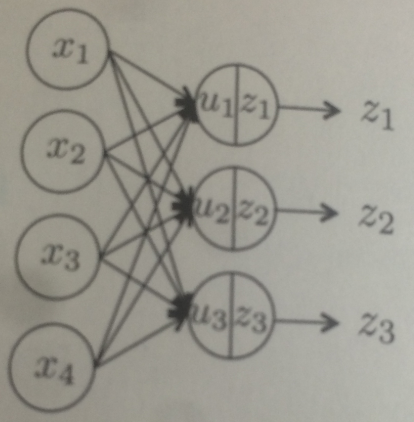
# 1 node / 1 layer



* f : activation function

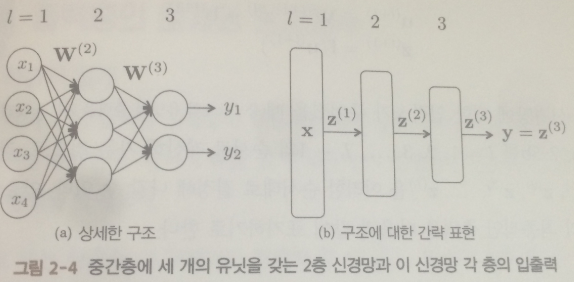
# multi node / 1 layer



* : ith @ l 🡪 jth @ (l + 1)

= +

# multi node / multi layer



At layer 2, sum of inputs

= +

At layer 2, outputs

At layer 3,

At layer ( l + 1)

# Loss function

Training data of size N with input dimension I, ouput dimension K(or K classes)

where,

nth 입력과 현재의 weight에 대한, 모델의 출력값

# Mean Squared Error

# 머신 러닝의 목표

주어진 E(w)에 대해,

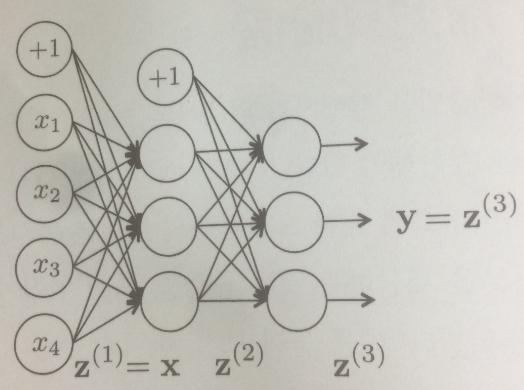
이 되는 w를 구하는 것.

# Gradient descent

(t+1)th epoch시 가중치 업데이트

Where = learning rate

# Back propagation



(bias를 0th 노드로 포함)

jth node at layer l, let =

**Ex) regression with 2 layers, activation function f, identity map as output function, MSE as loss function**

Output of layer 2 of jth node,

At Output layer, jth output is

Then loss function is,

**Gradient of loss function is..**

At layer 3, for jith weight, 미분값은,

At layer 2, for jith weight, 미분값은,

**Generalization**

At layer l,

Where delta is,

아래는 위의 공식에서 해당하는 부분을 간략하게 설명하기 위한 것

아래는 실제 공식

# 학습 프로세스

