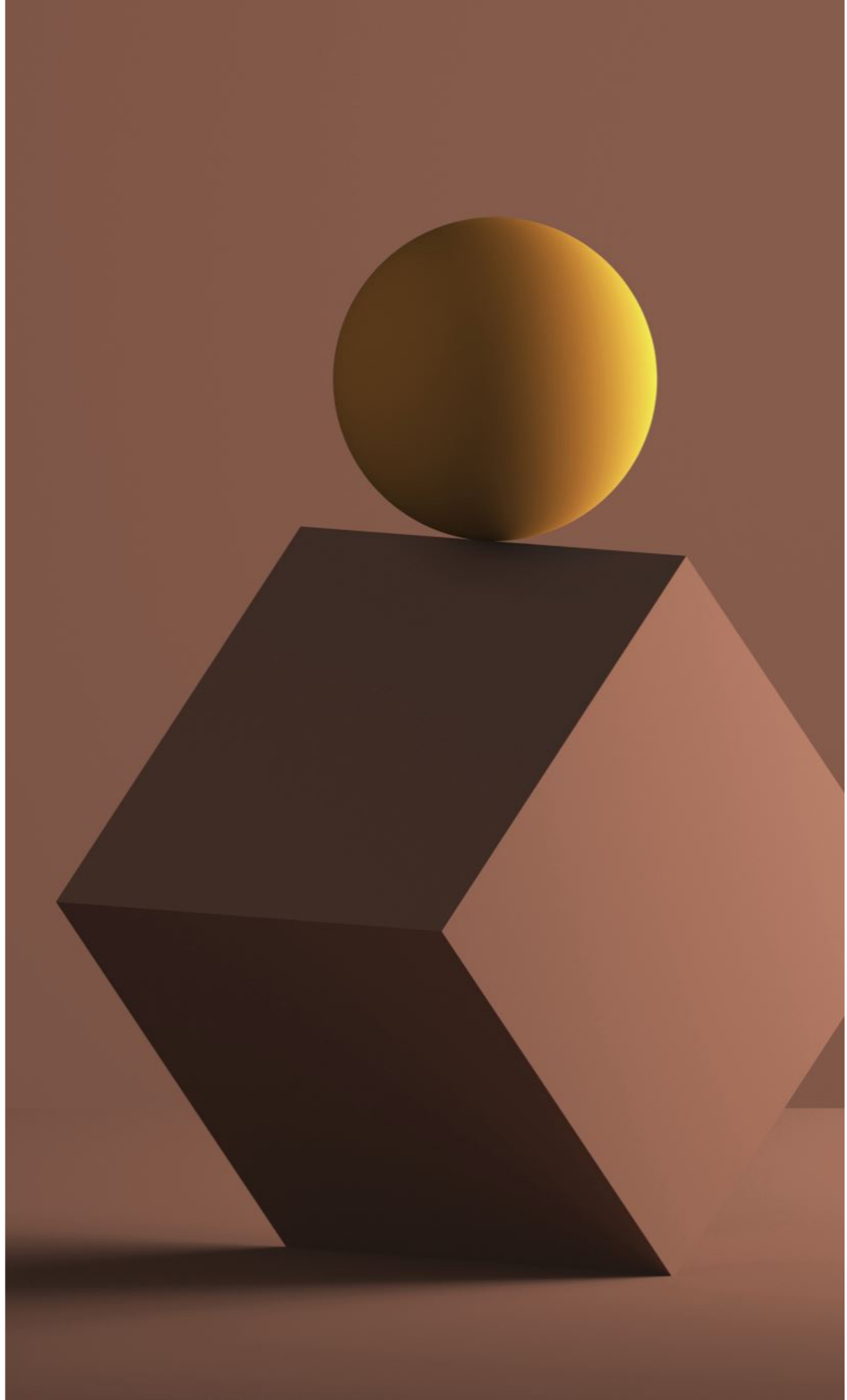
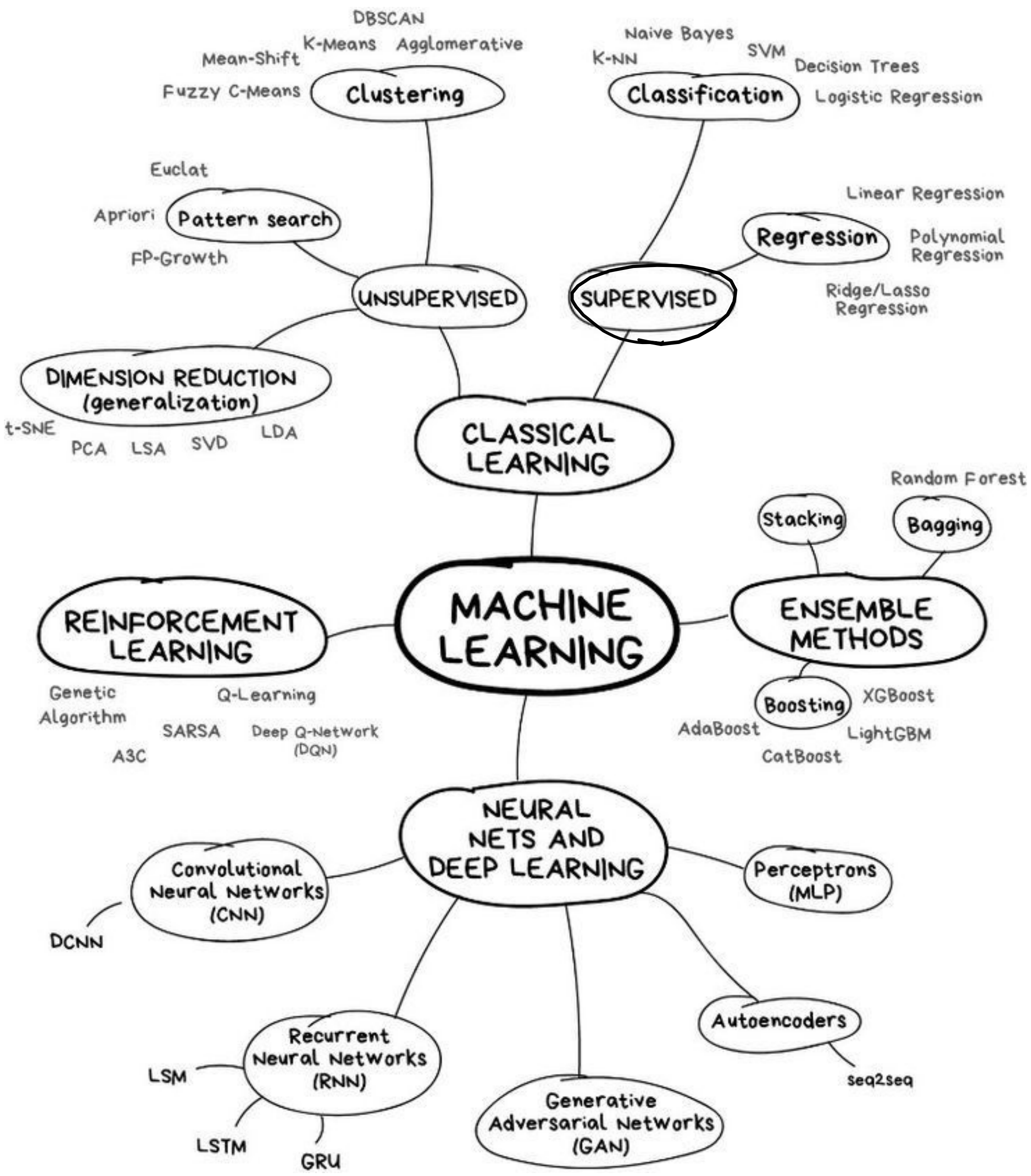
Basic models (2)

Artificial Intelligence

Woohwan Jung




MACHINE LEARNING
CLASSICAL LEARNING
UNSUPERVISED
SUPERVISED
Clustering
DBSCAN
K-Means
Agglomerative
Mean-Shift
Fuzzy C-Means
Classification
Naive Bayes
SVM
K-NN
Decision Trees
Logistic Regression
Pattern search
Euclat
Apriori
FP-Growth
Regression
Linear Regression
Polynomial Regression
Ridge/Lasso Regression
DIMENSION REDUCTION (generalization)
t-SNE
PCA
LSA
SVD
LDA
ENSEMBLE METHODS
Stacking
Bagging
Random Forest
Boosting
XGBoost
AdaBoost
LightGBM
CatBoost
REINFORCEMENT LEARNING
Genetic Algorithm
Q-Learning
SARSA
Deep Q-Network (DQN)
A3C
NEURAL NETS AND DEEP LEARNING
Convolutional Neural Networks (CNN)
DCNN
Perceptrons (MLP)
Recurrent Neural Networks (RNN)
LSM
LSTM
GRU
Generative Adversarial Networks (GAN)
Autoencoders
seq2seq

	Linear Regression	Logistic Regression
Problem	Regression	Classification
Model	$\hat{y} = \boldsymbol{w}^{\top}\boldsymbol{x} + b$ Parameters: $\boldsymbol{w} \in \mathbb{R}^n$, $b \in \mathbb{R}$	$\hat{y} = \sigma\left(\boldsymbol{w}^{\top}\boldsymbol{x} + b\right)$ Parameters: $\boldsymbol{w} \in \mathbb{R}^n$, $b \in \mathbb{R}$
Loss	Squared Error $L(y, \hat{y}) = (y - \hat{y})^2$	Binary Cross Entropy (BCE) $L(y, \hat{y}) = -y \log \hat{y} - (1 - y) \log(1 - \hat{y})$

Cost function: $J(\boldsymbol{w}, b) = \frac{1}{m} \sum_{i=1}^{m} L\left(\hat{y}^{(i)}, y^{(i)}\right)$

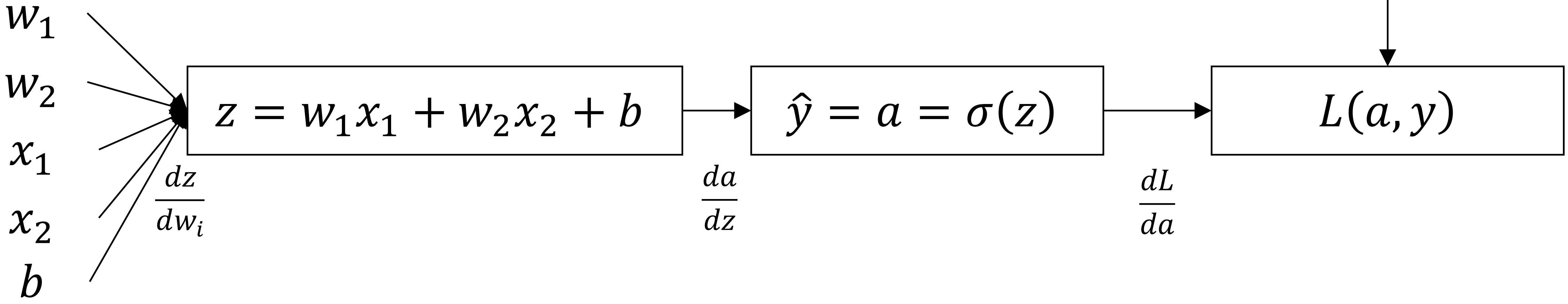
Gradient Descent
:Linear Regression

$$L(a, y) = -y \log a - (1 - y) \log(1 - a)$$

Logistic Regression Recap

For the simplicity

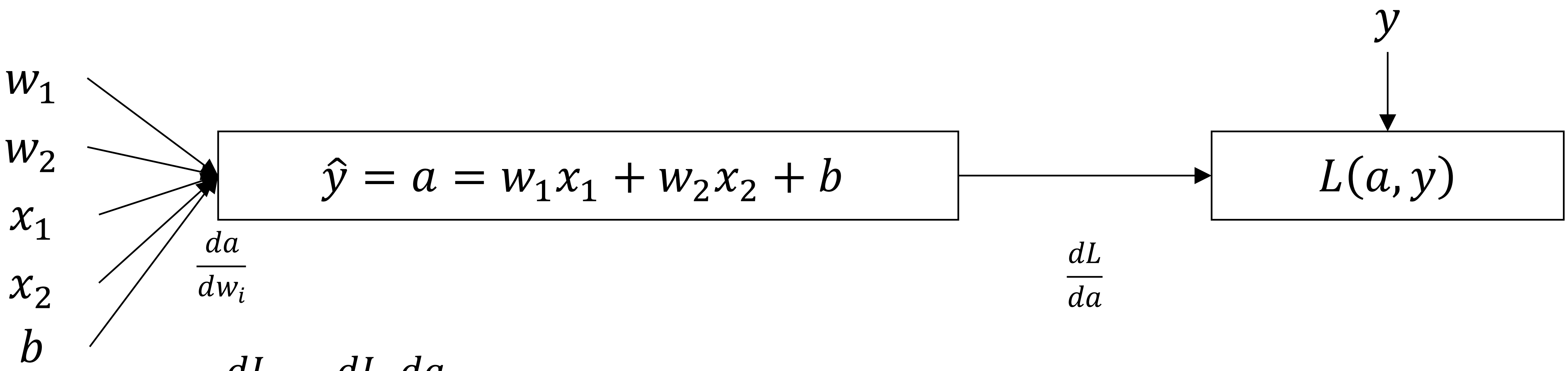
$$\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \boldsymbol{w} = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}$$



$$\frac{dL}{dw_i} = \frac{dL}{da} \frac{da}{dz} \frac{dz}{dw_i}$$

$$L(a, y) = (y - a)^2$$

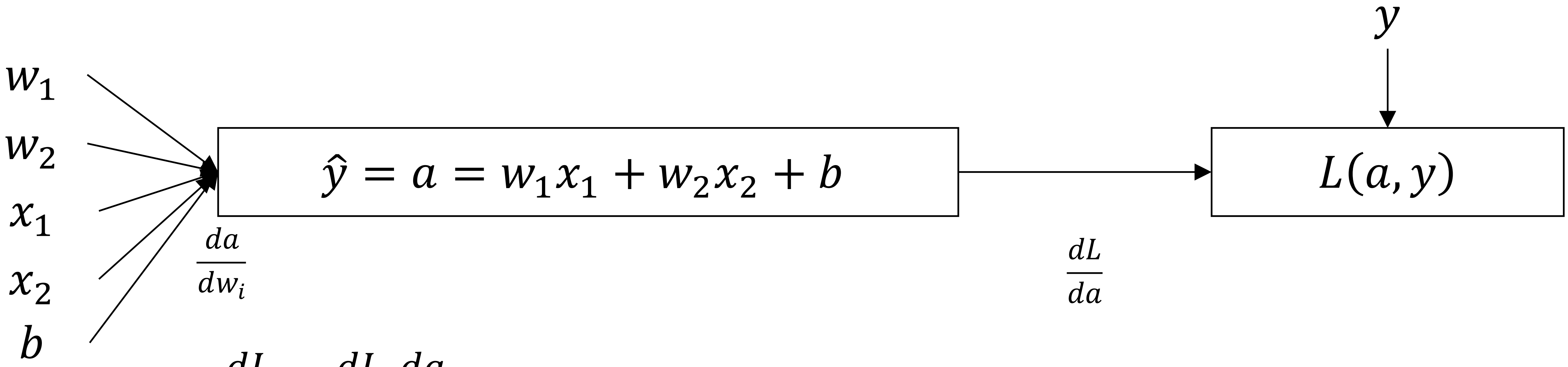
Linear Regression Recap



$$\frac{dL}{dw_i} = \frac{dL}{da}\frac{da}{dw_i} = -2(y - a)x_i$$

$$L(a, y) = (y - a)^2$$

Linear Regression Recap



$$\frac{dL}{dw_i} = \frac{dL}{da}\frac{da}{dw_i} = -2(y - a)x_i$$

$$\frac{dL}{db} = \frac{dL}{da}\frac{da}{db} = -2(y - a)$$

Gradient descent on m examples

$$J(\boldsymbol{w}, b) = \frac{1}{m}\sum_{i=1}^{m} L\big(\hat{y}^{(i)}, y^{(i)}\big)$$

$$\frac{d}{dw_k}J(\boldsymbol{w}, b) = \frac{1}{m}\sum_{i=1}^{m} \frac{d}{dw_k} L\big(\hat{y}^{(i)}, y^{(i)}\big) = \frac{2}{m}\sum_{i=1}^{m}(a - y)x_k$$

$$\frac{d}{db}J(\boldsymbol{w}, b) = \frac{1}{m}\sum_{i=1}^{m} \frac{d}{db} L\big(\hat{y}^{(i)}, y^{(i)}\big) = \frac{2}{m}\sum_{i=1}^{m}(a - y)$$

Gradient descent for training a Linear Regression model

- Randomly Initialize $\boldsymbol{w}, b$
- lr = 0.1
- For e = 1 to n_{epoch}:
  - J = 0; d_w1 = 0; d_w2 = 0; d_b=0
  - For i= 1 to m:
    - $z = w_1 x_1^{(i)} + w_2 x_2^{(i)} + b$
    - $a = \sigma(z)$
    - d_w1 $+= 2(a - y)x_1^{(i)}$
    - d_w2 $+= 2(a - y)x_2^{(i)}$
    - d_b += 2($a - y$)
  - $w_1 -= lr \, * d\_w1$/m
  - $w_2 -= lr \, * d\_w2$/m
  - $b -= lr \, * d\_b$/m

$$\frac{dL}{dw_k} = 2(a - y)x_k$$

$$\frac{dL}{db} = 2(a - y)$$