# 머신러닝 강의 03
# 기초 모델 구현

# Perceptron

**Contents**

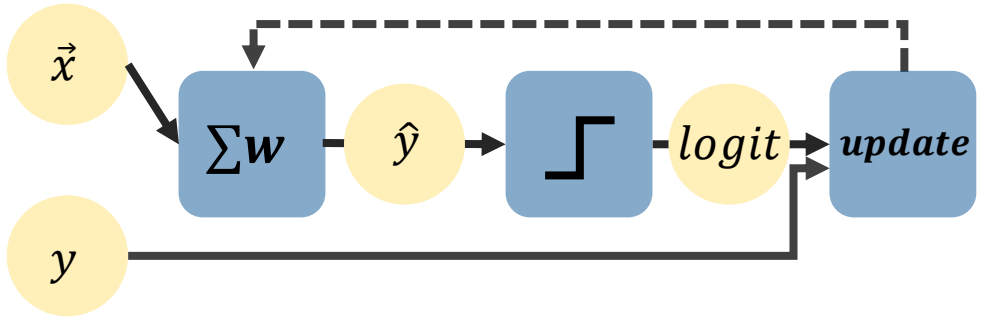
선형 함수

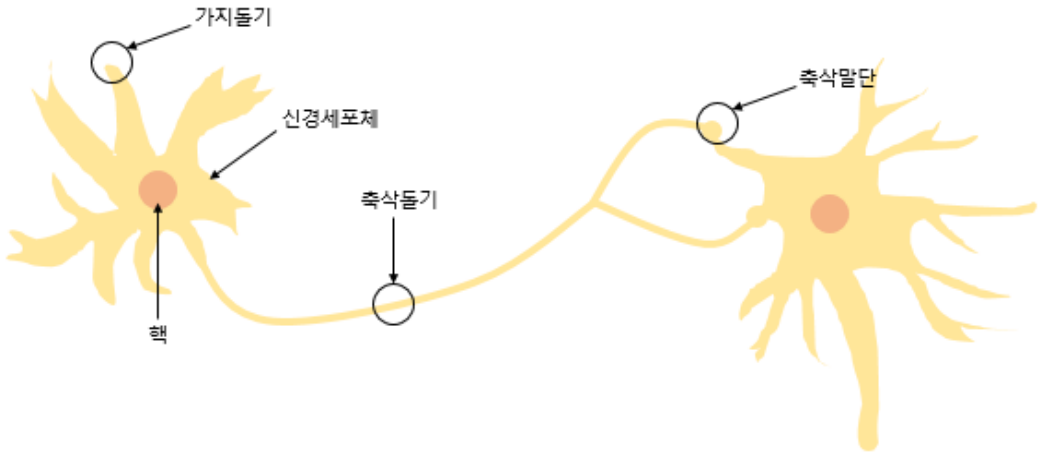
결정 함수

최적화

평가

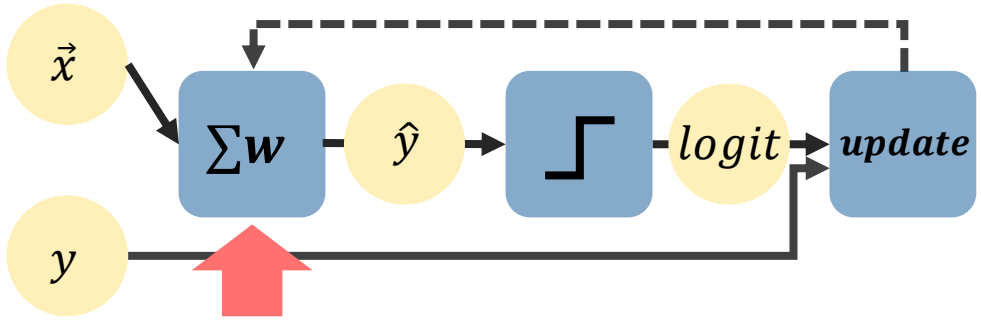
# 소개




```
class Perceptron():
  def __init__(self, lr=0.001, random_state=1):
    self.lr=lr
    self.seed = np.random.RandomState(random_state)
```

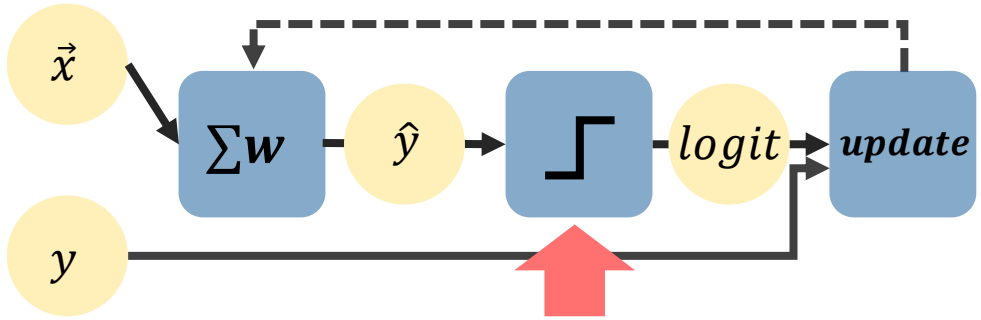
# 모델 구현 – 선형 함수

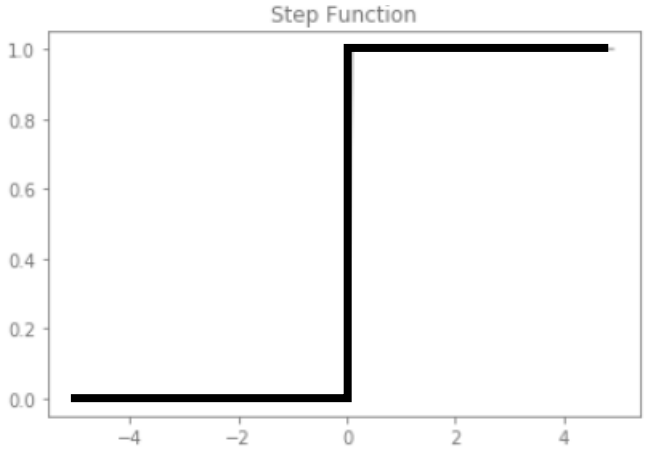



$$f(x) = \vec{w} \cdot \vec{x} + b$$
$$= w_1 x_1 + w_2 x_2 + \ \ldots w_m x_m + b$$

```
def init_weights



def model



```

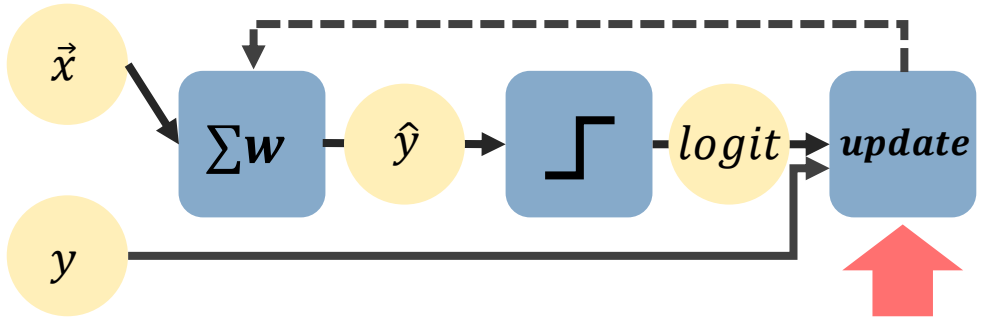
# 모델 구현 – 결정 함수




```
def predict
```

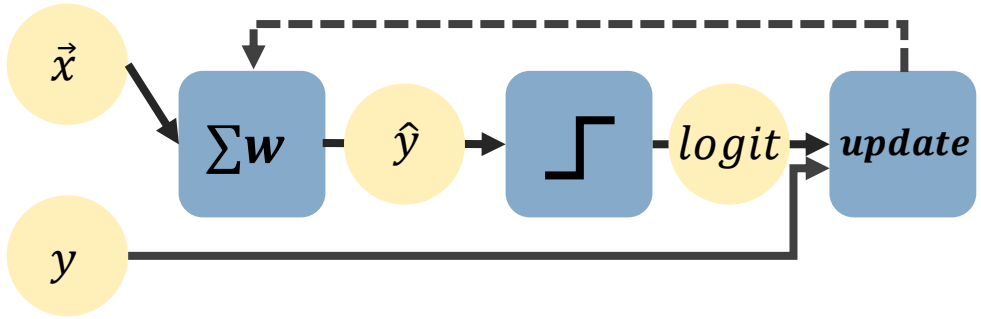
# 모델 구현 – 최적화




$$w_j := w_j + \Delta w_j$$
$$= w_j + \eta\left(y^{(i)} - logit^{(i)}\right)x_j^{(i)}$$

```
def fit
```

# 모델 구현 – 평가




$$accuracy = \frac{\text{정답을 맞춘 데이터 수}}{\text{전제 데이터 수}}$$

```
def evaluate
```

```
def loss
```

```
def accuracy
```

# Adaline

**Contents**

선형 함수

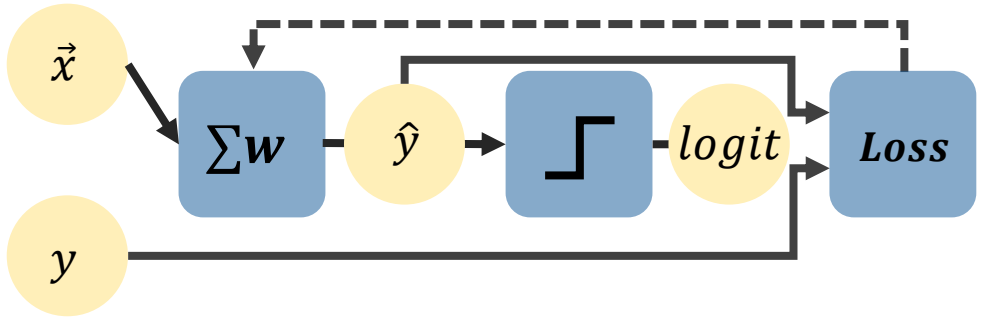
결정 함수

손실 함수

최적화

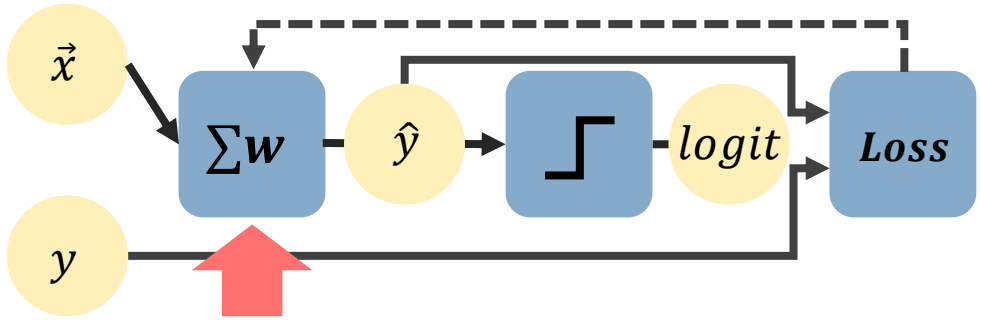
평가

# 소개




```
class Perceptron():
  def __init__(self, lr=0.001, random_state=1):
    self.lr=lr
    self.seed = np.random.RandomState(random_state)
```

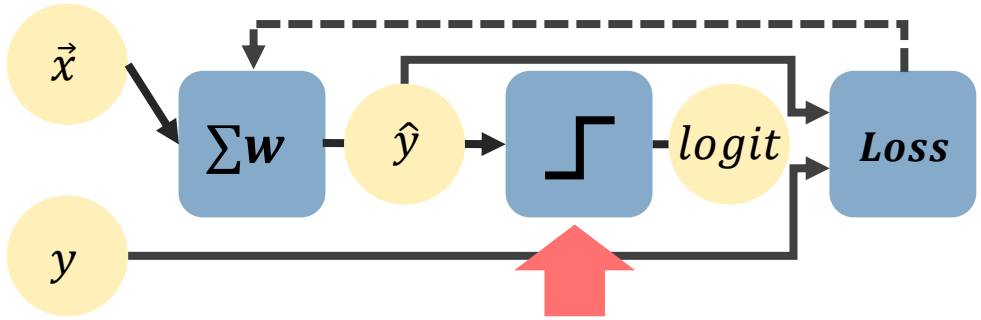
# 모델 구현 – 선형 함수

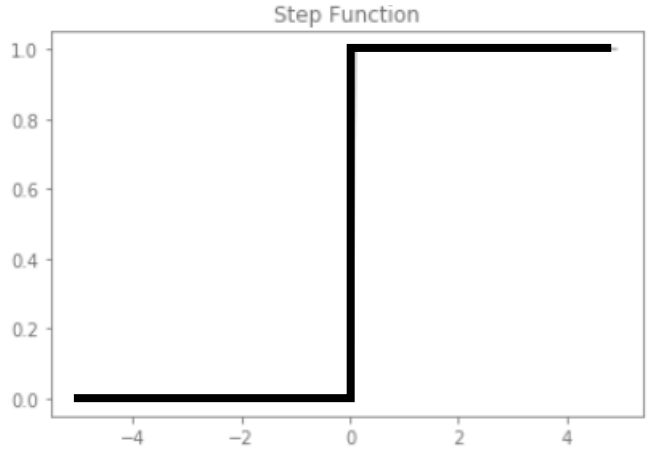



$$f(x) = \vec{w} \cdot \vec{x} + b$$
$$= w_1 x_1 + w_2 x_2 + \ \ldots w_m x_m + b$$

```
def init_weights



def model



```

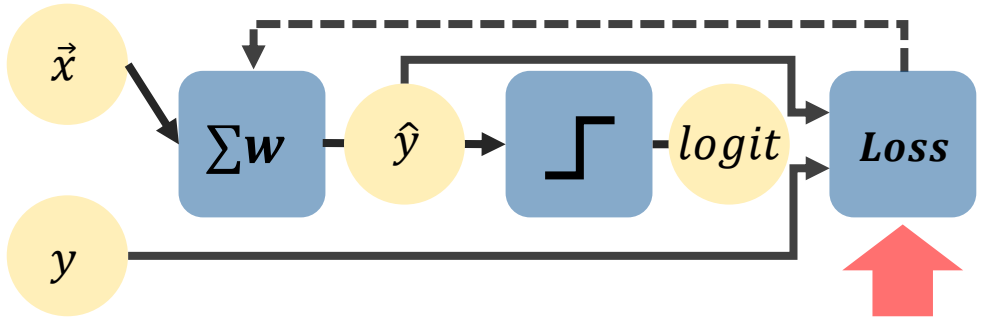
# 모델 구현 – 결정 함수




```
def predict
```

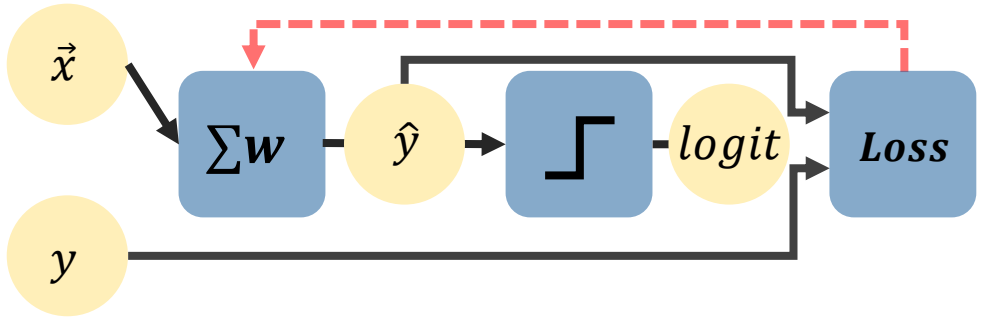
# 모델 구현 – 손실 함수




$$L(w) = \frac{1}{2}\sum_{i}(y^{(i)} - \hat{y}^{(i)})^2$$

```
def loss
```

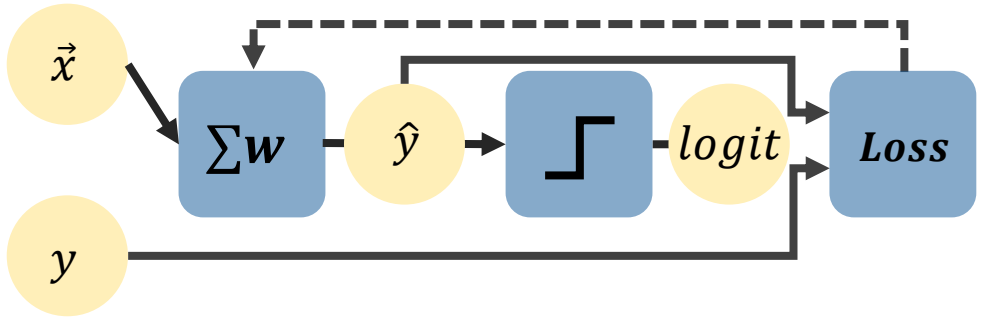
# 모델 구현 – 최적화




$$w_j := w_j + \Delta w_j$$
$$= w_j - \eta \nabla L(w_j)$$
$$= w_j + \eta \sum_i (y^{(i)} - \hat{y}^{(i)}) x_j^{(i)}$$

```
def fit
```

# 모델 구현 – 평가




$$accuracy = \frac{\text{정답을 맞춘 데이터 수}}{\text{전제 데이터 수}}$$

```
def evaluate
```

# 선형회귀

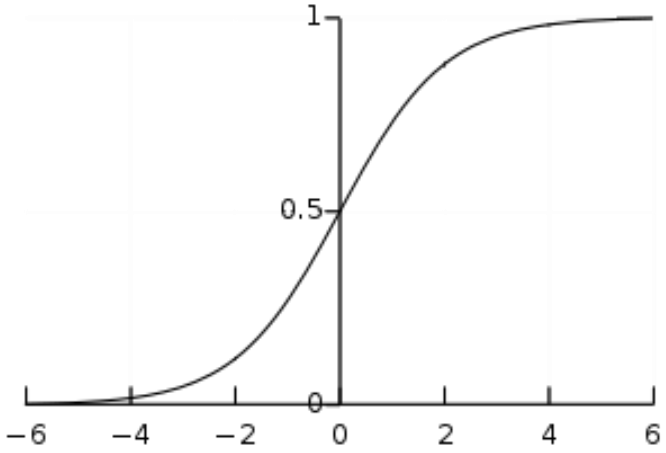
**Contents**

Linear regression

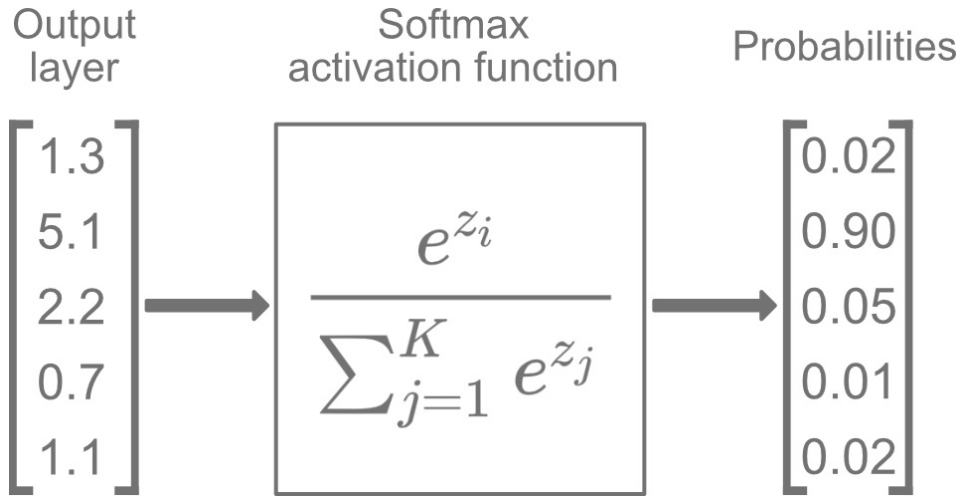
Rogistic regression

Softmax regression

# 모델 구현 – 평가




$$S(x) = \frac{1}{1 + e^{-x}} = \frac{e^{x}}{e^{x} + 1}$$




```
def sigmoid
```

```
def softmax
```

# 함수의 인풋과 아웃풋

linear

sigmoid

softmax

# 회귀

linear

sigmoid

softmax

Linear regression

# 이진분류

linear

sigmoid

softmax

_____ regression

# 다중 클래스 분류

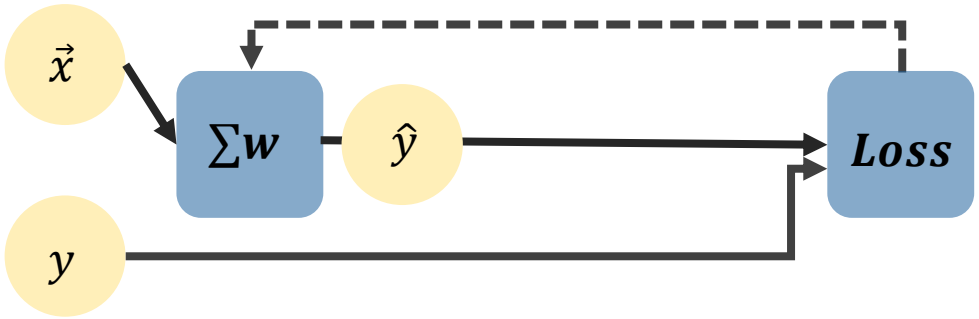
linear

sigmoid

softmax

_____ regression

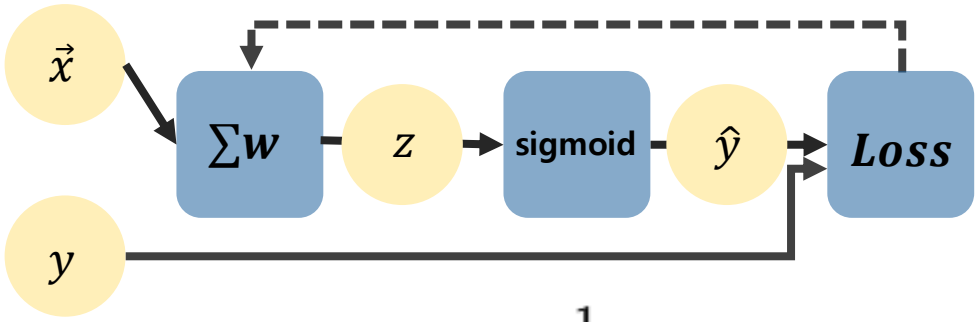
# 모델 구현 – linear regression




$$f(x) = \vec{w} \cdot \vec{x} + b$$
$$= w_1 x_1 + w_2 x_2 + \ \ldots w_m x_m + b$$

$$L(w) = \frac{1}{2} \sum_i (y^{(i)} - \hat{y}^{(i)})^2$$

# 모델 구현 – logistic regression

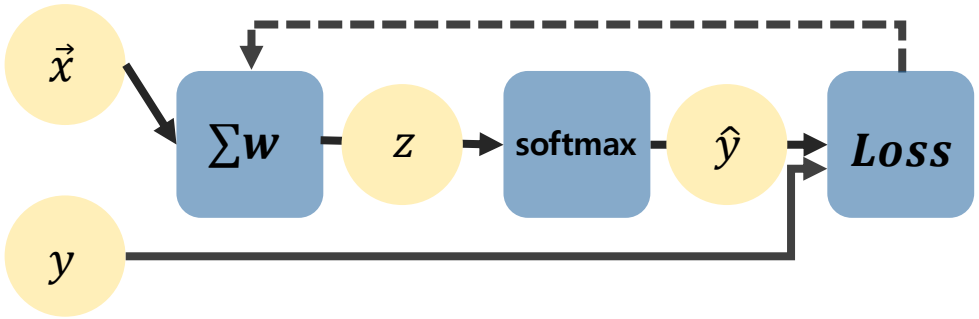



$$\frac{1}{1+e^{-x}}$$

$$f(x) = \vec{w} \cdot \vec{x} + b$$
$$= w_1 x_1 + w_2 x_2 + \ldots w_m x_m + b$$

$$L = -\frac{1}{n}\sum_{i=1}^{n}[y^{(i)} \log \hat{y} + (1 - y^{(i)}) \log(1 - \hat{y})]$$

# 모델 구현 – softmax regression




$$f(x) = \vec{w} \cdot \vec{x} + b$$
$$= w_1 x_1 + w_2 x_2 + \ldots w_m x_m + b$$

$$\frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$$

$$L = -\frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{k} y_j^{(i)} \, log(p_j^{(i)})$$