

로지스틱 회귀와 퍼셉트론

-모두를 위한 딥러닝 시즌2

한신대학교 AINC Lab
노진산

2024.11.18(월)

목 차

- ▶ 로지스틱 회귀
- ▶ 시그모이드 함수
- ▶ 시그모이드 함수 만들어진 과정
- ▶ 퍼셉트론

로지스틱 회귀(Logistic Regression)

임계값을 기준으로 데이터를 분류, 실제 값과 비교로 예측 성능을 향상시킬 수 있다

```
In [18]: x_data = [[1, 2], [2, 3], [3, 1], [4, 3], [5, 3], [6, 2]]
          y_data = [[0], [0], [0], [1], [1], [1]]
          x_train = torch.FloatTensor(x_data)
          y_train = torch.FloatTensor(y_data)

In [19]: # 모델 초기화
          W = torch.zeros((2, 1), requires_grad=True)
          b = torch.zeros(1, requires_grad=True)
          # optimizer 설정
          optimizer = optim.SGD([W, b], lr=1)

          nb_epochs = 1000
          for epoch in range(nb_epochs + 1):

              # Cost 계산
              hypothesis = torch.sigmoid(x_train.matmul(W) + b) # or .mm or @
              cost = -(y_train * torch.log(hypothesis) +
                       (1 - y_train) * torch.log(1 - hypothesis)).mean()

              # cost로 H(x) 개선
              optimizer.zero_grad()
              cost.backward()
              optimizer.step()

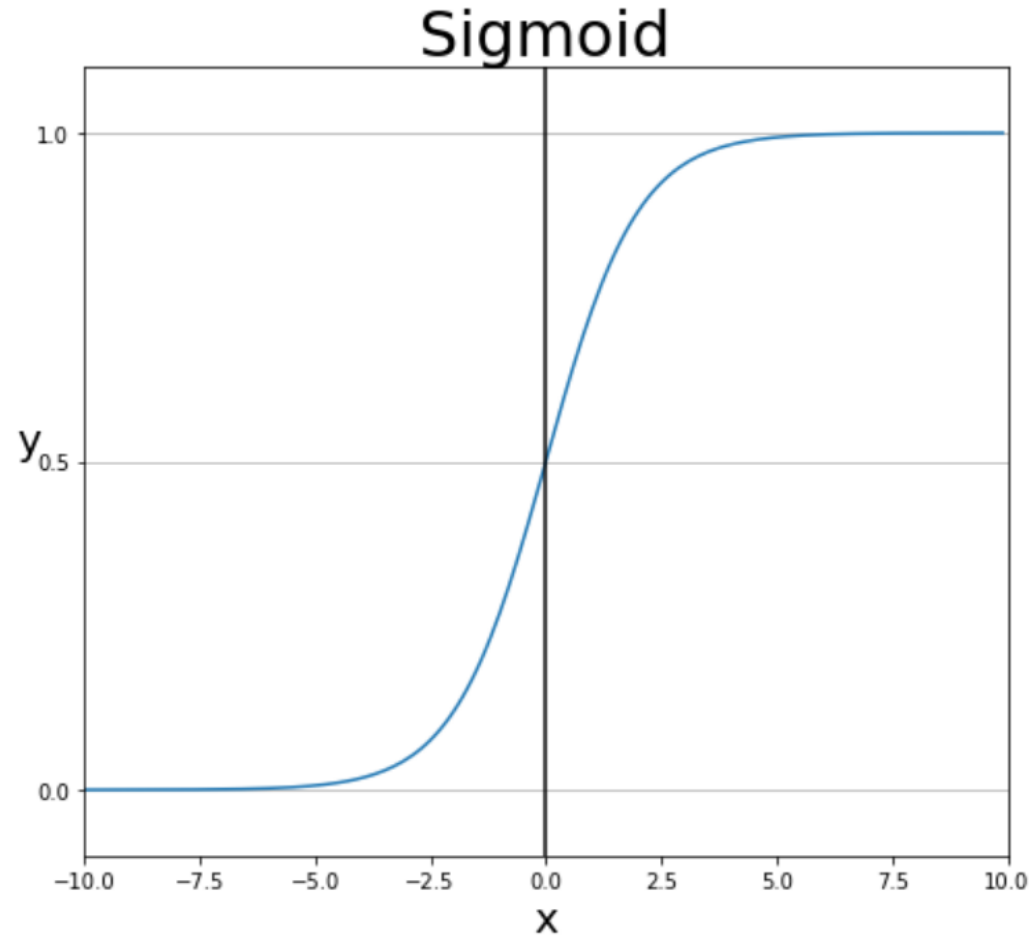
              # 100번마다 로그 출력
              if epoch % 100 == 0:
                  print('Epoch {:4d}/{:4d} Cost: {:.6f}'.format(
                      epoch, nb_epochs, cost.item()
                  ))
```

```
Epoch 0/1000 Cost: 0.693147
Epoch 100/1000 Cost: 0.134722
Epoch 200/1000 Cost: 0.080643
Epoch 300/1000 Cost: 0.057900
Epoch 400/1000 Cost: 0.045300
Epoch 500/1000 Cost: 0.037261
Epoch 600/1000 Cost: 0.031673
Epoch 700/1000 Cost: 0.027556
Epoch 800/1000 Cost: 0.024394
Epoch 900/1000 Cost: 0.021888
Epoch 1000/1000 Cost: 0.019852
```

시그모이드 함수(Sigmoid Function)

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

시그모이드의 함수 식



<https://gooopy.tistory.com/52>

시그모이드 함수 만들어진 과정

$$\frac{p}{1-p}$$

p: 성공 확률

$$\text{Logit}(p) = \log\left(\frac{p}{1-p}\right)$$

로짓 함수

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

시그모이드 함수

$$e^z = \frac{p}{1-p}$$

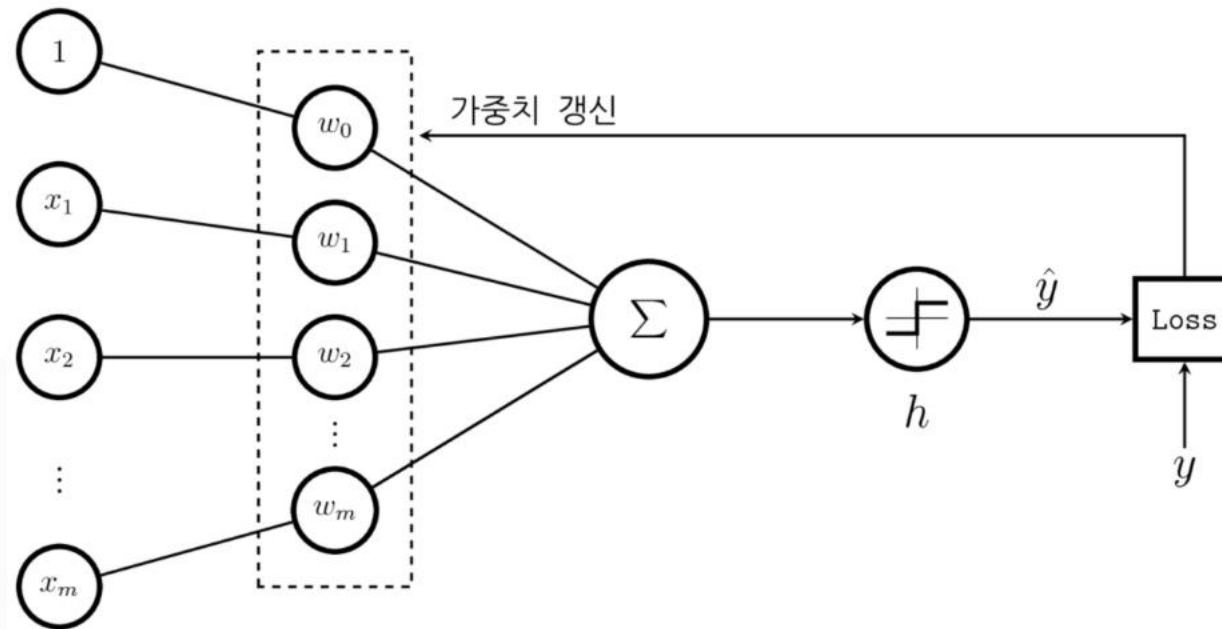
→

$$p = \frac{e^z}{1 + e^z}$$

→

$$p = \frac{1}{1 + e^{-z}}$$

퍼셉트론(Perceptron)



$$w_i = w_i + \Delta w_i$$

$$b = b + \Delta b$$

ai를 이용하여 생성함

https://heung-bae-lee.github.io/2020/04/21/machine_learning_11/