인공신경망 총정리 20241519 조예성

구글에서 만든 딥러닝 전용 라이브러리는?: 텐서플로우 Tensorflow

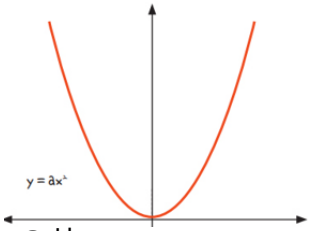
텐서플로의 사용 난이도 문제를 해결해 주기 위한 개발된 것은?: 케라스 Keras

1차 함수 식: y = ax=b

1차 함수 사용: 선형 회귀

2차 함수 식: y= ax^2

2차 함수 그래프 모양:



2차 함수 사용: 평균 제곱 오차, Mean Square Error

오차가 최소화 되는 지점을 찾기 위해서 사용하는 함수는?: 미분

미분을 사용하는 이유: 순간 변화율 구하기 위하여

미분 계수가 의미하는 것: 그래프에서의 기울기

미분의 기본 공식

1 | $f(x) = x$일 때 $f'(x) = 1$

2 | $f(x) = a$에서 a가 상수일 때 $f'(x) = 0$

3 | $f(x) = ax$에서 a가 상수일 때 $f'(x) = a$

4 | $f(x) = x^a$에서 a가 자연수일 때 $f'(x) = ax^{a-1}$

5 | $f(g(x))$에서 $f(x)$와 $g(x)$가 미분 가능할 때 $\{f(g(x))\}' = f'(g(x)) \times g'(x)$

우리가 원하는 한가지 변수만 미분하고 모두 상수로 취급하는 함수의 명칭은? 편미분

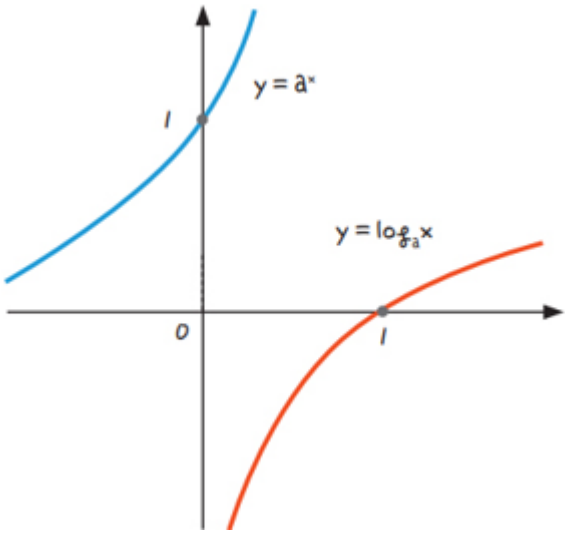
지수함수란? 지수를 구하는 함수

시그모이드 함수의 사용: 활성화 함수

시그모이드 함수는 지수 함수에서 밑 값이 (자연상수 e)인 함수를 의미함.

시그모이드 함수 식: $$f(x) = \frac{1}{1+e^{-x}}$$

로그함수 설명: n제곱했을 때 y가 되는 수



로그함수 그래프:
로그함수 사용: 로지스틱 회귀

Y = ax+b에서 y는? 종속 변수

y = ax+b에서 x는? 독립 변수

하나의 x값으로 y값을 설명할 수 있으면? 단순 선형 회귀

x값이 여러 개 필요하다면? 다중 선형 회귀

오차를 구할 때 가장 많이 사용하는 것?: 평균 제곱 오차

미분 기울기를 사용하는 오차 감소시키는 방법은?: 경사하강법

최솟점 m을 찾아가기 위해 부호를 바꾸어 이동시키는데, 그 거리를 칭하는 명칭은?: 학습률
참 거짓을 판단하는 모델: 로지스틱 회귀 모델
시그모이드 함수의 a가 작아질수록 오차는 (무한대로 커지지만) a가 커진다고 해서 (오차가 없어지지는 않는다)

실습 | 선형 회귀 모델 실습

```
import numpy as np
import matplotlib.pyplot as plt

# 공부 시간 X와 성적 y의 넘파이 배열을 만듭니다.
x = np.array([2, 4, 6, 8])
y = np.array([81, 93, 91, 97])

# 데이터의 분포를 그래프로 나타냅니다.
plt.scatter(x, y)
plt.show()
```

```
# 기울기 a의 값과 절편 b의 값을 초기화합니다.
a = 0
b = 0

# 학습률을 정합니다.
lr = 0.03

# 몇 번 반복될지 설정합니다.
epochs = 2001

# x 값이 총 몇 개인지 셉니다.
n = len(x)
```

```
# 경사 하강법을 시작합니다.
for i in range(epochs):              # 에포크 수만큼 반복합니다.
    y_pred = a * x + b               # 예측 값을 구하는 식입니다.
    error = y - y_pred               # 실제 값과 비교한 오차를 error로 놓습니다.

    a_diff = (2/n) * sum(-x * (error))   # 오차 함수를 a로 편미분한 값입니다.
    b_diff = (2/n) * sum(-(error))       # 오차 함수를 b로 편미분한 값입니다.

    a = a - lr * a_diff   # 학습률을 곱해 기존의 a 값을 업데이트합니다.
    b = b - lr * b_diff   # 학습률을 곱해 기존의 b 값을 업데이트합니다.

    if i % 100 == 0:      # 100번 반복될 때마다 현재의 a 값, b 값을 출력합니다.
        print("epoch=%.f, 기울기=%.04f, 절편=%.04f" % (i, a, b))

# 앞서 구한 최종 a 값을 기울기, b 값을 y 절편에 대입해 그래프를 그립니다.
y_pred = a * x + b

# 그래프를 출력합니다.
plt.scatter(x, y)
plt.plot(x, y_pred,'r')
plt.show()
```

변수 개수가 증가된 선형회귀: 다중 선형 회귀

다중 선형 회귀 식: $y = a_1x_1 + a_2x_2 + b$

평균 제곱 오차의 다른 명칭: 손실 함수 loss function

경사하강법의 다른 명칭: 옵티마이저 optimizer