1. Machine Learning의 개념
   * + 1. 3요소
          1. experience E: 표본
          2. task T: 목표
          3. performance P: 성능
       2. 분류
          1. Supervised Learning(지도학습)

미리 입력해둔 Training Data를 통해 정답을 학습한 후 일반화

* + - * 1. Unsupervised Learning(비지도학습, 자율학습)

Training Data 없이 데이터가 어떻게 구성되었는지 알아내는 것

1. Supervised Learning (지도학습)
   * 1. Linear Regression (선형 회귀 분석)
        1. 기본 개념

i번째 Data의 값

ex) : 2번째 Data의 값

* + - * 1. θ(Theta)(=Parameters)

Parameters를 결정해 식을 완성할 수 있음

적합한 Parameters 값을 찾아내는 것이 목표

* + - * 1. 식

(와 는 각각 n+1차원 vector)

* + - * 1. Cost

실제 결과값과 Regression을 통해 나온 값 사이의 차이값

Cost를 최소화하는 값(Parameters)을 찾아 를 완성하는 것이 목표

* + - 1. Cost Function
         1. 목적: 실제 Data값과 예측값 사이의 차이를 측정하기 위한 함수
         2. Cost Function은 [Dataset의 Data]와 [각각의 Data에 상응하는 예측값]사이의 **제곱오차의 합**

J() =

* + - * 1. Cost Function이 최솟값이 되는 θ들을 찾아야 하기 때문에 Cost Function을 미분해야 함
        2. Cost Function을 미분하기 쉽게 만들기 위해 식 변경 ( 곱함)

J() =

* + - 1. Gradient Descent (경사하강법)
         1. 개념: 함수를 기울어진 경사면이라 가정했을 때, 경사면에 공을 놓으면 공은 해당 지점에서 가장 가파른 방향으로 움직이는데, 이를 끊임없이 반복하면 결국 공은 최저점에 도달한다는 것에 착안한 방법
         2. Cost Function에 적용해 Cost Function이 최솟값이 되는 들을 찾는데 사용
         3. 기본 개념:

(은 매우 작은 양의 정수)

단순화)

증명)

가 최소인 값을 찾아야 하기 때문에 는 최소여야 함

는 와 의 내적

내적의 최솟값은 두 벡터 값의 방향이 반대가 되어야 함.

(은 매우 작은 양의 정수)

* + - * 1. Cost Function에 Gradient Descent적용

J()

J() =

* + - 1. Feature Scaling
         1. 여러 Features가 있을 때, 각각의 Features마다 크기가 다를 수 있음
         2. Features 사이에 크기 차이가 많이 나게 될 경우 Gradient Descent 수행 시 시간 증가
         3. 식:
      2. 값 (Learning Rate 학습률) 설정
         1. 적절한 값일 경우: J()가 점차 감소하는 그래프
         2. 값이 너무 클 경우: J()가 점차 증가 or 증가 및 감소하는 그래프

보폭이 너무 커져 최솟값을 찾지 못할 수 있음

* + - * 1. 값이 너무 작을 경우

보폭이 너무 작아져 최솟값 찾는 데 오랜 시간이 걸림

* + - * 1. 적절한 값 찾는 방법

0.001 < < 1 범위 내에서 선택 (주로 0.001)

3배 단위로 증가시키면서 찾아가기

* + - 1. Normal Equation (정규방정식)
         1. 를 찾는 (Gradient Descent를 대체하는) 방법
         2. 식

(X: [features개수+1] Vector, y: [n+1] Vector)

* + - * 1. Gradient Descent와의 비교

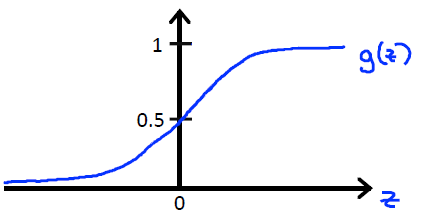
학습률 값 찾을 필요 없음

반복 작업 없이 한 번에 계산 가능 -> 빠름

계산으로 인해 features 개수가 많아질수록 계산 속도 느려짐

(features 개수가 10000 이하일 때 적합)

* + 1. Logistic Regression – Classification
       1. Sigmoid Function (=Logistic Function)
          1. 목적: 넓은 범위의 값을 0~1사이의 값으로 변환하는 값
          2. 식
          3. 그래프



* + - 1. Logistic Regression
         1. 식
         2. 확률의 의미

이면 일 확률이 1, 일 확률이 0

이면 일 확률이 0, 일 확률이 1

* + - * 1. Decision Boundary: 가 0인지 1인지 판단하는 기준선
      1. Cost Function
         1. 또는 두 가지 경우의 수밖에 존재하지 않음
         2. 일 때

이면

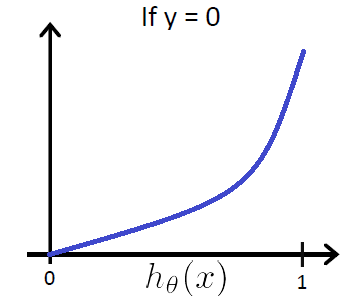
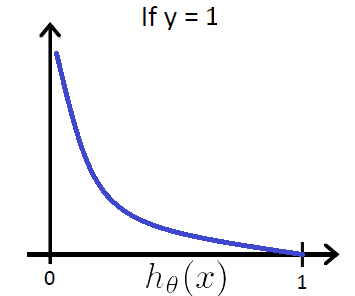
이면

* + - * 1. 일 때

이면

이면

* + - * 1. 그래프: 함수 차용

* + - * 1. 식
      1. Multiclass classification
         1. 이분법이 아닌, 결과값이 여러 개로 분류되는 경우 ()
         2. One vs Rest 방식
         3. 각각의 는 일 확률을 뜻함
         4. 새로운 입력 이 추가되었을 때:

을 모두 파악해본 후, 가장 높은 을 가진 로 을 분류함

* + 1. Regularization (정규화)
       1. 개요
          1. Overfitting(과적합): 최적화가 너무 과도하게 진행되어, 일반화하기 어려운 상태 (새로운 입력이 들어왔을 때 제대로 예측하리라고 보기 어려운 상태)

결과에 큰 영향을 미치지 않는 features를 삭제하면서 features의 수를 줄여서 해결 가능하나 위험성 존재하므로 Regularization을 통해 해결함이 바람직함

* + - * 1. Regularization: 일부 parameters를 작은 값(0에 수렴하게)으로 만들어 를 단순화

정규화 시 상수 parameter(은 포함하지 않음

* + - 1. Linear Regression의 정규화
         1. Linear Regression - Cost Function의 정규화

원리: Cost Function의 뒤에 parameters와 큰 수(lamda)를 곱한 값을 더함

Cost Function을 최소화해야 하는 것이 목적이므로, 계산 결과 해당 parameters는 아주 작은 값을 가질 수밖에 없음

* + - * 1. Linear Regression - Gradient Descent의 정규화

원리: 에 1보다 작은 값을 곱해 를 줄여나감

이므로

* + - * 1. Linear Regression - Normal Equation의 정규화
      1. Logistic Regression의 정규화
         1. Logistic Regression – Cost Function의 정규화
         2. Logistic Regression – Gradient Descent의 정규화

* + 1. Neural Networks
    2. Support Vector Machines

1. Unsupervised Learning
   1. K-means
   2. PCA
   3. Anomaly Detection