

정보검색과 데이터마이닝

hw3 - 머신러닝/
딥러닝 강의 보고서

20142772 최승호

lecture 0 - 강의개요

알파고를 도입부에 보여주며 컴퓨터는 직관적인 것뿐만 아니라 인간의 사고 정도까지 학습할 수 있다는 것을 알려줌

목표

1. 머신러닝에 대한 기본적인 이해

- linear regression, logistic regression에서 나아가
neural network같은 딥러닝 분야

2. 파이썬 텐서플로우를 활용한 문제해결

나머지는 강의에 대한 설명

- 이해하기 쉽다
- 블랙박스형식으로 내부구현을 몰라도 사용할수 있다
- 텐서플로우 사용
- 레퍼런스
- 강의 구성 등

lecture 1 - 머신러닝의 용어와 개념 설명

머신러닝이란?

컴퓨터가 학습해서 배우는 능력을 가지는 것을 연구하는 분야를 머신러닝이라고 함

기본적으로 컴퓨터는 explicit programming(정확하게)을 하는데 실생활에서는 스팸 필터, 자율주행만 해도 경우의 수가 너무 많아 제약이 크다.

그래서 머신러닝이 필요하다.

러닝의 종류는 크게 supervised/unsupervised learning로 나뉘고
 먼저, supervised는 정해진 데이터(training set)로 훈련하는 것이다.
 supervised 예를 들면, 개 고양이, 컵 사진 구분하는 것이다.
 정답을 알려주고 학습을 시켜서 supervised다.



unsupervised learning은 정답이 없이 스스로 학습하는 것이다.
 예로는 구글 뉴스 그룹화, 단어 그룹화

supervised learning은

- image labeling
- email spam filter
- predicting exam score 등등에 쓰인다.

알파고도 supervised learning을 활용한 예이다.

training data set

x와 y가 있는데 여러개의 x중에서 정답 y가 있다. ex) x = 공부시간 y = 점수
 이 데이터를 토대로 기계학습을 시킨다.

이 데이터가 training data set이다.

알파고 같은 경우는 기존에 사람들이 바둑을 둔 것 = training data set

supervised learning의 여러 유형

시험점수 예측 : 시험점수(0 ~ 100)이면 regression

pass/non-pass 예측 : classification 분류하는 것(binary)

grade(A, B, C) 예측 : mulit-label classification

lecture 2 - linear regression

x는 feature라고 부름, y는 결과값

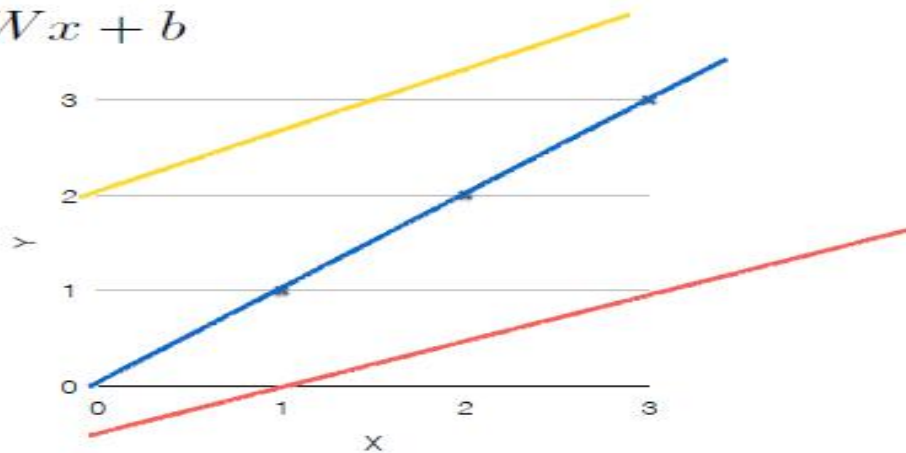
hypothesis 가정

실생활에서 적용되는 경우 많음 ex) 집크기 가격, 공부시간 성적 등 하나의 선으로 표현되는 경우가 regression

$H(x) = Wx + b$ 하나의 선이 될거라고 가설을 세운다는 것임

(Linear) Hypothesis

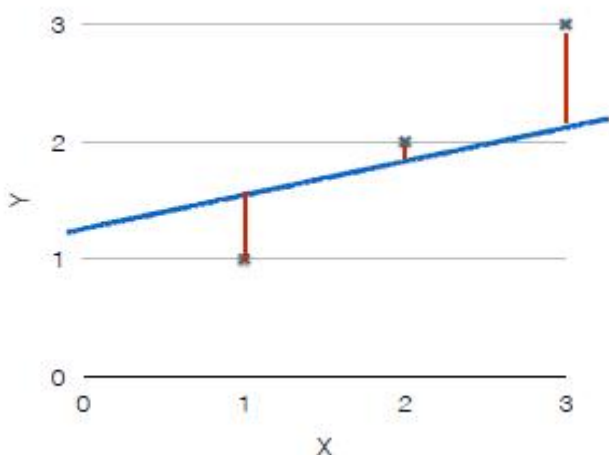
$$H(x) = Wx + b$$



여기서 W와 b를 예측해야함

어느 가설이 더 좋을지는 선과 실제 데이터(점)과의 거리로 안다.

이것을 cost function(loss)이라고 부름



$H(x) - y$ 는 음수일수도 있어서 제곱을함

$$cost = \frac{1}{m} \sum_{i=1}^m (H(x^{(i)}) - y^{(i)})^2 \quad \rightarrow \quad cost(W, b) = \frac{1}{m} \sum_{i=1}^m (H(x^{(i)}) - y^{(i)})^2$$

데이터가 잘 학습되기 위해서 목표는 cost를 minimize(최소화)하는 것에 있다!

lecture 3 - linear regression cost 최소화

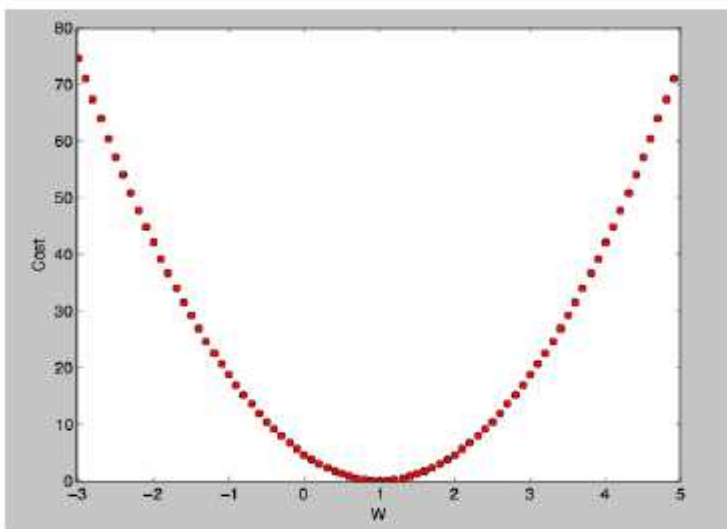
simplified hypothesis

b를 없애고 $H(x) = W(x)$

cost 함수에서도 b가 다 사라짐

w에 값을 넣어서 cost를 계산해보면 볼록한 모양으로 보임

$$cost(W) = \frac{1}{m} \sum_{i=1}^m (Wx^{(i)} - y^{(i)})^2$$



cost가 제일 낮은 점을 기계적으로 어떻게 찾냐가 관건!

gradient decscent algorithm

기울기 하강 알고리즘 설명

- cost function을 최소화함
- 많은 값들이 있는 경우에도 알고리즘 적용 가능하다.
- 아무점에서 시작할 수 있음
- W를 조금 바꾼다음에 cost를 변경한다
- 어떤 점에서 시작해도 최저점에 도달할 수 있다.
- 경사도는 어떻게 구하냐? 미분!

Formal definition

$$W := W - \alpha \frac{\partial}{\partial W} \frac{1}{2m} \sum_{i=1}^m (W x^{(i)} - y^{(i)})^2$$

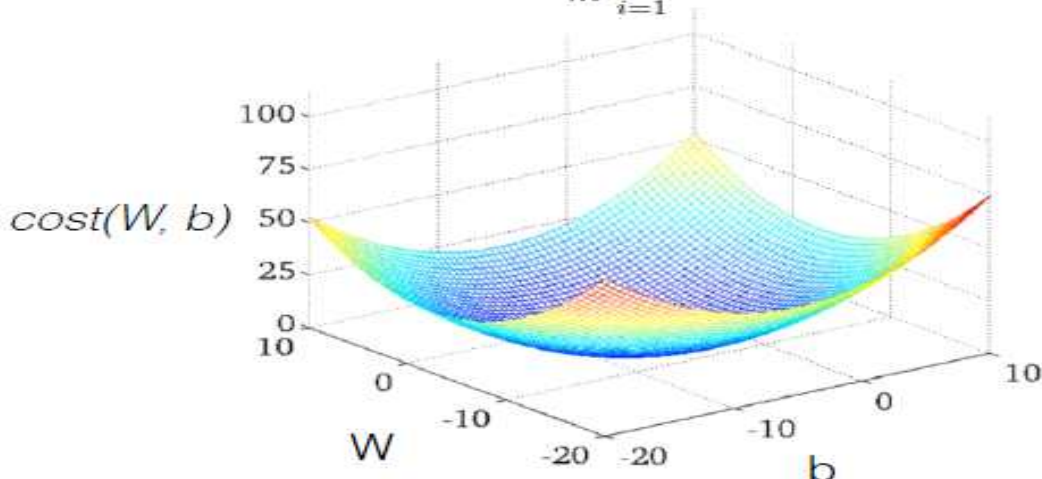
$$W := W - \alpha \frac{1}{2m} \sum_{i=1}^m 2(W x^{(i)} - y^{(i)}) x^{(i)}$$

$$W := W - \alpha \frac{1}{m} \sum_{i=1}^m (W x^{(i)} - y^{(i)}) x^{(i)}$$

w, b, cost(W, b)가 세축으로 한 convex function
cost function이 저런 모양인지 확인하면

Convex function

$$cost(W, b) = \frac{1}{m} \sum_{i=1}^m (H(x^{(i)}) - y^{(i)})^2$$



바로 gradient decscent algorithm을 쓰면 된다.

lecture 4 - multivariable linear regression

여태까지 x , y 가 각각 1개인 경우만 배웠다.

그러나 다른경우에는 여러가지 x 가 생긴다.

퀴즈 1, 퀴즈 2, 중간 -> 기말

hypothesis는 넓게만 해주면 됨

$$h(x_1, x_2, x_3) = w_1x_1 + w_2x_2 + w_3x_3 + b$$

cost는 똑같이 해주면 된다.

Hypothesis using matrix

$$\begin{pmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \\ x_{31} & x_{32} & x_{33} \\ x_{41} & x_{42} & x_{43} \\ x_{51} & x_{52} & x_{53} \end{pmatrix} \cdot \begin{pmatrix} w_1 \\ w_2 \\ w_3 \end{pmatrix} = \begin{pmatrix} x_{11}w_1 + x_{12}w_2 + x_{13}w_3 \\ x_{21}w_1 + x_{22}w_2 + x_{23}w_3 \\ x_{31}w_1 + x_{32}w_2 + x_{33}w_3 \\ x_{41}w_1 + x_{42}w_2 + x_{43}w_3 \\ x_{51}w_1 + x_{52}w_2 + x_{53}w_3 \end{pmatrix}$$

$[n, 3]$

$[3, 1]$

$[n, 1]$

$$H(X) = XW$$

매트릭스를 쓸때는 보통 x 를 앞에둬, 그리고 대문자를 씀

매트릭스 곱 일일이 해주기 귀찮음

인스턴스의 수 만큼 row개수를 주면 된다.

매트릭스의 장점이다!

매트릭스로 multivariable, n 개의 instances 쉽게 처리 가능

lecture 5 - Logistic(regression) classification

classification

처음에는 binary배울거임

예를 들어, spam detection, facebook feed

credit card fraudulent transaction detection: legitimate/fraud

y가 yes/no true/false등으로 되는 것임(0, 1로 encoding)

linear regression을 사용하여 일정값 이하면 fail, 이상이면 pass

그런데 공부시간이 엄청 큰 x가 들어오면 기울기 값이 크게 변한다.

또, y가 0 또는 1인데,

hypothesis는 0과 1사이의 값이 아닌 정답을 줄 수 있다.

이러한 문제 때문에!

logistic hypothesis를 0~1사이로 만들어 줘야하는 것이 필요하다!

sigmoid - s자함수인 $g(z)$ 를 사용해서 0 ~ 1 사이로 만들어주자!

$g(z)$ 는 아무리 커도 1보다 작고 아무리 작아도 0보다 크다.

$$H(X) = \frac{1}{1 + e^{-(W^T X)}}$$

$z = WX$, $H(x) = g(z)$ 이렇게 두면 된다.

lecture 5-2 - Logistic regression cost함수 설명

cost function

가설이 조금 바뀔 0 ~ 1사이로

기존에는 매끈한 밥그릇 모양인데

바뀐 hypothesis는 꾸불꾸불한 밥그릇

각 최소점은 local minimum 전체의 최저점은 global minimum

우리는 global minium을 찾아야하는데

gradient descent를 쓸 수 없음

새로운 cost function이 필요

New cost function for logistic

$$\text{cost}(W) = \frac{1}{m} \sum c(H(x), y)$$

$$c(H(x), y) = \begin{cases} -\log(H(x)) & : y = 1 \\ -\log(1 - H(x)) & : y = 0 \end{cases}$$

두 가지 경우를 합치면, 위에서 본 매끈한 그릇 그래프가 된다.

$$c(H(x), y) = y \log(H(x)) - (1 - y) \log(1 - H(x))$$

cost minimize는 똑같이 해주면 된다.

Minimize cost - Gradient decent algorithm

$$\text{cost}(W) = -\frac{1}{m} \sum y \log(H(x)) + (1 - y) \log(1 - H(x))$$



$$W := W - \alpha \frac{\partial}{\partial W} \text{cost}(W)$$



공부한 흔적들

강의필기

이름	수정된 날짜	유형	크기
Lab02-Linear Regression.pdf	2018-10-07 오전 8:09	Adobe Acrobat Doc...	548KB
lab-03.ipynb	2018-10-07 오후 11:	IPYNB 파일	28KB
Lab03-Minimizing Cost.pdf	2018-10-07 오전 9:33	Adobe Acrobat Doc...	425KB
lab-04.ipynb	2018-10-08 오전 12:	IPYNB 파일	52KB
lab-04-2.ipynb	2018-10-08 오전 12:	IPYNB 파일	27KB
Lab04-Multi-variable linear regression.pdf	2018-10-07 오후 11:	Adobe Acrobat Doc...	796KB
lab-05.ipynb	2018-10-08 오후 5:01	IPYNB 파일	3,203KB
Lab05-Logistic (regression) classifier.pdf	2018-10-08 오전 12:	Adobe Acrobat Doc...	433KB
lec0.pdf	2018-10-06 오후 10:	Adobe Acrobat Doc...	2,698KB
lec1.pdf	2018-10-06 오후 10:	Adobe Acrobat Doc...	1,393KB
lec2.pdf	2018-10-07 오전 8:09	Adobe Acrobat Doc...	782KB
lec3.pdf	2018-10-07 오전 9:31	Adobe Acrobat Doc...	1,759KB
lec4.pdf	2018-10-07 오전 9:33	Adobe Acrobat Doc...	386KB
lec5.pdf	2018-10-08 오전 12:	Adobe Acrobat Doc...	3,478KB
lecture0.txt	2018-10-06 오후 10:	텍스트 문서	1KB
lecture1.txt	2018-10-06 오후 10:	텍스트 문서	2KB
lecture2.txt	2018-10-07 오후 1:26	텍스트 문서	1KB
lecture3.txt	2018-10-07 오후 2:41	텍스트 문서	2KB
lecture4.txt	2018-10-07 오후 11:	텍스트 문서	2KB
lecture5.txt	2018-10-08 오후 4:18	텍스트 문서	1KB
lecture5-2.txt	2018-10-08 오후 4:35	텍스트 문서	1KB

```
lecture1.txt
1 머신러닝의 용어와 개념 설명
2
3 1
4 수업할 것 설명
5
6 2
7 머신러닝 설명
8 explicit programming - 정확하게!는 제약이 있다.
9 스팸 필터같은 경우는 규칙들이 너무 많다.
10 자율주행도 너무 많음!
11
12 컴퓨터가 학습해서 배우는 능력을 가지는 것을 머신러닝이라고 함
13
14 3
15 supervised/unsupervised learning
16 supervised는 정해진 데이터다 - training set
17
18 4
19 supervised 예를 들면, 개 고양이, 컵 사진 구분
20 정답을 알려주고 학습을 시켜서 supervised
21
22 5
23 unsupervised learning
24 정답이 없이 스스로 학습!
25 예로는 구글 뉴스 그룹화, 단어 그룹화
26
27 6
28 supervised learning
29 - image labeling
30 - email spam filter
```

실습

 jupyter lab-05 Last Checkpoint: 7분 전 (autosaved)  Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

In [1]:

```
import tensorflow as tf
import numpy as np
```

In [2]:

```
x_data = [[1, 2], [2, 3], [3, 1], [4, 3], [5, 3], [6, 2]]
# 0 fail 1 pass
y_data = [[0], [0], [0], [1], [1], [1]]

#placeholder shape에서 주의
x = tf.placeholder(tf.float32, shape = [None, 2])
y = tf.placeholder(tf.float32, shape = [None, 1])
```

In [3]:

```
# 애도 shape가 중요
w = tf.Variable(tf.random_normal([2, 1]), name = "weight")
b = tf.Variable(tf.random_normal([1]), name = "bias")
```

In [4]:

```
#sigmoid가 알아서 해줌
hypothesis = tf.sigmoid(tf.matmul(x, w) + b)
```

In [5]:

```
cost = -tf.reduce_mean(y * tf.log(hypothesis) + (1 - y) * tf.log(1 - hypothesis))
```

In [6]:

```
train = tf.train.GradientDescentOptimizer(learning_rate=0.01).minimize(cost)
```