Contents

1 Supervised Machine Learning: Regression and Classification

1.1 Week 1 linear Regression with One Variable

1.1.1 Learning Objectives

- Define machine learning
- Define supervised learning
- Define unsupervised learning
- Write and run Python code in Jupyter Notebooks
- Define a regression model
- Implement and visualize a cost function
- Implement gradient descent
- Optimize a regression model using gradient descent

Some notations

- In the training set, there are features (x) and targets (y)
- x is the input variable (feature)
- y is the output variable (target)
- m = number of training examples
- (x, y) = single training example
- $(x^{(i)}, y^{(i)}) = i^{th}$ training example $(1^{st}, 2^{nd}, 3^{rd}...)$
- $\hat{y}$ means prediction(estimated y)
- f is the function (model)(hypothesis)
- $f_{w,b}(x) = wx + b$
- Parameters: w, b
- Squared error cost function: $J(w,b) = \frac{1}{2m}\sum_{i=1}^{m}(\hat{y}^{(i)} - y^{(i)})^2$
- Find w, b : $\hat{y}^{(i)}$ is close to $y^{(i)}$ for all $(x^{(i)}, y^{(i)})$

$$J(w,b) = \frac{1}{2m}\sum_{i=1}^{m}(f_{w,b}(x^{(i)}) - y^{(i)})^2$$

- Goal: $minimize J(w,b)$

1.1.2 Gradient Descent

repeat until convergence{

$$w = w - \alpha \frac{\partial}{\partial w} J(w, b)$$

$$b = b - \alpha \frac{\partial}{\partial b} J(w, b)$$

}
Simultaneously update w and b
α is the learning rate(usually between 0 and 1, maybe 0.01)
$\frac{\partial}{\partial w} J(w, b)$ and $\frac{\partial}{\partial b} J(w, b)$ are derivative terms
Near a local minimum,

- Derivative becomes smaller
- Update steps become smaller

1.1.3 Gradient descent for linear regression

$$\frac{\partial}{\partial w} J(w, b) = \frac{1}{m} \sum_{i=1}^{m} (f_{w,b}(x)^{(i)} - y^{(i)}) x^{(i)}$$

$$\frac{\partial}{\partial b} J(w, b) = \frac{1}{m} \sum_{i=1}^{m} (f_{w,b}(x)^{(i)} - y^{(i)})$$

repeat until convergence{

$$w = w - \alpha \frac{1}{m} \sum_{i=1}^{m} (f_{w,b}(x)^{(i)} - y^{(i)}) x^{(i)}$$

$$b = b - \alpha \frac{1}{m} \sum_{i=1}^{m} (f_{w,b}(x)^{(i)} - y^{(i)})$$

}

1.1.4 Lab notes

Deep copy is a process in which the copying process occurs recursively. It means first constructing a new collection object and then recursively populating it with copies of the child objects found in the original. In case of deep copy, a copy of object is copied in other object. It means that any changes made to a copy of object do not reflect in the original object. In python, this is implemented using "deepcopy()" function.
0.3e : example: 1.949e+02 (Three decimal places are reserved before e)
8.4f : example: 199.9929 (Including the decimal point, there are eight decimal places, and four decimal places are reserved after the decimal point)

1.2 Week 2 Regression with multiple input variables

1.2.1 Learning Objectives

- Use vectorization to implement multiple linear regression
- Use feature scaling, feature engineering, and polynomial regression to improve model training
- Implement linear regression in code

1.2.2 Multiple Linear Regression

$2^2 + 2^2 = 8$ and $2 \times 2 = 4$

$$\cos^2\theta + \sin^2\theta = 1$$

The union of two sets A and B is denoted as $A \cup B = \{x \in A \text{ or } x \in B\}$
We are learning fractions $\frac{a}{\frac{b}{c}} \times \frac{\frac{d}{e}}{f} \geq 1$ which is good.

$$\frac{a}{\frac{b}{c}} \times \frac{\frac{d}{e}}{f} \geq 1$$

$$\left\{\left(\frac{a}{b}\right) + \left(\frac{c}{d}\right)\right\}$$

$$\sum_{i=a}^{b} g(i) = 0, \text{for } b < a$$

$$\sum_{i=1}^{n} i = \frac{n(n+1)}{2}$$

2 More in Mathematics

$\int_2^4$

$lim_{2\to 4}$

$$\int_0^\infty f(x)dx$$

$$\lim_{x\to c} f(x) = L$$

$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} + \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix} = \begin{bmatrix} 6 & 8 \\ 10 & 12 \end{bmatrix}$

3 Equations

$$3x + 5y = 2 \tag{1}$$

$$5x + 8y = 3 \tag{2}$$

$$x^2 - y^2 = (x+y)(x-y) \tag{3}$$

$$
\begin{aligned}
3x - 6 &= 9 \\
3x &= 9 + 6 \\
x &= \frac{9+6}{3} \\
x &= 5
\end{aligned}
\tag{4}
$$