WEEK 1

machine learning defination:

Field of study that gives Computers the ability to learn without being explicitly programmed.

—Arthur Samuel

A computer is said to learn from experience E with respect to some class of tasks T and performance P, if its performance at tasks in T, as measured by P, improves with experience E.

—Tom Mitchell

Intro to supervised learning

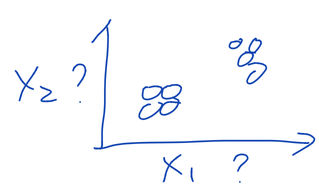
"right answers" given

Categorized as:

- Regression: Predict continuous valued output
- Classification: Discrete valued output

Intro to unsupervised learning

Approach problems with little or no ideas about what the results should look like



Supervised learning model

Training set

linear regression with one variable:

$$h_\theta(x) = \theta_0 + \theta_1 x$$

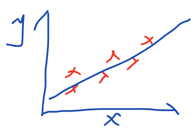
$$\downarrow$$

Learning Algorithm

$$\downarrow$$

$$\text{input} \longrightarrow \underbrace{h}_{\text{hypothesis}} \longrightarrow \text{output}$$

Cost function



θ_0, θ_1 to minimize: $\frac{1}{2m}\sum_{i=1}^{m}\left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$

m: # training examples

$h_\theta(x^{(i)}) = \theta_0 + \theta_1 x^{(i)}$

y: actual value

Cost function: $J(\theta_0, \theta_1) = \frac{1}{2m}\sum_{i=1}^{m}\left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$

(Squared error function)

Commonly used for regression problems

Intuition:

hypothesis: $h_\theta(x) = \theta_0 + \theta_1 x$

Parameters: θ_0, θ_1

Cost Function: $J(\theta_0, \theta_1) = \frac{1}{2m}\sum_{i=1}^{m}\left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$

Goal: $\underset{\theta_0, \theta_1}{\text{minimize}}\ J(\theta_0, \theta_1)$

Linear regression with one variable —— Gradient descent

$J(\theta_0, \theta_1) \longrightarrow$ want $\min_{\theta_0, \theta_1} J(\theta_0, \theta_1)$

Outline:

start with some θ_0, θ_1

keep changing θ_0, θ_1 to reduce $J(\theta_0, \theta_1)$ until hopefully end up with a minimium

repeat until convergence: {

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1) \quad \text{for } j=0 \text{ and } j=1 \quad \}$$

α: learning rate

Simultanesly update θ_1, θ_0

Gradient decent for linear regression

$$\frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1) = \frac{\partial}{\partial \theta_j} \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2 = \frac{\partial}{\partial \theta_j} \frac{1}{2m} \sum_{i=1}^{m} (\theta_0 + \theta_1 x^{(i)} - y^{(i)})^2$$

$$\begin{cases} \frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1) = \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) \\ \frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1) = \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) \cdot x^{(i)} \end{cases}$$

$$\begin{cases} \theta_0 := \theta_0 - \alpha \frac{1}{m} \sum_{j=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) \\ \theta_1 := \theta_1 - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) \cdot x^{(i)} \end{cases}$$

update $\theta_0\ \theta_1$ simultaneonsly

Cost function for linear regression will always be a convex function (Bowl-shaped), 1 global/local min

"Batch" Gradient Descent

"Batch": Each step of gradient descent uses all ~~the~~ the training examples.

Linear Algebra Review

skip, I know everything