Week 01

- Introduction
    - Wtf?
    - Supervised/ Unsupervised
- Linear regression with one variable
    - Model and Cost Function
        - Model representation
        - Cost function
    - Parameter Learning
        - Gradient descent
        - Gradient descent for linear regression
- Linear algebra

I. Intro

1. Defintions:

- Arthur Samuel, 59' => ML gives computer ability to learn without being explicitly programmed;
- Tom Mitchell, 98': a computer is said to learn from experience (E) with respect to task (T) and some performances (P), if its performances on T as measured on by (P) improves (E).
- Example:
    - Email program watches you mark or no as Spam, and learns how to better filter Spam.
        - T = Classify emails
        - E = Watching you label emails as spam or not spam
        - P = The number of emails correctly classified
    - Playing checkers.
        - T = playing
        - E = play many games
        - P = Probablilty that program will win the next game
- ML algo.: Supervised and unsupervised learning + (bonus: reinforcement & recommender)

2. Supervised ML

Right answers are given and task of algorithm is to produce more right answers.

There's a relationship between the input and the output.

- Example:
- **Beast cancer:**

  - It is called classification problem: label sizes to 1 for malignant or 0 for benign.
  - pTo Predict results in a discrete output.
  - → classification is about predicting a label.
- **Housing price prediction:**
  - It is called regression problem to predict continuous valued output (prices).
  - Price as a function of size is a continuous output.
  - Map input variables to some continuous function.
  - → regression is about predicting a quantity.
  - Can turn to into classification problem: output whether the house "sells for more or less than the asking price."
  - Here we are classifying the houses based on price into two discrete categories.
- **Age**
  - Given a picture, predict the age.

3. Unsupervised ML

- Approach problems with little or no idea what our results should look like.
- Can derive results structure by clustering it based on relationships among the variables in the input.
- No feedback based on the prediction results.
- Example
  - Cocktail party problem: identify human voices and music => Non-clustering.
  - Discover market segments and group customers into different market segments.
  - Group articles into sets about the same stories.
  - Create groups from a collection of 1,000,000 different genes based on lifespan, location, roles, etc.

II. Model and cost function

1. Model representation:

- $x^{(i)}$ = “input” variables = input features.
- $y^{(i)}$ = “output” variables = target variable. => example: predict price (y) from living area (x)
- couple $(x^{(i)}, y^{(i)})$ is called training example.
- couples $(x^{(i)}, y^{(i)}); i = 1..m$ is called training set.
- X space of input values; Y space of output values; $X = Y = \mathbb{R}$
- Supervised learning problem:
  - given a training set, to learn $h : X \rightarrow Y$ so h(x) ≈ y
  - h is called **hypothesis**

[[An algorithm will learn from a training set how to predict y when x is provided using h]]

Such as in our housing example, we call the learning problem a regression problem.

- Goal is to found h and its parameters as $h(x) = ax + b$

2. Cost function

- Cost function = average difference of all results of all x's & y's
  - $J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=0..m} (h_\theta(x^{(i)}) - y^{(i)})^2;$
  - m is the number of training examples
- Hypothesis $h_\theta(x) = \theta_0 + \theta_1 x$
  - Parameters: θ_0, θ_1
  - Choose these parameters so $h_\theta(x)$ is close to y for the training couple (x, y)
- Goal: $minimize J(\theta_0, \theta_1)$

Intuition 1:

- Ideally, the line should pass through all the points of our training data set to minimize $J(\theta_0, \theta_1)$.
- $h_\theta(x)$ is plotted : a linear function passing through some points of the training set.
- Plot $h_\theta(x)$ and $J(\theta_0, \theta_1)$
- $\theta_1 = 0$ => $J(0) = 2.3$
- $\theta_1 = 0.5$ => $J(0.5) = 0.58$
- $\theta_1 = 1$ => $J(1) = 0$
- $\theta_1 = 1.5$ => $J(1.5) = 0.58$
- $\theta_1 = 2$ => $J(0) = 2.3$

  → Thus as a goal, try to minimize the cost function. In this case, $\theta_1 = 1$ is the global minimum.

III. Parameter learning

1. Gradient descent

- Have some function $J(\theta_0, \theta_1)$
- Want to $minimize J(\theta_0, \theta_1)$
- **Outline**
  - Start with some θ_0, θ_1.
  - Keep changing θ_0, θ_1 to reduce $J(\theta_0, \theta_1)$ to find up a minimum.
- Estimate the parameters in the hypothesis function → Gradient descent