# A. Introduction

### I. What is Machine Learning (ML)

Define ML:

- Field of study that gives computer the ability to learn without being exblicitly programmed. (Máy tính có khả năng học hỏi mà ko cần được lập trình một cách rõ ràng)
- Well-posed Learning Problem: A computer program is said to learn from experience E with respect to some task T and some performance measure P, if its performance on T, as measured by P, improves with experience E.

Machine learning algorithms:

- Supervised learning (Học máy có giám sát)*
- Unsupervised learning (Học máy không có giám sát)
- *Others:* Reinforcement learning, Recommender system

II. Supervised Learning

In Supervised Learning, we have data set and correct output, and have a relationship between input and output.

Categorize:

- Regression (Hồi quy): Predict (dự đoán) result within a continuos output, meaning try to map input variables to some continuous function.
- Classification (Phân loại): Predict results in a discrete output, meaning try to map input variables into discrete categories.

III. Unsupervised Learning

Unsupervised Learning allow appoarch (tiếp cận) problems with little or no results, can derive (nhận ra) structure from data and don't necessarily know the effect of variables by clustering (thu góp) the data based on relationships among the variables in the data.

With Unsupervised Learning, there is no feedback based on the prediction results.

# B. Linear Regression with One Variable

### I. Model and Cost Function

1. Model Representation

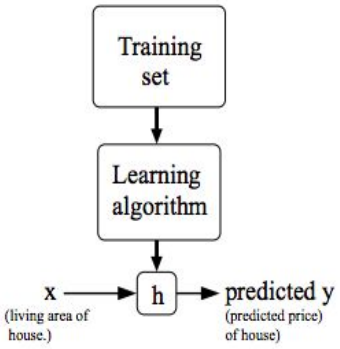
$x_i$ - "input" variables (input features)

$y_i$ - "output" or target variables

A pair $(x_i, y_i)$ is called a training example.

$(x_i, y_i)$; i = 1..m is call dataset.

The supervised learning problem, provide a training set, to learn a function h: X→Y so that h(x) is the good predictor for the corresponding (tương ứng) value of y.



- When target variable is continuos → the learning problem is regression problem.
- When terget variable can take on only small number of discrete (rời rạc) values → it is classification problem.

2. Cost function

Consider output y relates to input x follow linear function: $y = h_\theta(x) = \theta_0 + \theta_1 . x$

Idea: Choose $\theta_0$ and $\theta_1$ so that $h_\theta(x)$ is close to $y$ for training set.

Cost function (Squared error function or Mean squared error) takes an average difference of all results with inputs x and outputs y.

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} (\widehat{y}_i - y_i)^2 = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x_i) - y_i)^2$$

A contour plot (biểu đồ đường đồng mức) is the graph that contains many contour lines (đường đồng mức). A contour line of two variable function has a contain value at all point of the same line.

Any points in each contour line would to get same value of cost function.

**II. Parameter Learning**

1. Gradient Descent

We have hypothesis function and have a way of measuring how well it fits into data. Gradient descent was used to estimate parameters in the hypothesis function.

The gradient descent algorithms: repeat until convergence (hội tụ):

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1)$$

Where j = 0, 1.

At each iteration j, must update the parameters.

2. Gradient Descent Intuition

We should adjust α to ensure that the gradient descent algorithm converges in a reasonable time.

If α is too small, grandient descent can be slow.

If α is too large, gradient descent can overshoot the minimum, it may fail for converge, or even diverge (phân ra).

We don't need to decrease α over time.

3. Gradient Descent For Linear Regression

Apply gradient descent for linear regression, we have new function

repeat until convergence: {

$$\theta_0 := \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x_i) - y_i)$$

$$\theta_1 := \theta_1 - \alpha \frac{1}{m} \sum_{i=1}^{m} ((h_\theta(x_i) - y_i) x_i)$$

}

Where m is the training size, $\theta_0$, $\theta_1$ is the constant will be changed, $x_i$, $y_i$ are values of the given training set (data).

The point of all this is that if we start with a guess for our hypothesis and then repeatedly apply these gradient descent equations, our hypothesis will become more and more accurate.

This method looks at every example in the entire training set on every step, and is called "batch gradient descent".

**III. Linear Algebra Review**

1. Matrices and Vector

Matrices is the 2-dim arrays.

A vector is the matix is 1 column and many rows. Vector is the subset of matrices.

Notation and terms:

+ $A_{ij}$ refers to the element in the ith row and jth column of matrix A.
+ A vector with 'n' rows is referred to as an 'n'-dimensional vector.
+ $v_i$ refers to the element in the ith row of the vector.
+ In general, all our vectors and matrices will be 1-indexed. Note that for some programming languages, the arrays are 0-indexed.
+ Matrices are usually denoted by uppercase names while vectors are lowercase.
+ "Scalar" means that an object is a single value, not a vector or matrix.
+ **R** refers to the set of scalar real numbers.
+ $\mathbf{R}^n$ refers to the set of n-dimensional vectors of real numbers.

2. Addition and Scalar Multiplication

Addition and subtraction are "element-wise", so you simply add or subtract each corresponding element.

To add or subtract two matrices, their dimensions must be the same.

In scalar multiplication, we simply multiply every element by the scalar value. Similar with division

3. Matrix-Vector multiplication

We map the column of the vector onto each row of the matrix, multiplying each element and summing the result.

The result is a vector. The number of columns of the matrix must equal the number of rows of the vector.

An m x n matrix multiplied by an n x 1 vector results in an m x 1 vector.

4. Matrix-Matrix multiplication

We multiply two matrices by breaking it into several vector multiplications and concatenating the result.

An m x n matrix multiplied by an n x p matrix results in an m x p matrix.

To multiply two matrices, the number of columns of the first matrix must equal the number of rows of the second matrix.

5. Matrix multiplication properties

Matrices are not commutative: $A*B \neq B*A$

Matrices are associative (liên kết): $A*(B*C) = (A*B)*C$

The identity matrix ($I_n$), when multiplied by any matrix of the same dimensions, results in the original matrix. It's just like multiplying numbers by 1. The identity matrix simply has 1's on the diagonal (upper left to lower right diagonal) and 0's elsewhere.

6. Inverse and Transpose

The inverse of a matrix A is denoted $A^{-1}$. Multiplying by the inverse results in the identity matrix.

A non square matrix does not have an inverse matrix. Matrices that don't have an inverse are singular or degenerate.

The transposition of a matrix is like rotating the matrix 90° in clockwise direction and then reversing it.