SWINBURNE UNIVERSITY OF TECHNOLOGY

COS30082
Applied Machine Learning

Lecture 2

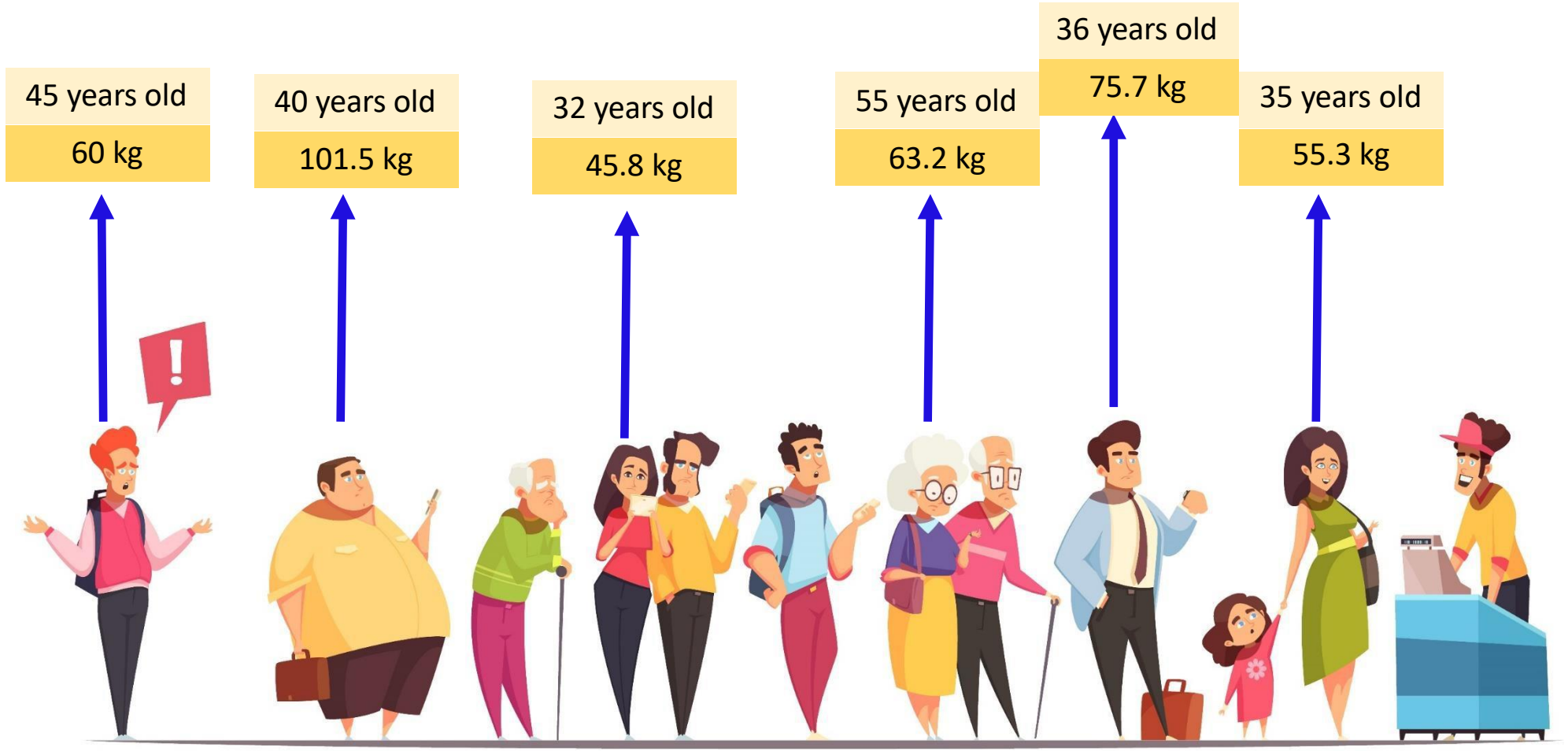
Linear Regression

Topics

- Understanding of regression problem
- To find out how linear regression works
- To learn linear regression optimization techniques
- To solve a machine learning problem using a linear regression algorithm

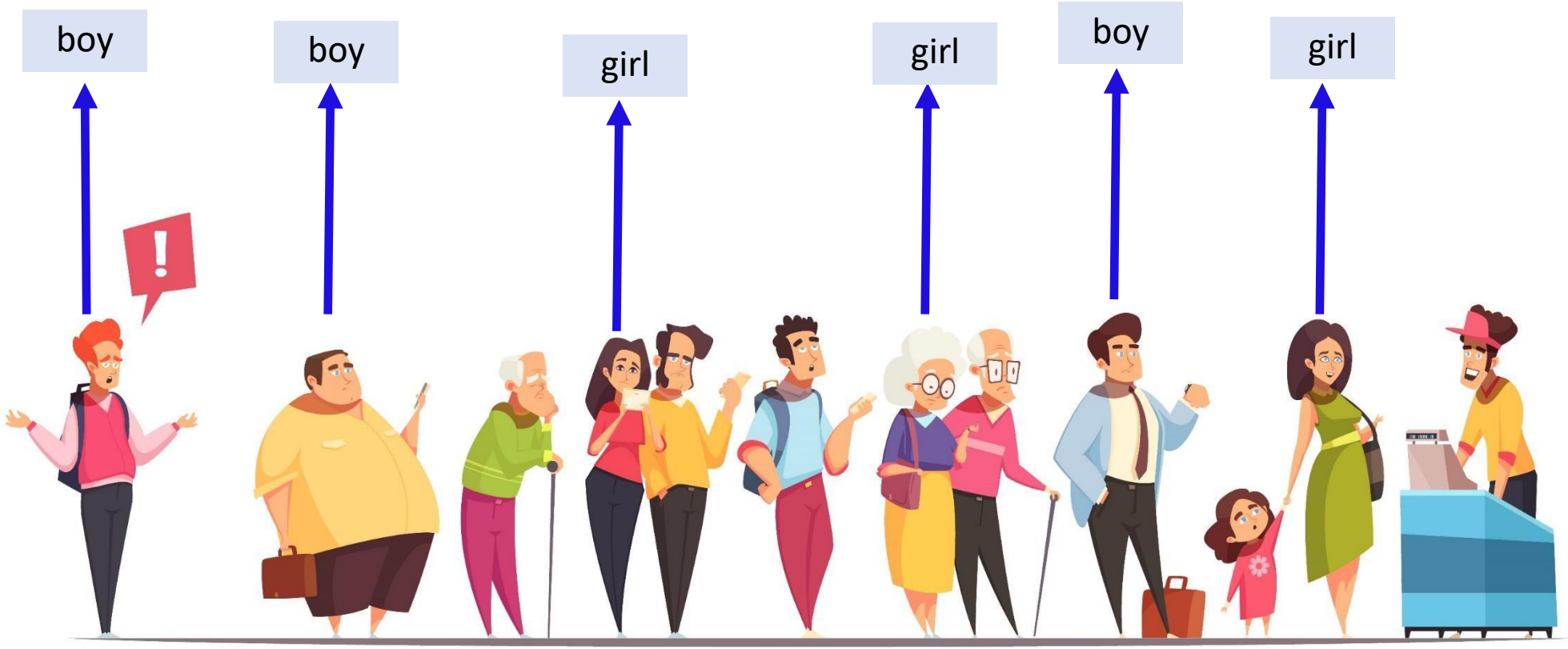
What is Regression problem?

- A regression problem is when the target output can be represented by a **real** or **continuous** value, such as “age” or “weight”.



What is Regression problem?

- Example of non-regression problem.



Introduction of Linear Regression

- Linear regression is an algorithm used in machine learning to model the relationship between two variables, x and y . The y denotes the **dependent variable** while the x denotes the **independent variables**.

- The model assumes a linear relationship between the dependent and independent variables .

- Linear regression is one of the supervised learning techniques.

Introduction of Linear Regression

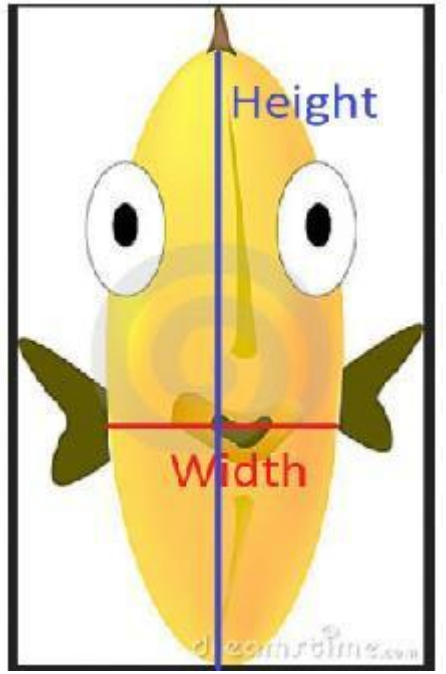
- **Simple linear regression** refers to a method when dealing with a **single** independent variable x.

- **Multiple linear regression** refers to a method when dealing with **multiple** independent variable x.

Introduction of Linear Regression

- For example, from a series of N training set, we want to model the relationship between the *height* and *width* of sea bream (a type of fish species).

Height	Width
11.52	4.02
12.48	4.3056
12.3778	4.6961
12.73	4.4555
12.444	5.134
⋮	⋮
⋮	⋮



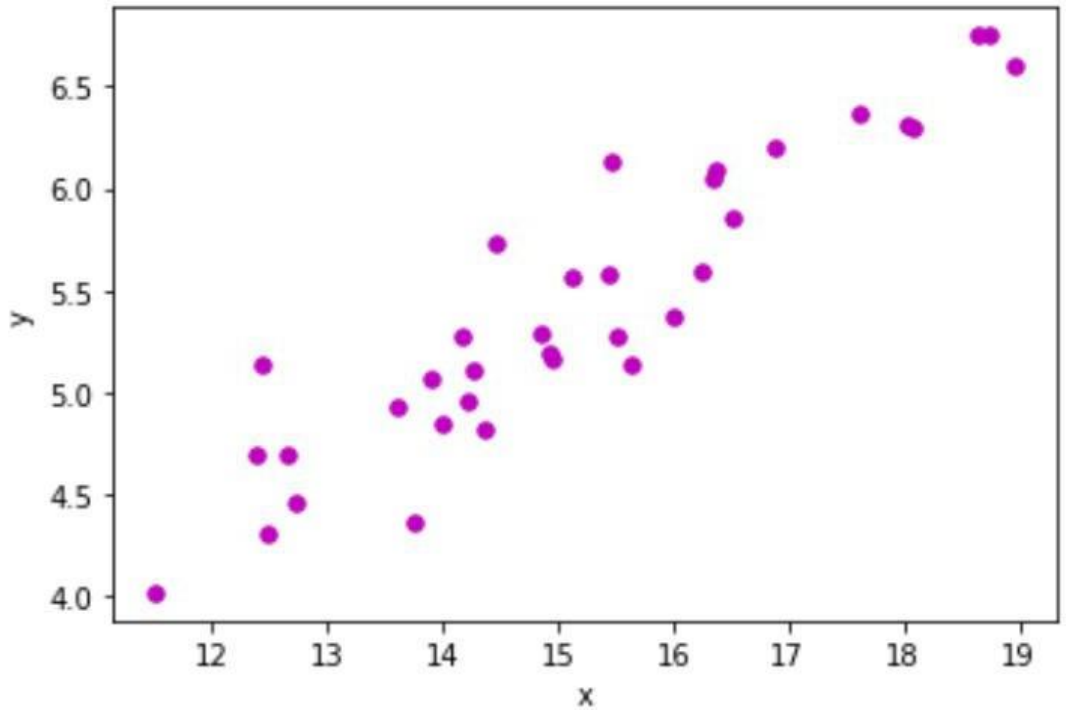
Source from [1]

[1] Aung Pyae, 2019, Fish Market database of common fish species for fish market, Kaggle, viewed 22 April 2020,<https://www.kaggle.com/aungpyaeap/fish-market>

Simple linear regression

- We assign *height* as the independent variable, x and *width as the* dependent variable, y .
- If we plot the data, we can see a positive relationship between the two variables.

Height, x	Width, y
11.52	4.02
12.48	4.3056
12.3778	4.6961
12.73	4.4555
12.444	5.134
⋮	⋮
⋮	⋮

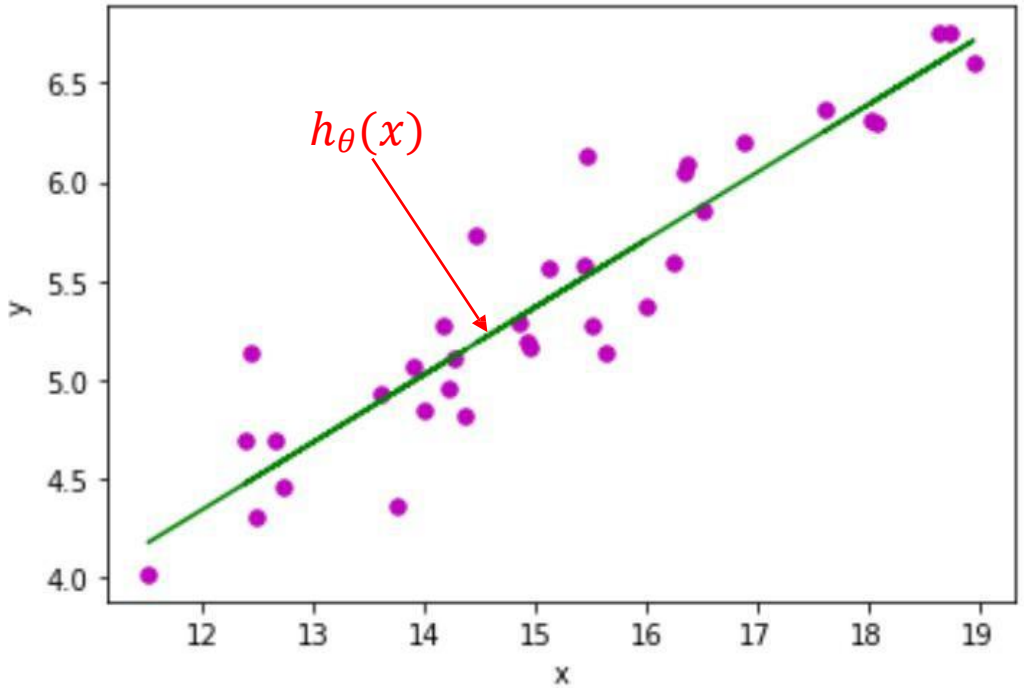


Simple linear regression

- The relationship between the two variables x and y can be modeled by a linear equation:

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

given θ_0 is the bias (intercept) , θ_1 is the weight (coeficient) associated to x



Simple linear regression

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

- Imagine if you fit a linear model and have the solutions for θ_0 and θ_1,

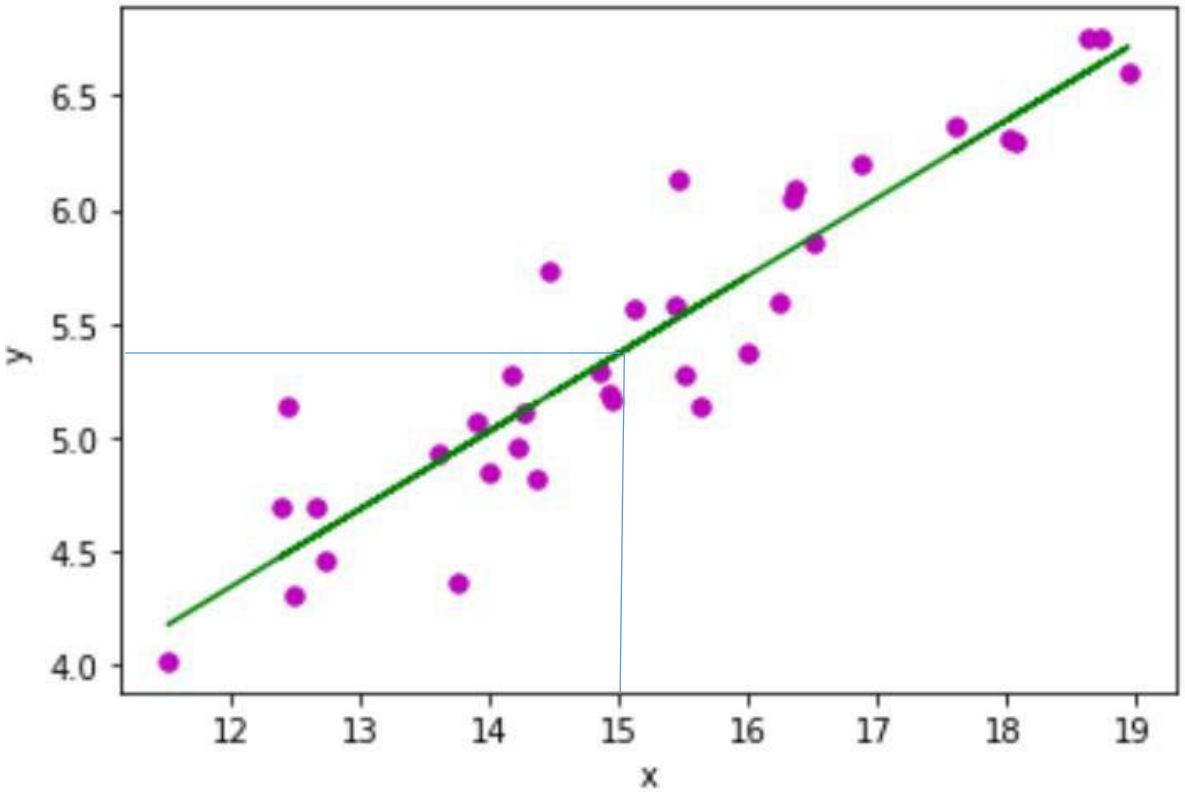
  θ_0 = 0.261

  θ_1 = 0.340

If x = 15,

$h_\theta(x) = 0.261 + 0.340\ (15)$

$= 5.361$



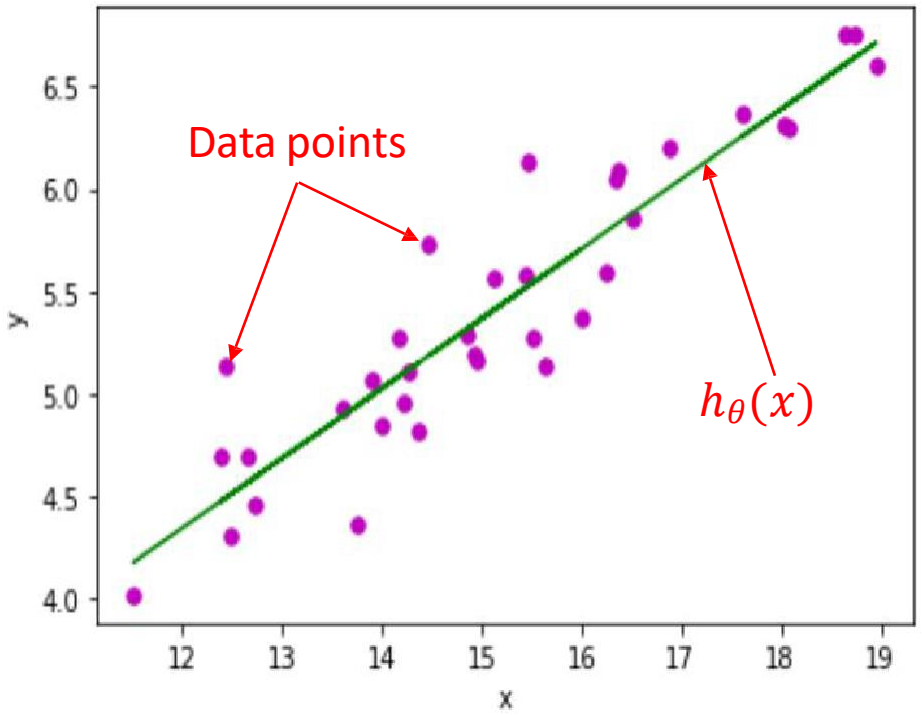
Simple linear regression

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

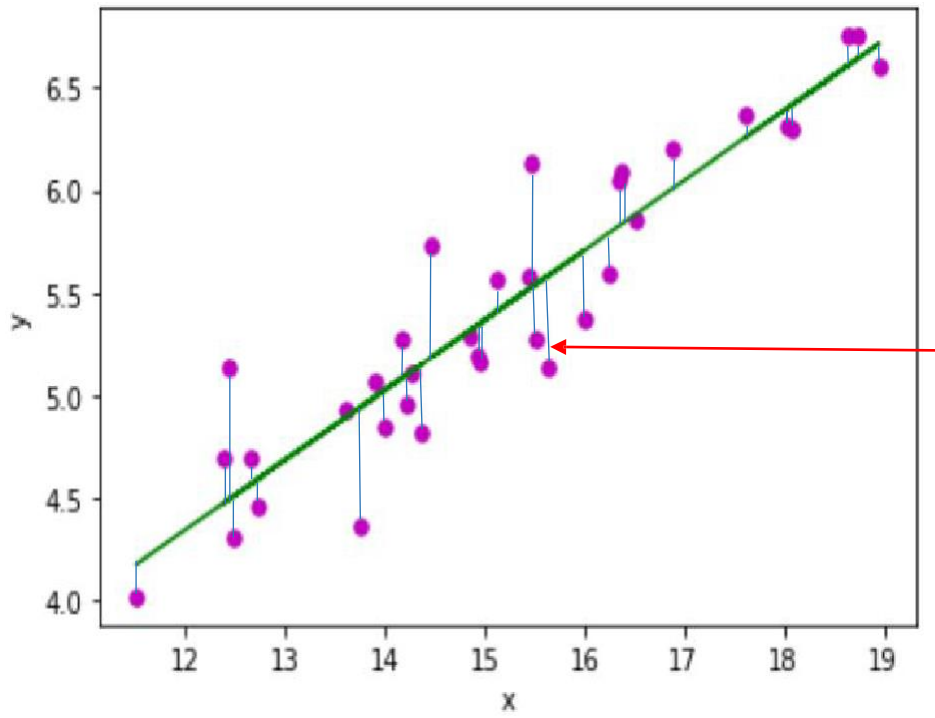
- Below is the optimization methods to find the parameters θ_0 and θ_1:
  - Ordinary Least Square (OLS)
  - Gradient Descent

Ordinary Least Square

- Given *N* number of data points (training set) = $(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}),\ldots\ldots, (x^{(N)}, y^{(N)})$
- Goal: To find a linear model ($h_\theta(x) = \theta_0 + \theta_1 x$) that best fit all the data.



Ordinary Least Square



$|y^{(n)} - h_\theta(x^{(n)})|$

- Cost function

$$J(\theta_0, \theta_1) = \Sigma_{n=1}^{N} \left(y^{(n)} - h_\theta(x^{(n)})\right)^2 = \Sigma_{n=1}^{N} \left(y^{(n)} - (\theta_0 + \theta_1 x^{(n)})\right)^2$$

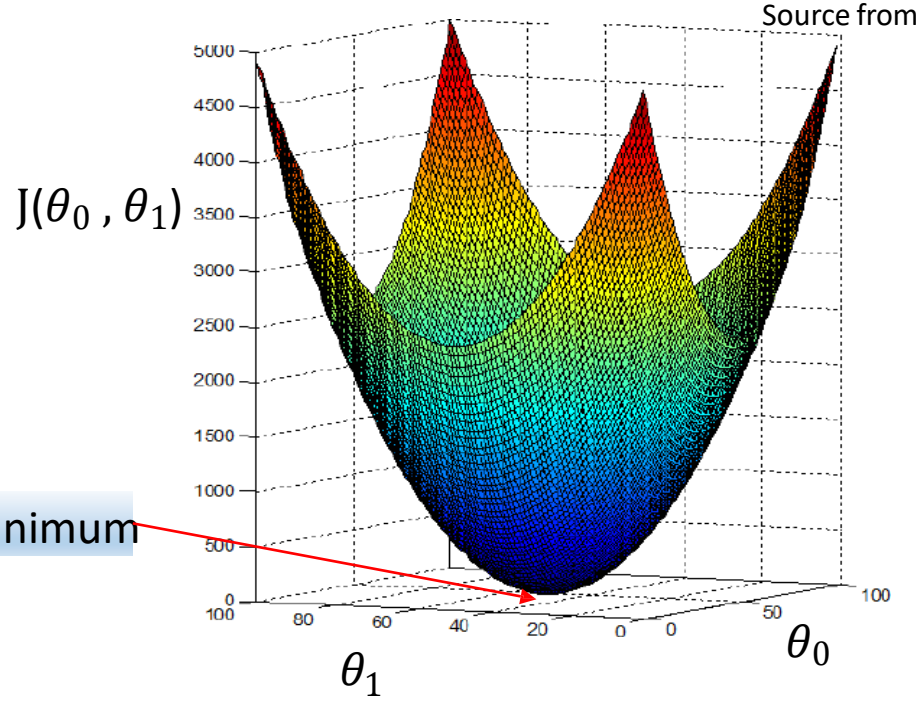
Called least square cost function

Ordinary Least Square

$$J(\theta_0 , \theta_1) = \Sigma_{n=1}^{N} \ (y^{(n)} - (\theta_0 + \theta_1 x^{(n)}))^2$$

- Characteristic of least square cost function:
  - Least square is one of the convex optimization problems
  - The convex function have a single global minimum

Source from [2]



[2] Optimization and Least Square, accessed 28 April 2020, <http://www.cs.cornell.edu/courses/cs1114/2009sp/lectures/CS1114-lec19.pdf>

Ordinary Least Square

- Method of least square directly computes the optimal choice of (θ_0 , θ_1) at the global minimum

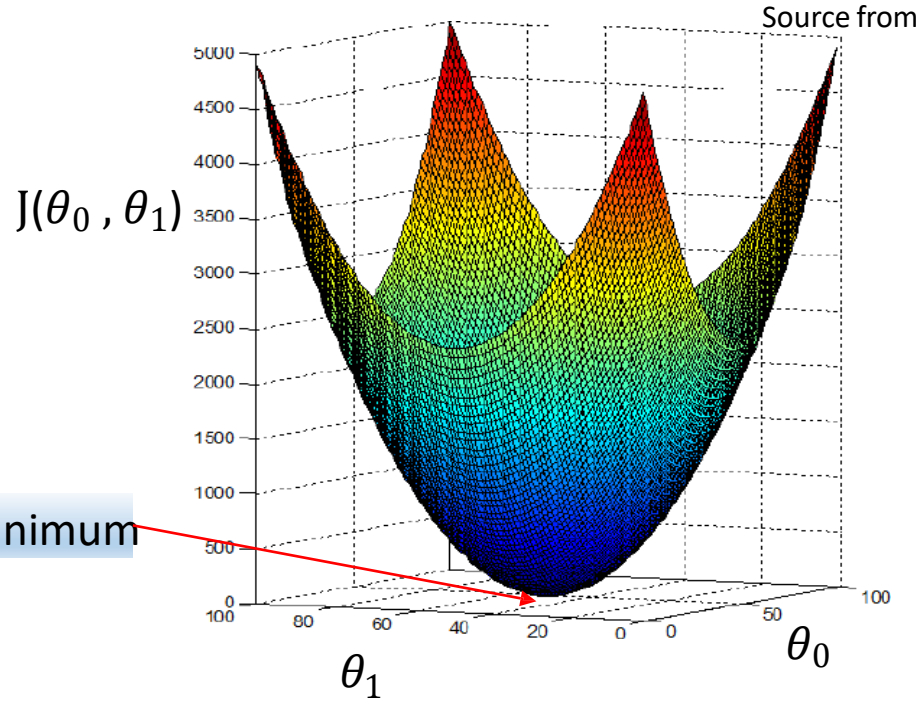
$$argmin_{\theta_0 , \theta_1} \ \mathrm{J}(\theta_0 , \theta_1)$$

setting:

$$\frac{\partial J}{\partial \theta_0} = 0$$

$$\frac{\partial J}{\partial \theta_1} = 0$$

Source from [2]



Ordinary Least Square

$$J(\theta_0, \theta_1) = \Sigma_{n=1}^{N} \left(y^{(n)} - h_\theta(x^{(n)})\right)^2 = \Sigma_{n=1}^{N} \left(y^{(n)} - (\theta_0 + \theta_1 x^{(n)})\right)^2$$

Differentiating wrt. θ_0 : $\frac{\partial J}{\partial \theta_0} = \sum_{n=1}^{N} 2(y^{(n)} - (\theta_0 + \theta_1 x^{(n)})) = 0$

Differentiating wrt. θ_1 : $\frac{\partial J}{\partial \theta_1} = \sum_{n=1}^{N} 2(y^{(n)} - (\theta_0 + \theta_1 x^{(n)})) . (-x^{(n)}) = 0$

$$\Rightarrow \quad \theta_0 = \frac{1}{N}\left[\Sigma_{n=1}^{N} y^{(n)} - \theta_1 \Sigma_{n=1}^{N} x^{(n)}\right]$$

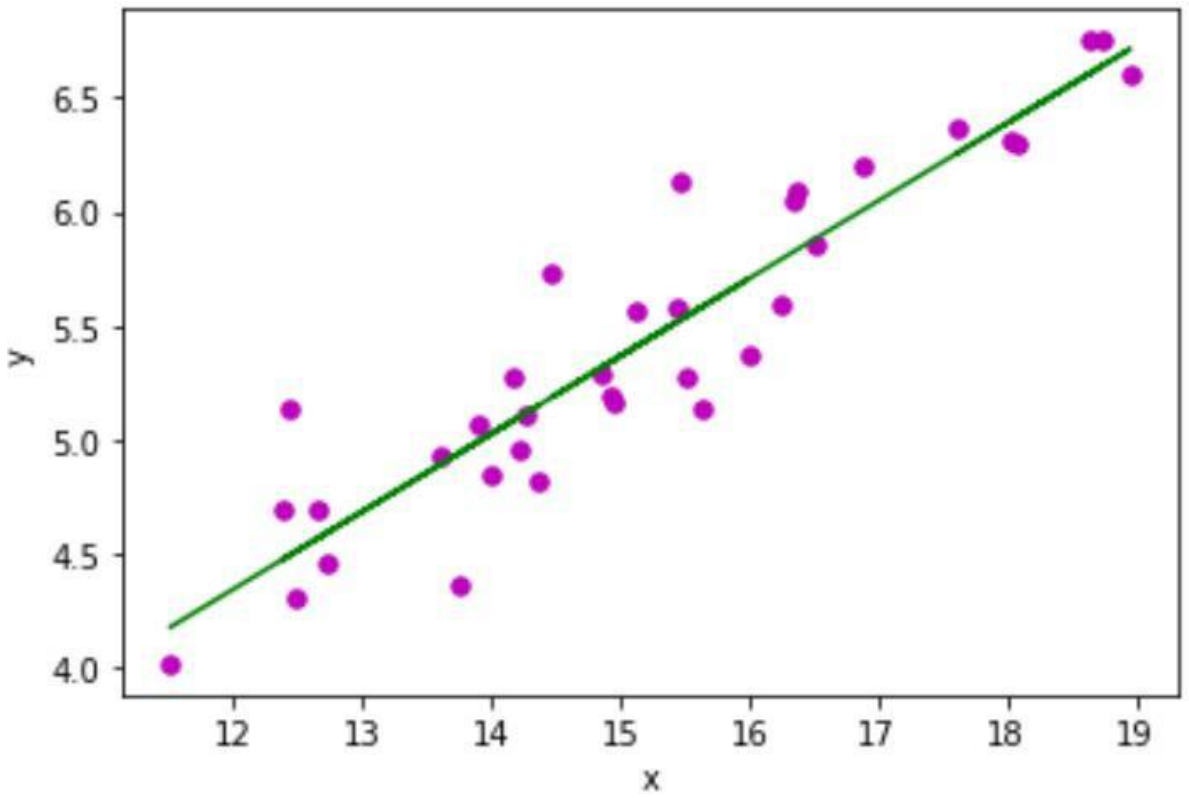
$$\Rightarrow \quad \theta_1 = \frac{N \sum_{n=1}^{N} x^{(n)} y^{(n)} - \left(\sum_{n=1}^{N} x^{(n)}\right)\left(\sum_{n=1}^{N} y^{(n)}\right)}{N \sum_{n=1}^{N} (x^{(n)})^2 - \left(\sum_{n=1}^{N} x^{(n)}\right)^2}$$

Ordinary Least Square

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

$$\theta_0 = \frac{1}{N}\left[\sum_{n=1}^{N} y^{(n)} - \theta_1 \sum_{n=1}^{N} x^{(n)}\right]$$

$$\theta_1 = \frac{N\sum_{n=1}^{N} x^{(n)} y^{(n)} - \sum_{n=1}^{N} x^{(n)} \sum_{n=1}^{N} y^{(n)}}{N\sum_{n=1}^{N}\left(x^{(n)}\right)^2 - \left(\sum_{n=1}^{N} x^{(n)}\right)^2}$$



Ordinary Least Square

- **Normal Equation** is an analytical approach to Linear Regression with a Least Square Cost Function.
- Normal equation provides a more convenient way to solve the linear regression optimization problem **without** going through all the long partial derivatives $\frac{\delta}{\delta}$.
- Normal equation is as follows:

$$\theta = (X^T X)^{-1} X^T y$$

Ordinary Least Square

- For example, to optimise two parameters θ_0 and θ_1 :

$$J(\theta_0 , \theta_1) = \sum_{n=1}^{N} (y^{(n)} - (\theta_0 + \theta_1 x^{(n)}))^2$$

Substitute X and Y into: $\theta = (X^T X)^{-1} X^T y$

where

$$X = \begin{bmatrix} x_0^{(1)} & x_1^{(1)} \\ x_0^{(2)} & x_1^{(2)} \\ x_0^{(3)} & x_1^{(3)} \\ \vdots & \vdots \end{bmatrix}_{(N \times 2)} \qquad y = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ y^{(3)} \\ \vdots \end{bmatrix}_{(N \times 1)}$$

Number of training data

Number of parameters

Simple linear regression

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

- Below is the optimization methods to find the parameters θ_0 and θ_1:
  - Ordinary Least Square (OLS)
  - Gradient Descent

Gradient Descent

Gradient descent is a way to minimize an objective function $\mathrm{J}(\theta)$ parameterized by a model's parameters $\theta \in \mathbb{R}^d$ by updating the parameters in the opposite direction of the gradient of the objective function with respect to the parameters.

Gradient Descent

- Hypothesis $h_\theta(x)$ is represented as:

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

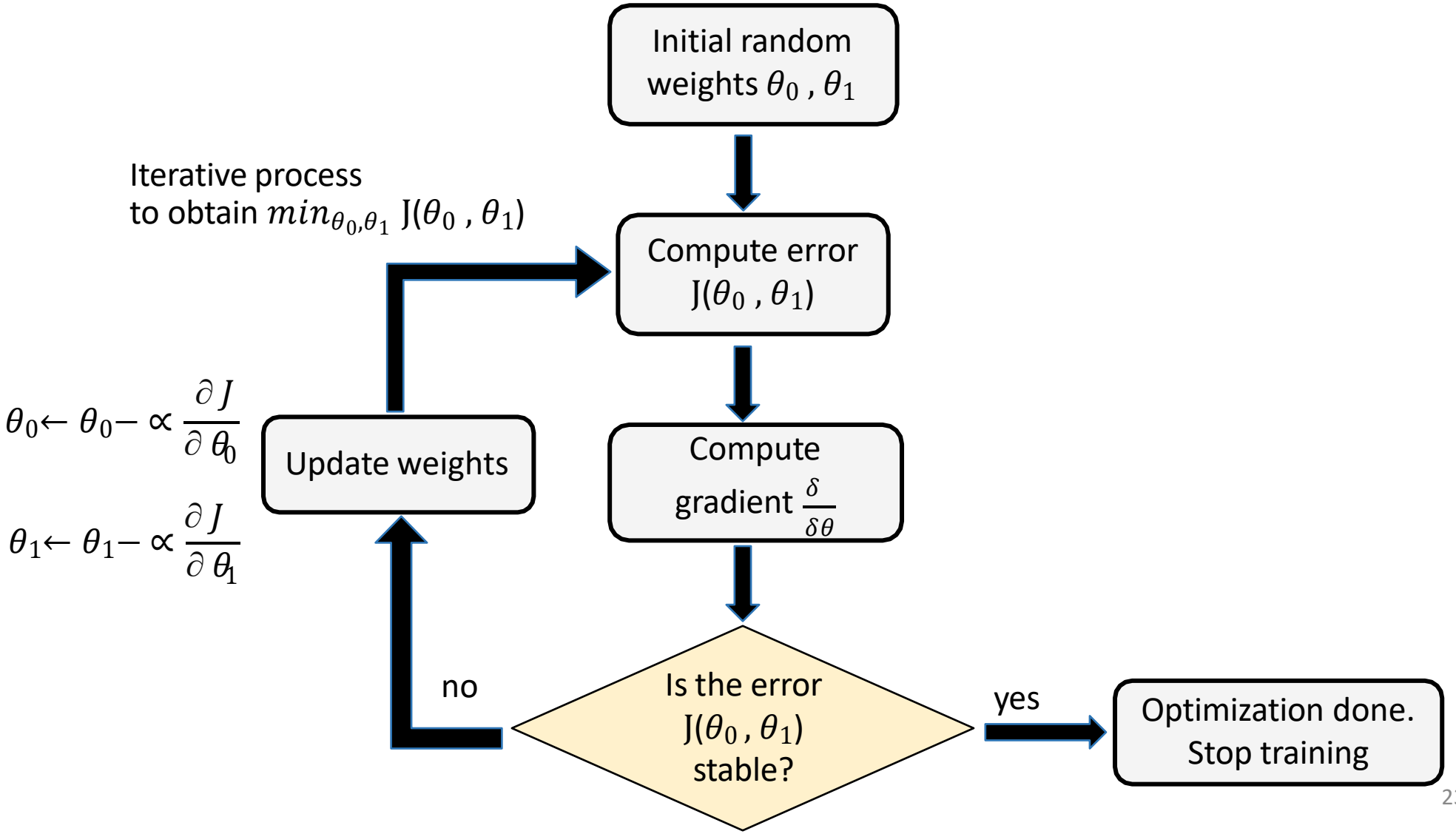
- Parameters: θ_0 and θ_1
- Cost function:

$$J(\theta_0, \theta_1) = \frac{1}{2N} \sum_{n=1}^{N} (h_\theta(x^{(n)}) - y^{(n)})^2$$

- Goal:

$$minimize \quad J(\theta_0, \theta_1)$$

How Gradient Descent works



How the Gradient Descent works

- Gradient descent update weights simultaneously:

Repeat

$$\theta_0 \leftarrow \theta_0 - \alpha \frac{\partial J}{\partial \theta_0}$$
$$\theta_1 \leftarrow \theta_1 - \alpha \frac{\partial J}{\partial \theta_1}$$

[1] $tmp_0 \leftarrow \theta_0 - \alpha \frac{\partial J}{\partial \theta_0}$

[2] $tmp_1 \leftarrow \theta_1 - \propto \frac{\partial J}{\partial \theta_1}$

[3] $\theta_0 \leftarrow tmp_0$

[4] $\theta_1 \leftarrow tmp_1$

Not in this way:

[1] $tmp_0 \leftarrow \theta_0 - \alpha \frac{\partial J}{\partial \theta_0}$

[2] $\theta_0 \leftarrow tmp_0$

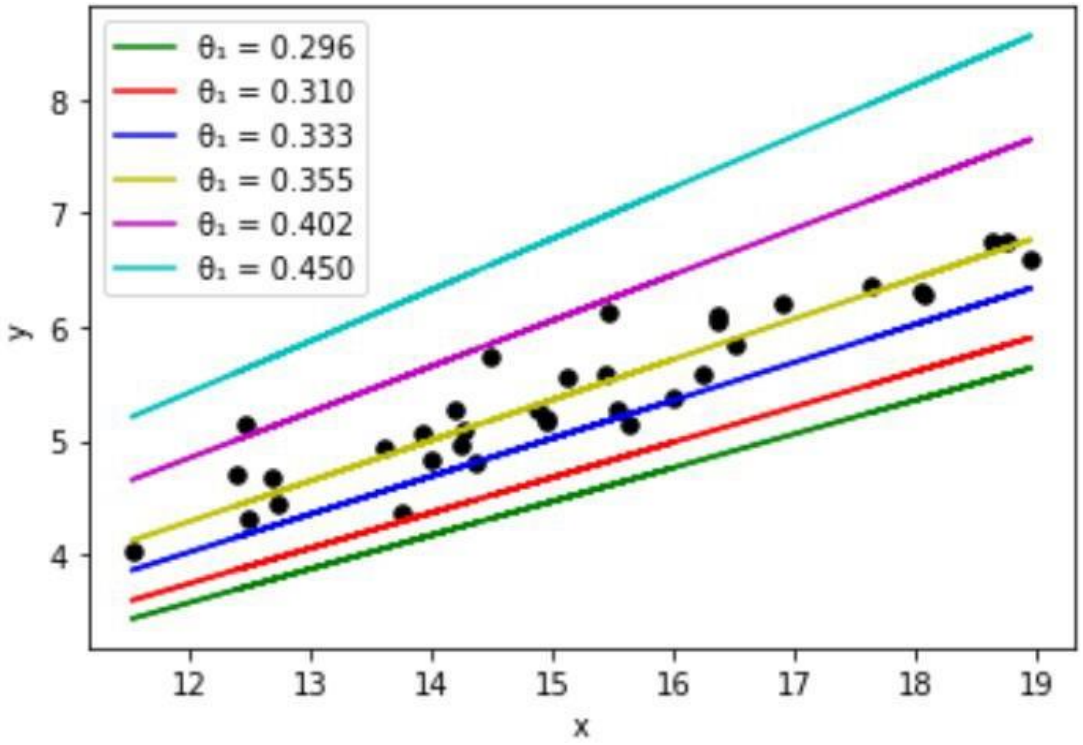
[3] $tmp_1 \leftarrow \theta_1 - \alpha \frac{\partial J}{\partial \theta_1}$

[4] $\theta_1 \leftarrow tmp_1$

How the Gradient Descent works

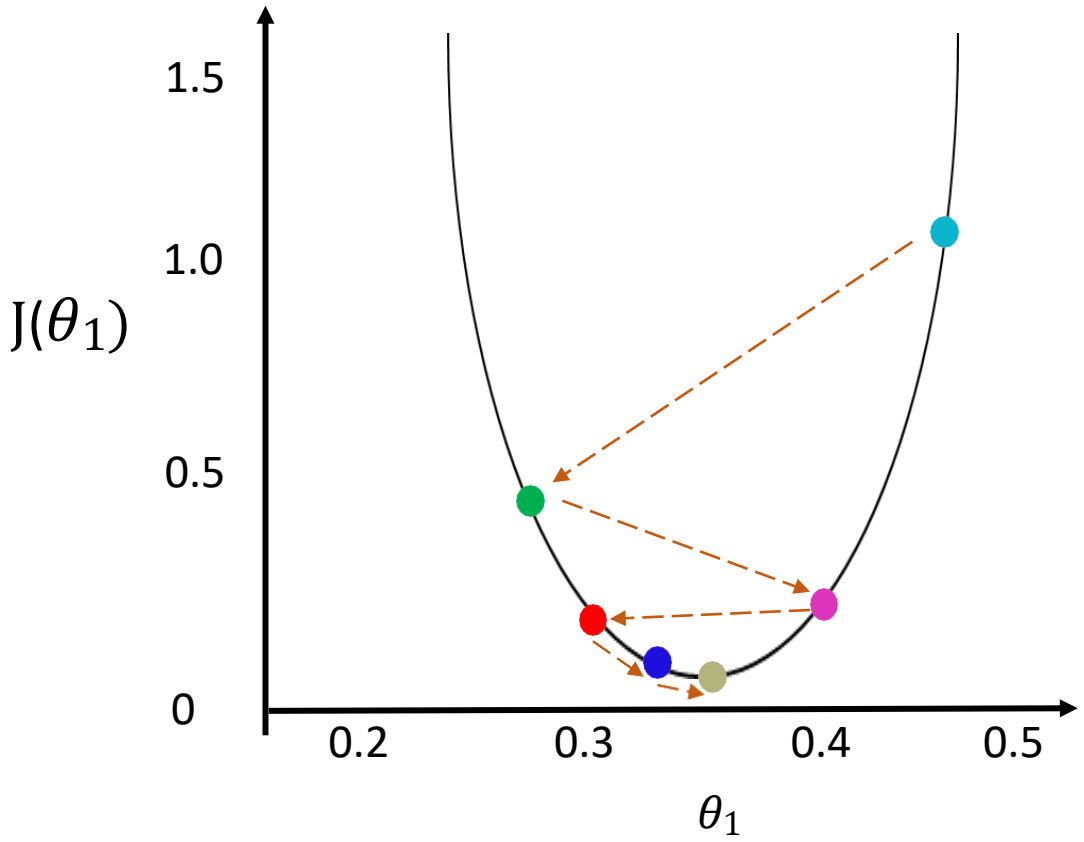
$h_\theta(x) = \theta_0 + \theta_1 x$

$h_\theta(x)$ for fixed θ_0

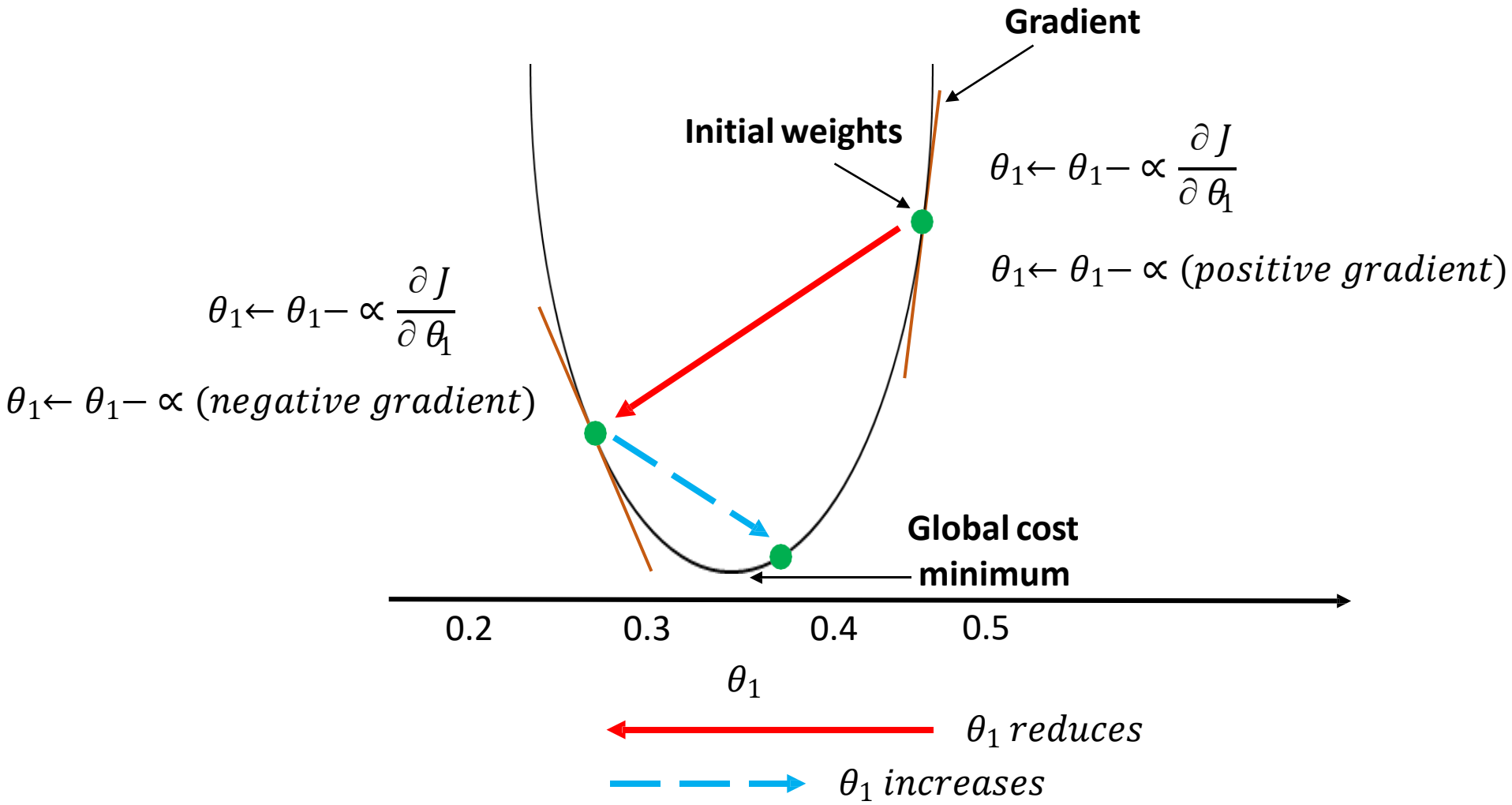


Iterative process to obtain $min_{\theta_1} J(\theta_1)$

$$\theta_1 \leftarrow \theta_1 - \propto \frac{\partial J}{\partial \theta_1}$$

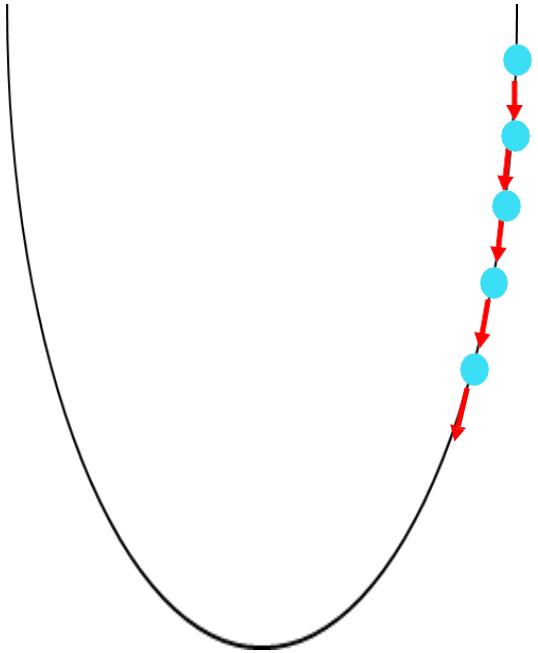


How the Gradient Descent works



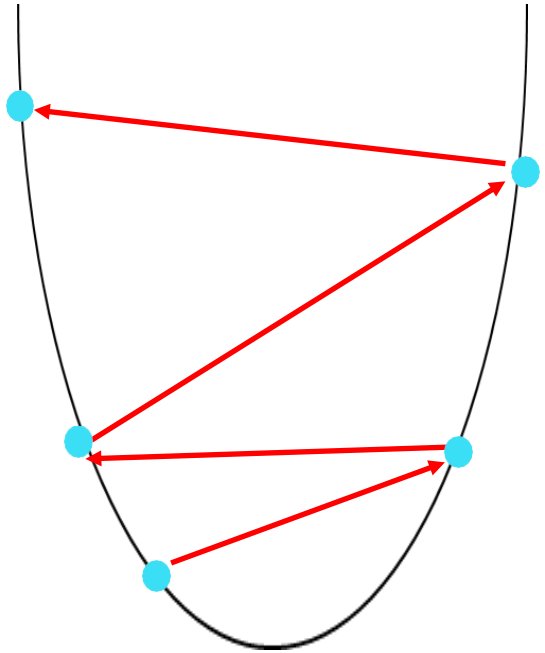
How the learning rate effects the learning

If $\propto$ is too small



Too long time to take to converge

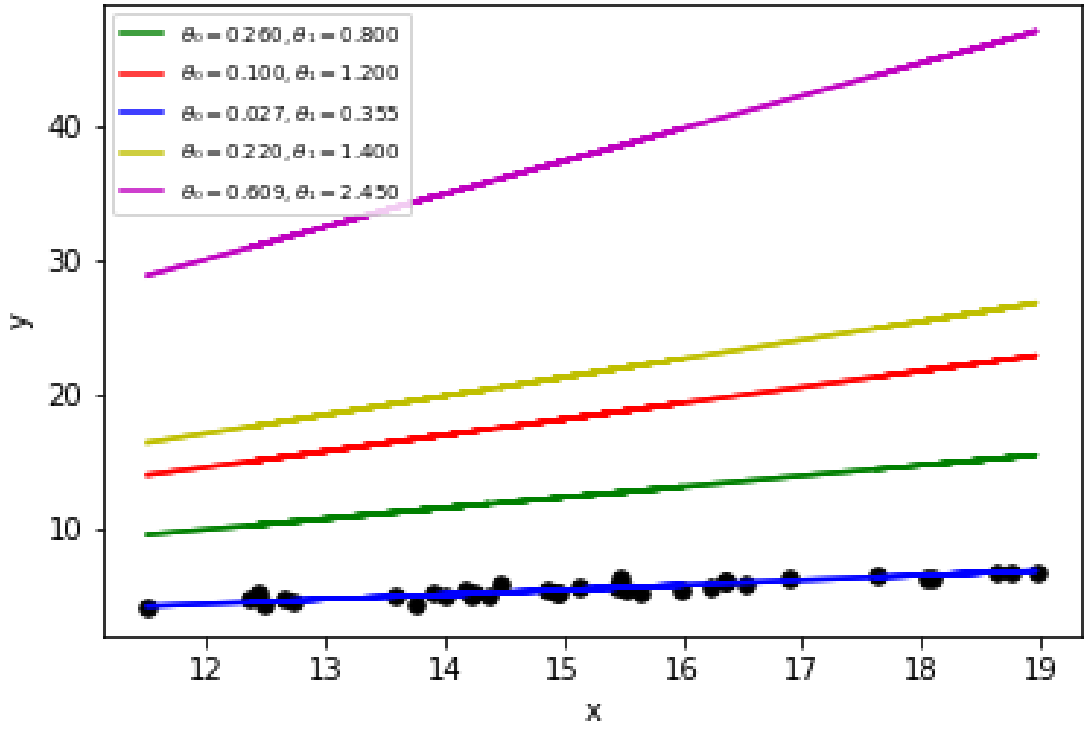
If $\propto$ is too large



It may fail to converge, causing gradient explode (loss increases)

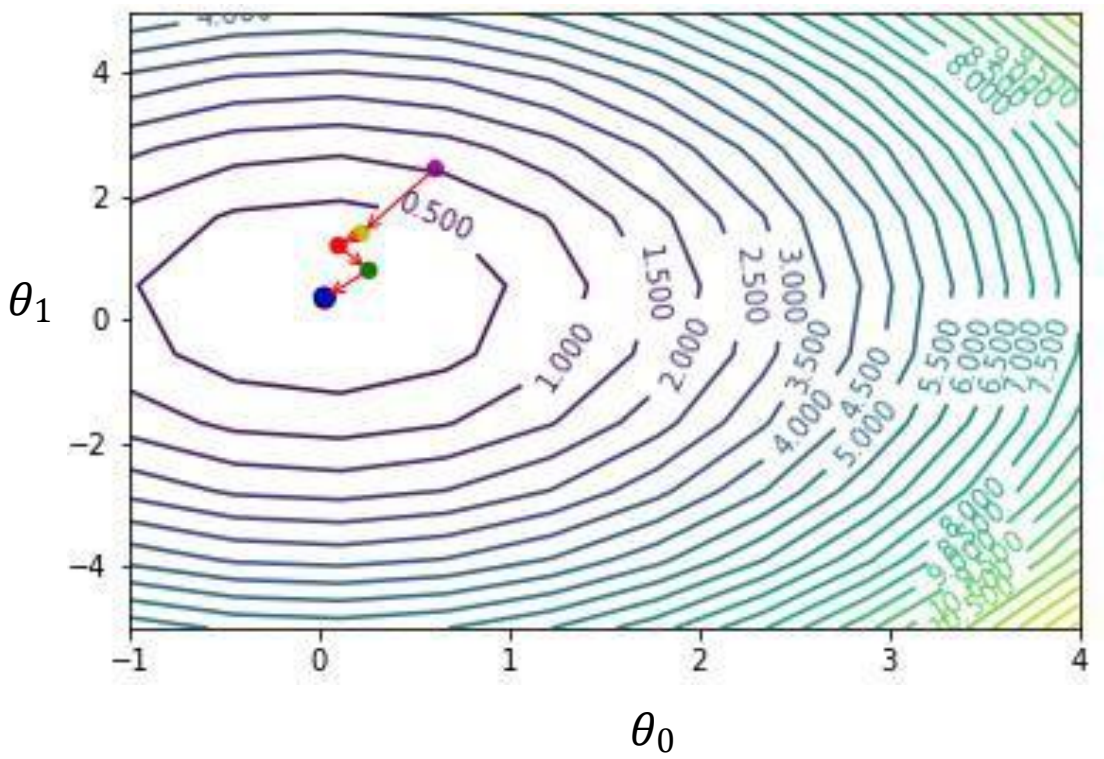
Gradient Descent to optimize $\boldsymbol{\theta_1}$ and $\boldsymbol{\theta_0}$

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



Iterative process to obtain $min_{\theta_1} J(\theta_1)$

$$\theta_0 \leftarrow \theta_0 - \propto \frac{\partial J}{\partial \theta_0} \qquad \theta_1 \leftarrow \theta_1 - \propto \frac{\partial J}{\partial \theta_1}$$

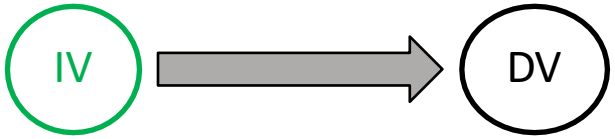


Multiple linear regression

- Multiple linear regression (MLR) is a method used to model the linear relationship between **two or more** independent variables and a dependent variable.
    - The price of a house is corelated to its size in square feet and the number of bedrooms.
- MLR based on the assumption that the independent variables are not too highly correlated with each other.
    - The size and number of bedrooms are not highly correlated.

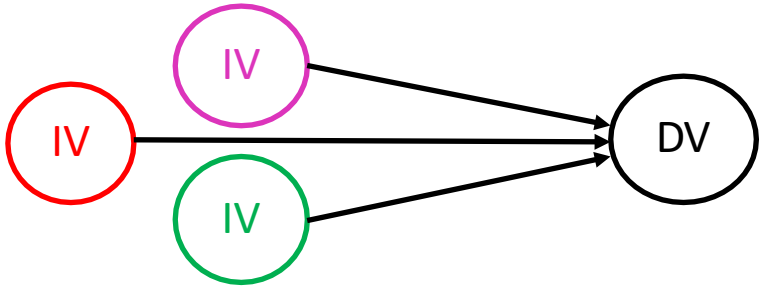
Simple vs. Multiple linear regression

- Simple linear regression - one-to-one



where,
IV: independent variables
DV: dependent variable

- Multiple linear regression - many-to-one



Multiple linear regression

- For example, from a series of N training set, to model the relationship between the *height* and *width* of sea bream.
- Hypothesis of a simple linear regression is represented as:

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

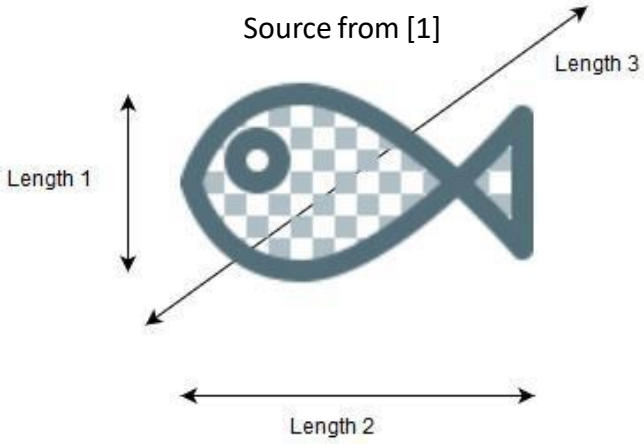
Height, x	Width, y
11.52	4.02
12.48	4.3056
12.3778	4.6961
12.73	4.4555
12.444	5.134
⋮	⋮
⋮	⋮

Multiple linear regression

- Other than *height*, to model the linear relationship betwe more independent variables such as the *weight*, *body height* and *diagonal length* with the dependent variable, *width*, the hypothesis is represented as follows:

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \theta_4 x_4 + \theta_5 x_5$$

Source from [1]



Weight, x_5	Length 1 (Body Height), x_4	Length 2 (Total Length), x_3	Length 3 (Diagonal Length), x_2	Height, x_1	Width, y
242	23.2	25.4	30	11.52	4.02
290	24	26.3	31.2	12.48	4.3056
340	23.9	26.5	31.1	12.3778	4.6961
363	26.3	29	33.5	12.73	4.4555
430	26.5	29	34	12.444	5.134
450	26.8	29.7	34.7	13.6024	4.9274
500	26.8	29.7	34.5	14.1795	5.2785
⋮	⋮	⋮	⋮	⋮	⋮

Multiple linear regression

- Hypothesis $h_\theta(x)$ is represented as:

$$h_\theta(x) = \theta_0 x_0 + \theta_1 x_1 + \theta_2 x_2 + \ldots + \theta_M x_M$$

- Parameters: $\theta_0, \ldots, \theta_M$ Number of parameters = $M + 1$
- Cost function:

$$\mathrm{J}(\theta) = \frac{1}{2N} \sum_{n=1}^{N} \left(h\ (x^{(n)}) - y^{(n)} \right)^2$$ Number of training data = N

- Goal:

$$minimize\ \ \mathrm{J}(\theta)$$

Multiple linear regression

- Two solutions for the MLR
  - Normal equation
  - Gradient Descents

Normal Equation

$$\theta = (X^T X)^{-1} X^T y$$

where X is a $N \times (M+1)$ matrix
Y is a $N \times 1$ matrix

Gradient Descent

Repeat

$$\{ \quad \boldsymbol{\theta_j} \leftarrow \boldsymbol{\theta_j} - \propto \frac{\partial \boldsymbol{J}}{\partial \boldsymbol{\theta_j}}$$

$$\}$$

θ is updated simultaneously for j = 0,…,M

Normal equation vs Gradient descent

SWINBURNE UNIVERSITY OF TECHNOLOGY SARAWAK CAMPUS

Normal Equation

- Pros
  - No need to adjust learning rate.
  - No iterative training.
- Cons

$$(X^T X)^{-1}$$

  - Computational expensive when number of parameters learned is too large (a hundred ~ ten thousand).
  - It is possible that $(X^T X)^{-1}$ is non-reversible if there are **redundant features** or **too many parameters.**

Gradient Descent

- Cons
  - Need to adjust learning rate.
  - Need iterative training.
- Pros
  - Still works efficiently when number of parameters learned is very large.

Polynomial regression

$$h_\theta(x) = \theta_0 x^0 + \theta_1 x^1 + \theta_2 x^2 + \ldots + \theta_M x^M$$

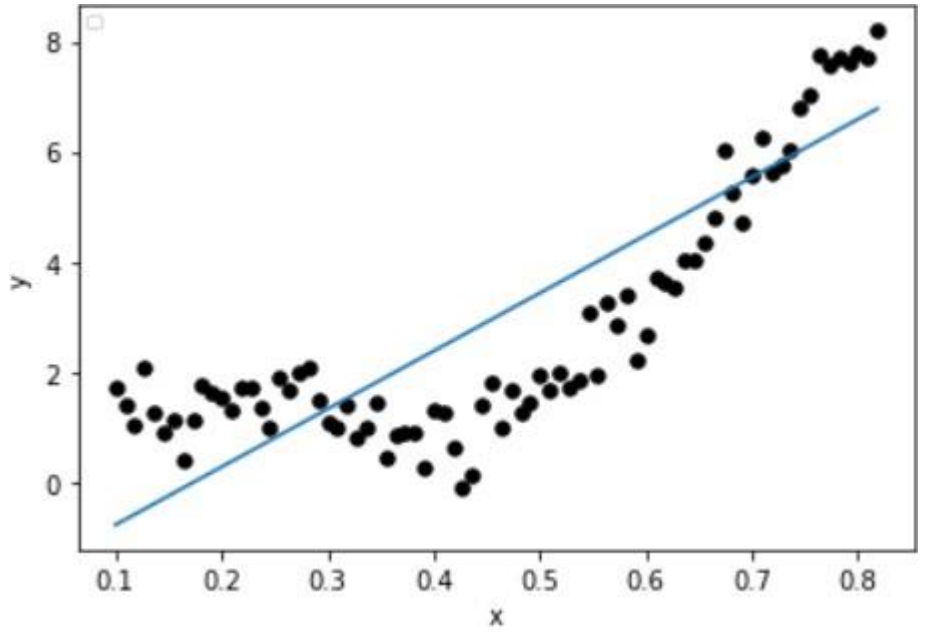
- In statistics, polynomial regression is a form of regression analysis in which the relationship between the independent variable x and the dependent variable y is modelled as an ***j*th degree** polynomial in *x*.

Polynomial regression

- Simple linear regression model ($h_\theta\ (x\ ) =\ \theta_0 + \theta_1 x\ $) could not fit the data well if the independent data, x exhibits nonlinear relationship with the dependent data, y

x	y
0.368	0.667
0.401	0.792
0.434	1.247
0.468	0.563
0.502	1.792
⋮	⋮
⋮	⋮

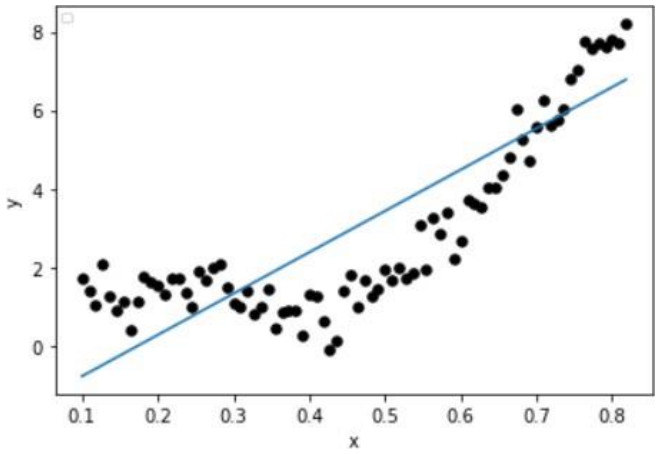
$$h_\theta(x) = \theta_0 + \theta_1 x$$



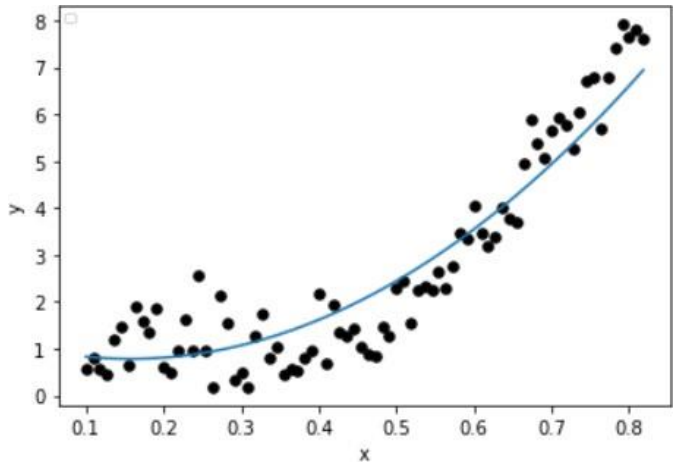
Polynomial regression

- Higher order Polynomial regression model can better fit the training data.

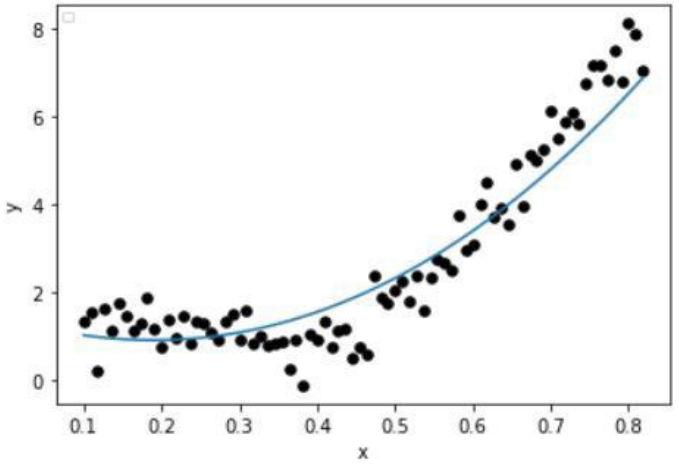
$h_\theta(x) = \theta_0 + \theta_1 x$



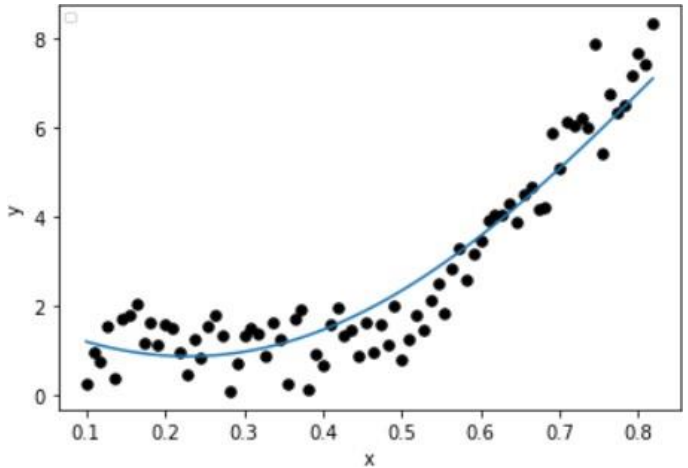
$h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2$



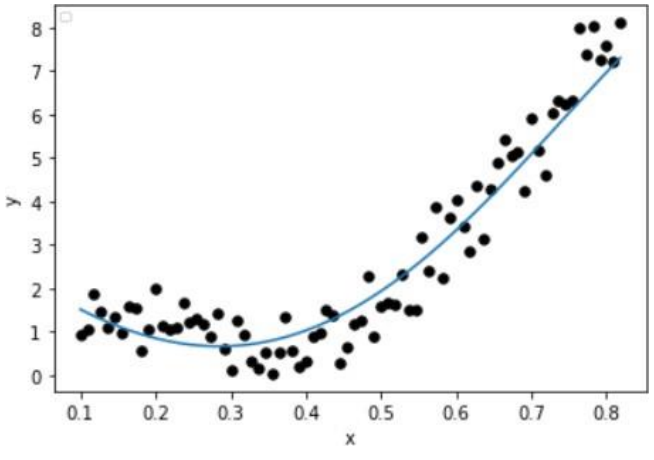
$h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3$



$h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$



$h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4 + \theta_5 x^5$



Polynomial regression

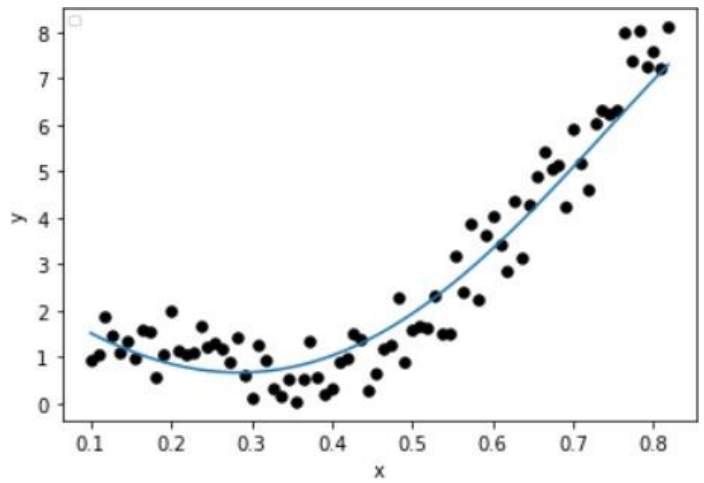
- Polynomial regression is consider a special case of multiple linear regression.
- Although polynomial regression fits a nonlinear model to the data, it is linear to the parameters $\theta_0, \theta_1, .., \theta_j$.

Polynomial linear regression | Multiple linear regression

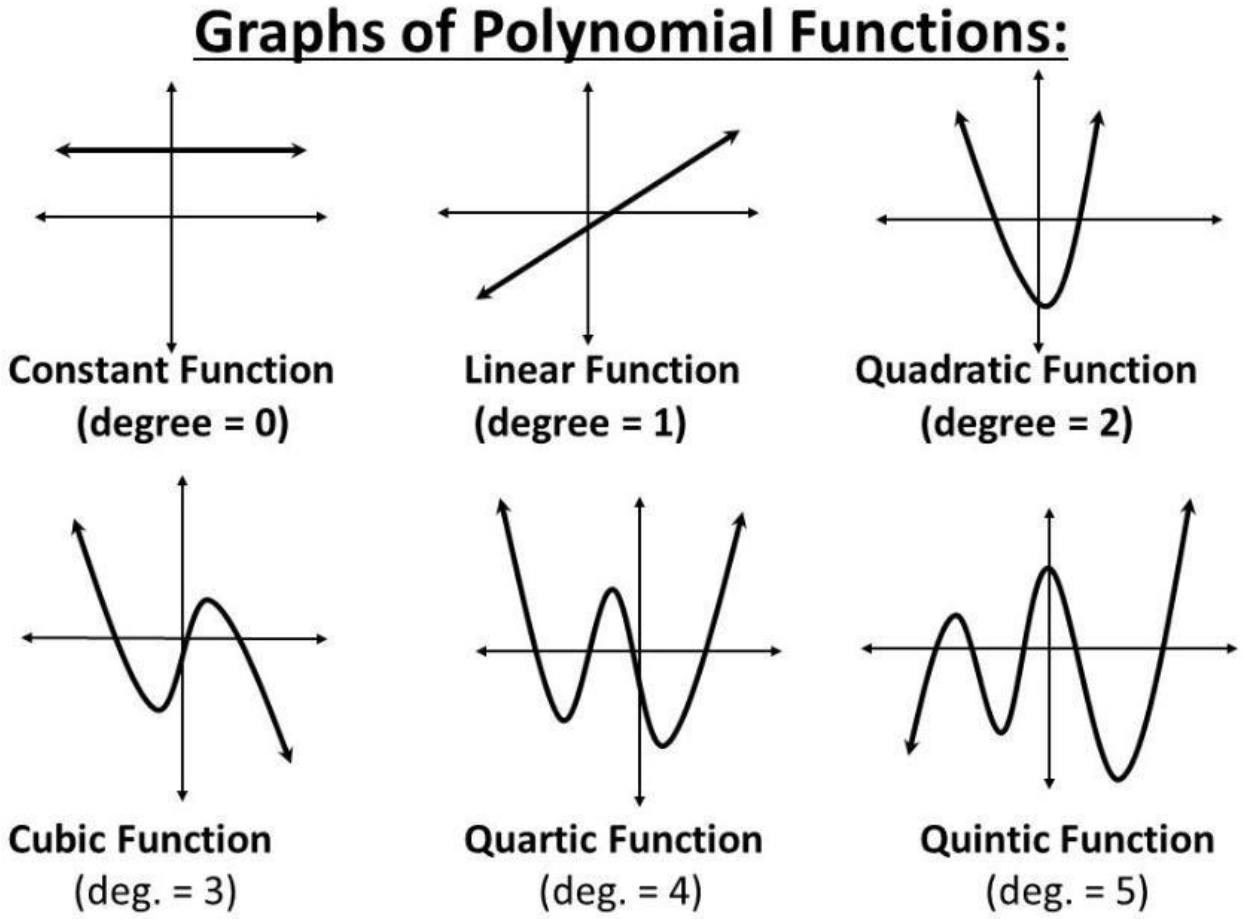
$$h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \ldots + \theta_5 x^5 \quad \subseteq \quad \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \ldots + \theta_5 x_5$$

$(x)^j$ is using the same variable x

$(x)^2$	x	y
$(0.368)^2$	0.368	0.667
$(0.401)^2$	0.401	0.792
$(0.434)^2$	0.434	1.247
$(0.468)^2$	0.468	0.563
$(0.502)^2$	0.502	1.792
⋮	⋮	⋮



Polynomial Functions with varying degree



Different types of polynomial functions

[4] Brandon John Grenier, (2017, January 27), Multivariate Polynomial Regression <http://brandon.ai/2017/01/27/multivariate-polynomial-regression.html>

Overfitting

- Although a higher order polynomial model may allow better modeling of the data, it is likely to lead to overfitting.
- What is overfitting?
  - A model that can fit well to the training data, but not generalized to new and unseen testing data.
  - Low bias, high variance
- Normal causes of overfitting:
  - Model is too complex.
  - Model was trained with a small amount of training data.

Bias: Reflects how well the model captures the true patterns. Low bias means capturing the patterns accurately, while high bias means missing them.
Variance: Indicates how much the model's predictions change with different training data. High variance means a lot of change, low variance means little change.

Underfitting

- What is Underfitting?
  - A model is too simple that performs poorly on both the training and testing data
  - High bias, low variance

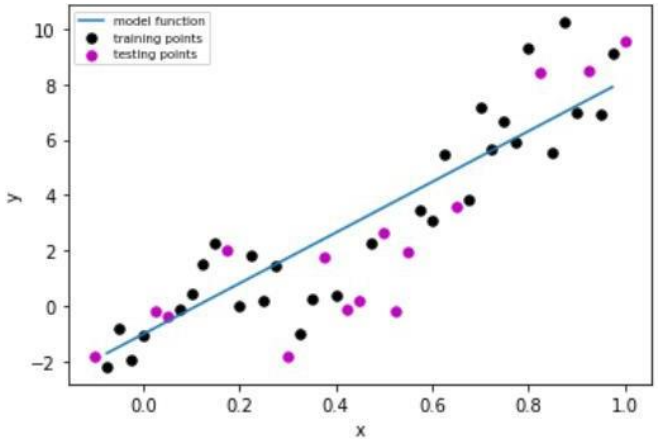
- Normal causes of Underfitting:
  - Model is too simple. An example of this situation would be the construction of a linear regression model on non-linear data.

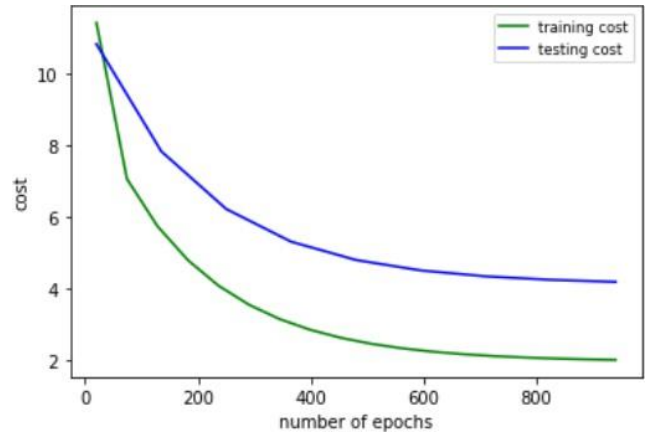
Examples of Overfitting and Underfitting

Underfitting: Model performs badly in both training and testing data

degree = 1

```
Epoch: 700 - Train Error: 2.1303 - Test Error: 4.3432
Epoch: 720 - Train Error: 2.1123 - Test Error: 4.3209
Epoch: 740 - Train Error: 2.0961 - Test Error: 4.3016
Epoch: 760 - Train Error: 2.0815 - Test Error: 4.2837
Epoch: 780 - Train Error: 2.0684 - Test Error: 4.2672
Epoch: 800 - Train Error: 2.0567 - Test Error: 4.2514
Epoch: 820 - Train Error: 2.0461 - Test Error: 4.2380
Epoch: 840 - Train Error: 2.0366 - Test Error: 4.2269
Epoch: 860 - Train Error: 2.0281 - Test Error: 4.2143
Epoch: 880 - Train Error: 2.0204 - Test Error: 4.2039
Epoch: 900 - Train Error: 2.0135 - Test Error: 4.1946
Epoch: 920 - Train Error: 2.0073 - Test Error: 4.1854
Epoch: 940 - Train Error: 2.0017 - Test Error: 4.1781
Converged.
```

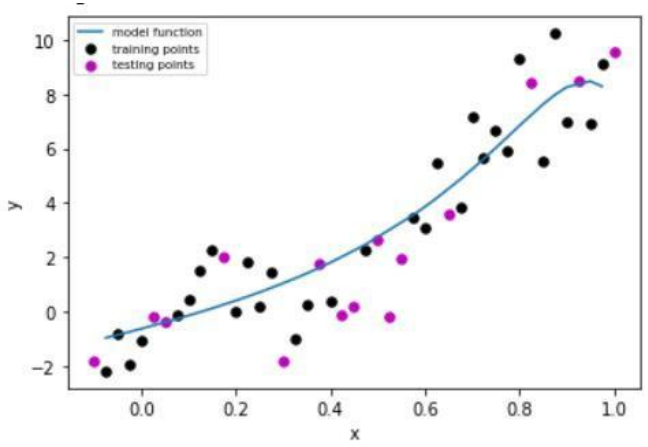


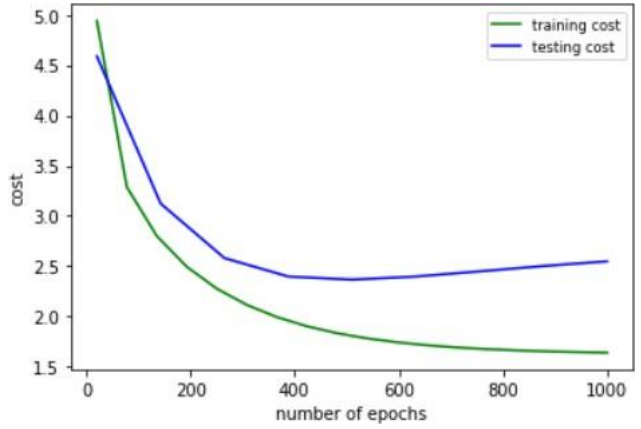


Overfitting: Model performs too well on the training data but the performance drops significantly over the testing data

degree = 19

```
Epoch: 740 - Train Error: 1.6817 - Test Error: 2.4412
Epoch: 760 - Train Error: 1.6761 - Test Error: 2.4500
Epoch: 780 - Train Error: 1.6711 - Test Error: 2.4589
Epoch: 800 - Train Error: 1.6665 - Test Error: 2.4677
Epoch: 820 - Train Error: 1.6624 - Test Error: 2.4763
Epoch: 840 - Train Error: 1.6586 - Test Error: 2.4847
Epoch: 860 - Train Error: 1.6552 - Test Error: 2.4937
Epoch: 880 - Train Error: 1.6521 - Test Error: 2.5019
Epoch: 900 - Train Error: 1.6493 - Test Error: 2.5104
Epoch: 920 - Train Error: 1.6468 - Test Error: 2.5181
Epoch: 940 - Train Error: 1.6445 - Test Error: 2.5260
Epoch: 960 - Train Error: 1.6423 - Test Error: 2.5337
Epoch: 980 - Train Error: 1.6404 - Test Error: 2.5406
Epoch: 1000 - Train Error: 1.6387 - Test Error: 2.5478
Converged.
```



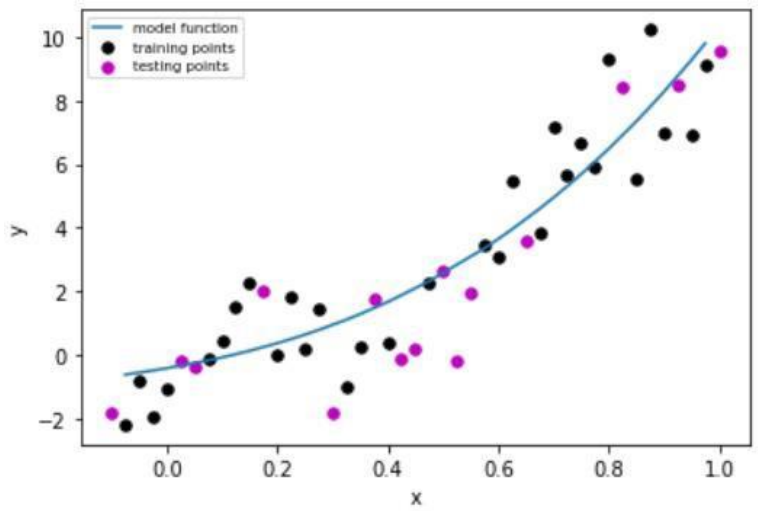


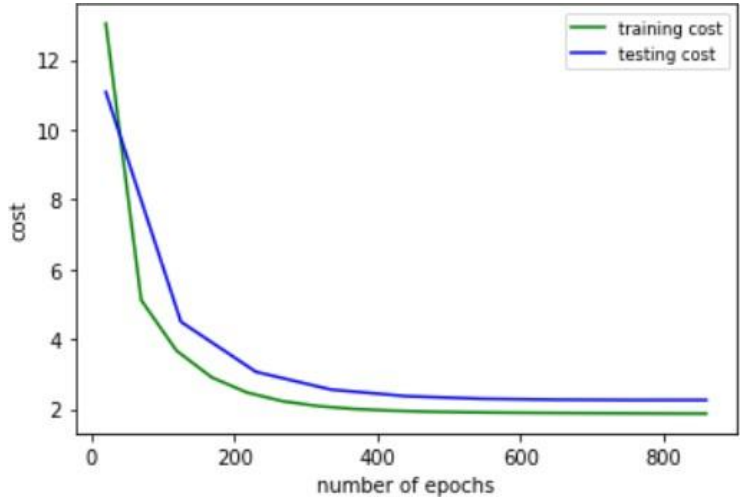
Examples of Overfitting and Underfitting

Degree = 3

```
Epoch: 640 - Train Error: 1.8837 - Test Error: 2.2646
Epoch: 660 - Train Error: 1.8814 - Test Error: 2.2605
Epoch: 680 - Train Error: 1.8793 - Test Error: 2.2590
Epoch: 700 - Train Error: 1.8773 - Test Error: 2.2563
Epoch: 720 - Train Error: 1.8755 - Test Error: 2.2550
Epoch: 740 - Train Error: 1.8738 - Test Error: 2.2532
Epoch: 760 - Train Error: 1.8722 - Test Error: 2.2528
Epoch: 780 - Train Error: 1.8706 - Test Error: 2.2526
Epoch: 800 - Train Error: 1.8691 - Test Error: 2.2520
Epoch: 820 - Train Error: 1.8677 - Test Error: 2.2521
Epoch: 840 - Train Error: 1.8663 - Test Error: 2.2522
Epoch: 860 - Train Error: 1.8649 - Test Error: 2.2530
Converged.
```

An acceptable model





Methods to overcome Overfitting

- Dealing with the complexity of a model by reducing the number of features by the following methods :
  - Manual selection of features
  - Model selection

- Adding regularization penalties
  - The size of the features is retained but the magnitude of the weights θ_j is reduced.

- Collecting more training data

Regularized linear regression

- Regularization is one of the techniques used to deal with overfitting by avoiding the learning of a more complex or flexible model.

$$J(\theta) = \frac{1}{2N}\left[\sum_{n=1}^{N}\left(h\ (x^{(n)}) - y^{(n)}\right)^2 + \boxed{\lambda \sum_{j=1}^{M} \theta_j^{\,2}}\right]$$

regularization term

- This regularization method is called L2 regularization.
- A regression model that uses L2 regularization technique is called ***Ridge Regression***.

Regularized linear regression

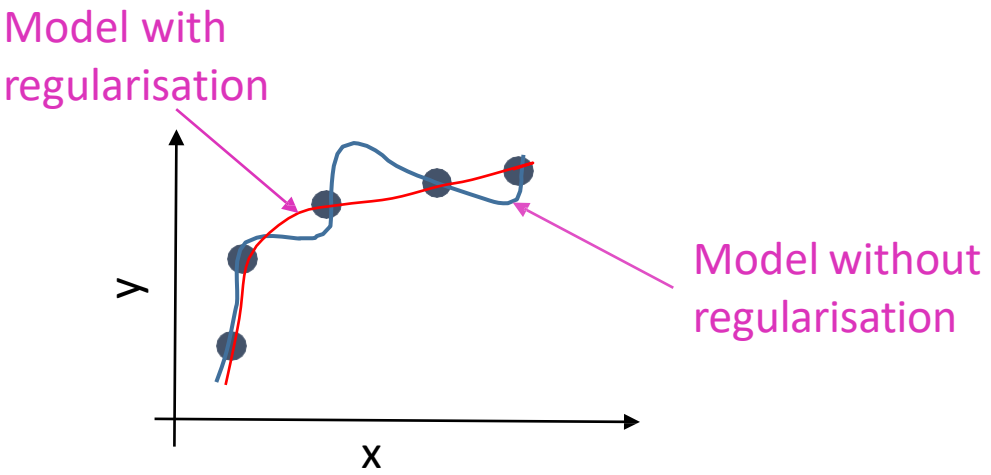
$$J(\theta) = \frac{1}{2N}\left[\sum_{n=1}^{N}\left(h\ (x^{(n)}) - y^{(n)}\right)^2 + \boxed{\lambda \sum_{j=1}^{M} \theta_j^{\,2}}\right]$$

regularization term

Regularization parameter (weight decay)

- Goal: minimise J(θ)

- To minimize J(θ), the learned model will try to shrink the regularization term by reducing the θ_j towards zero.
- The values of θ_j decrease and become smaller, leading to a simpler hypothesis/model.



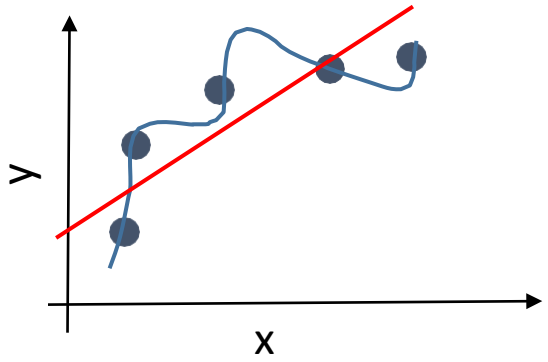
Regularized linear regression

$$J(\theta) = \frac{1}{2N}\left[\sum_{n=1}^{N}\left(h\ (x^{(n)}) - y^{(n)}\right)^2 + \boxed{\lambda \sum_{j=1}^{M} \theta_j^2}\right]$$

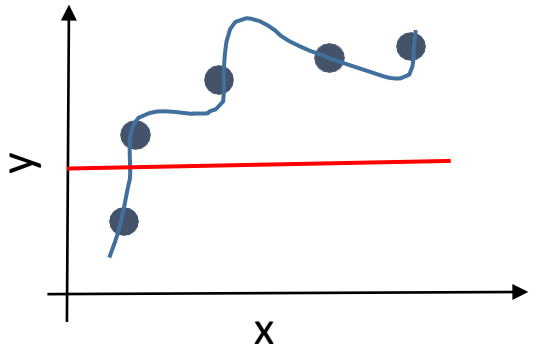
regularization term

- When λ is too *large*, the learned model will force the θ_j to shrink to a larger extent towards zero. The hypothesis will become almost linear.

$$h_\theta(x) = \theta_0 + \theta_1 x + \overset{\approx 0}{\theta_2 x^2} + \overset{\approx 0}{\theta_3 x^3}$$



$$h_\theta(x) = \theta_0 + \overset{\approx 0}{\theta_1 x} + \overset{\approx 0}{\theta_2 x^2} + \overset{\approx 0}{\theta_3 x^3}$$



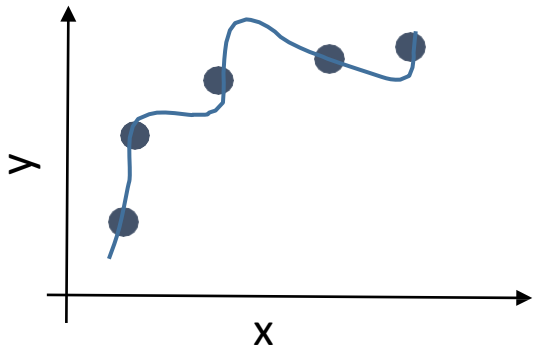
- Using a very large value of λ can lead to *underfitting* of the training set.

Regularized linear regression

$$J(\theta) = \frac{1}{2N}\left[\Sigma^{N}_{n=1}\left(h_\theta(x^{(n)}) - y^{(n)}\right)^2 + \lambda \Sigma^{M}_{j=1}\theta_j^2\right]$$

- When λ is too *small*, the regularisation term has almost no effect in regularizing the θ_j.

$$h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3$$



Regularized linear regression

$$J(\theta) = \frac{1}{2N}\left[\sum_{n=1}^{N}\left(h\ (x^{(n)}) - y^{(n)}\right)^2 + \boxed{\lambda \sum_{j=1}^{M} |\ \theta_j\ |}\ \right]$$

L1 regularization term

- This regularization method is called L1 regularization.
- A regression model that uses L1 regularization technique is called **Lasso Regression**.
- Lasso has the property that is able to shrink some of the weights to zero. Therefore, that feature can be removed from the model.
  - Lasso can be used to select important features of a dataset

Next Lecture

❖Logistic regression