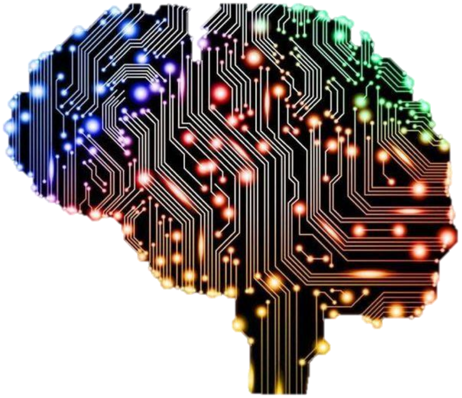
智慧化企業整合
Intelligent Integration of Enterprise

Introduction of Deep Learning

助教:黃日泓

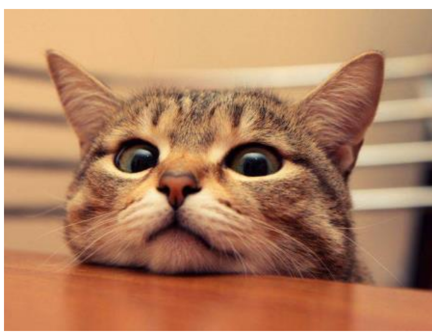
Outline

- Neural Network
- Loss function
- Gradient Descent



System

Input data	Model	Outputs



$f(x)$

Cat: 50%
Dog: 35%
Bird: 15%

Steps of Define Functions

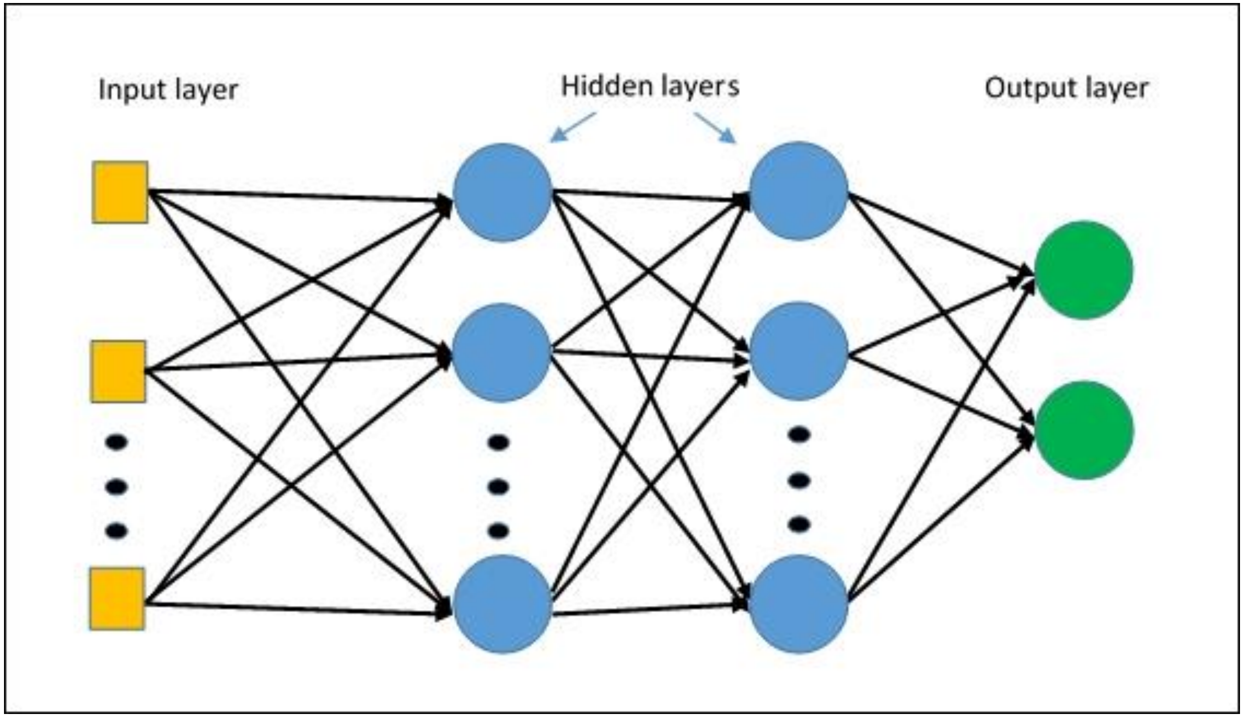
Step1: Define your network (function) structure

Step2: Measure the goodness of your function

Step3: Find the best function

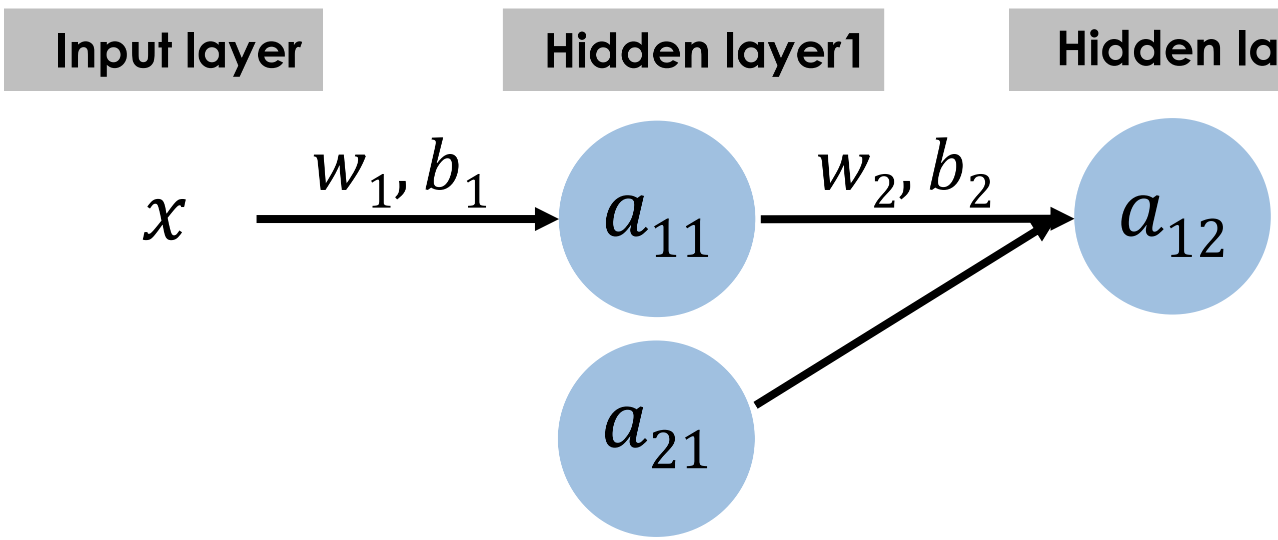
Step1: Function Structure

Neural Network



- Fully connected feedforward neural network
- A large function set that consists of lots of variables

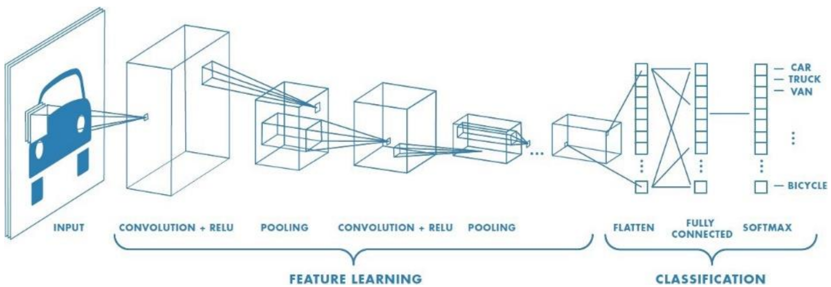
Neurons



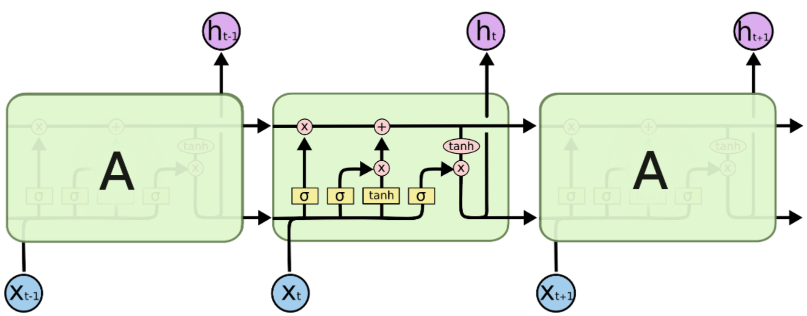
$$z_{11} = w_1 x + b_1$$
$$a_{11} = a(z_{11})$$
$$z_{12} = w_2 a_{11} + b_2$$
$$a_{12} = a(z_{12})$$

- $a(z)$ is an activation function
- $a(z)$ could be different in different layers
- # of neurons in different layers could be different

Common Architectures



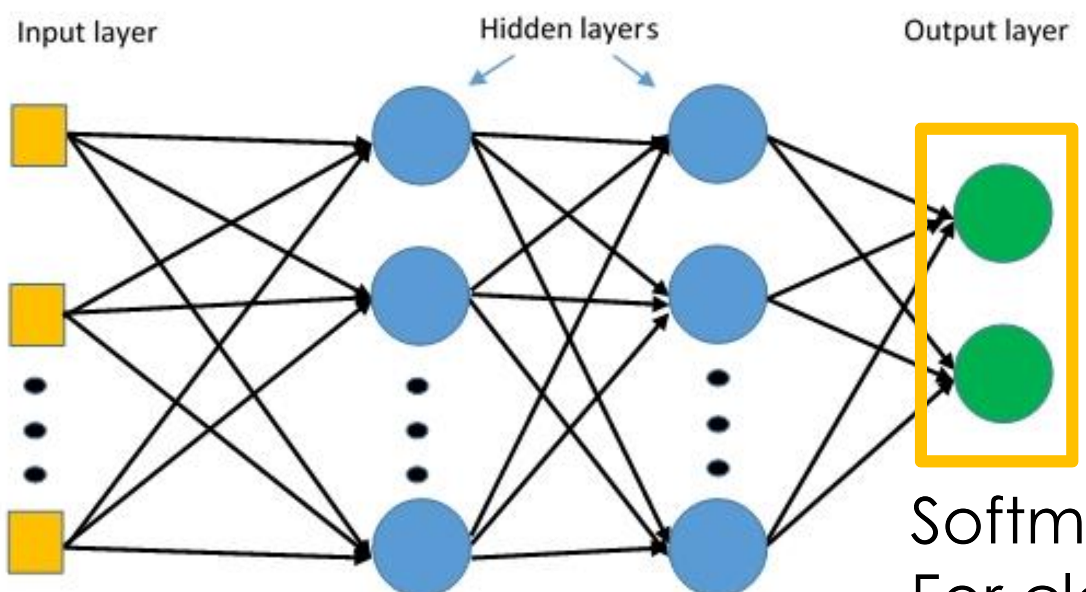
Convolutional neural network (CNN)



The repeating module in an LSTM contains four interacting layers.

Recurrent neural network (RNN/LSTM cell)

Output layer



Softmax Function
For classification task

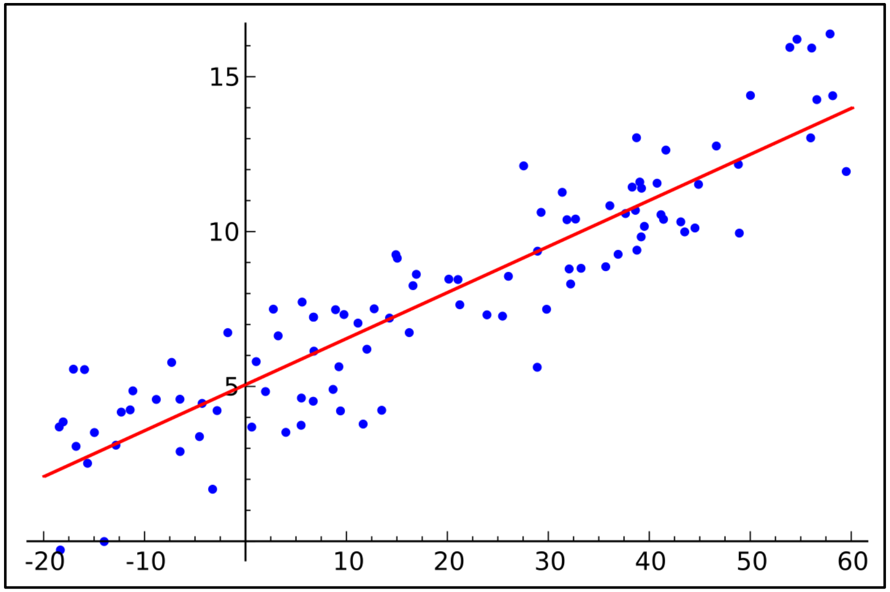
Softmax Function

$$\sigma(\mathbf{z})_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}} \quad \text{for } j = 1, \ldots, K.$$

- Output : [0,1]
- Sum of outputs is 1
- Simulation of probability
- Usually used in classification problems

Step2: Goodness of Function

Loss Function



- A measure of goodness
- Distance (error rate) between predictions and true labels
- Common loss functions: MSE, MAE, Cross entropy…

Loss Function

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|y_i - \hat{y}_i\right|$$

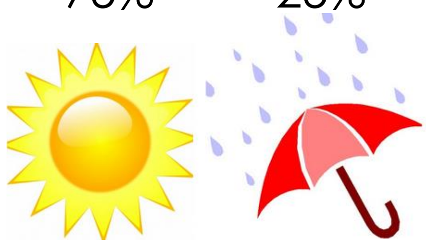
- Distance (error rate)
- Minimize loss function (Step3)

Shannon's Entropy

50% 50%

$$-0.5 \times \log_2 0.5 - 0.5 \times \log_2 0.5 = 1$$

75% 25%

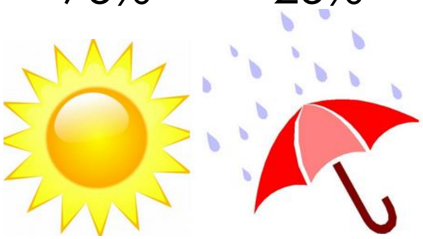


$$-0.75 \times \log_2 0.75 - 0.25 \times \log_2 0.25$$
$$= 0.31 + 0.5 = 0.81$$

$$S = -\sum_i P_i \log P_i .$$ [1]

Cross Entropy

75% 25%



$$-0.75 \times \log_2 0.75 - 0.25 \times \log_2 0.25$$
$$= 0.31 + 0.5 = 0.81$$

25% 75%

$$-0.75 \times \log_2 0.25 - 0.25 \times \log_2 0.75$$
$$= 1.5 + 0.1 = 1.6$$

$$H(p,q) = -\sum_x p(x)\, \log q(x).$$

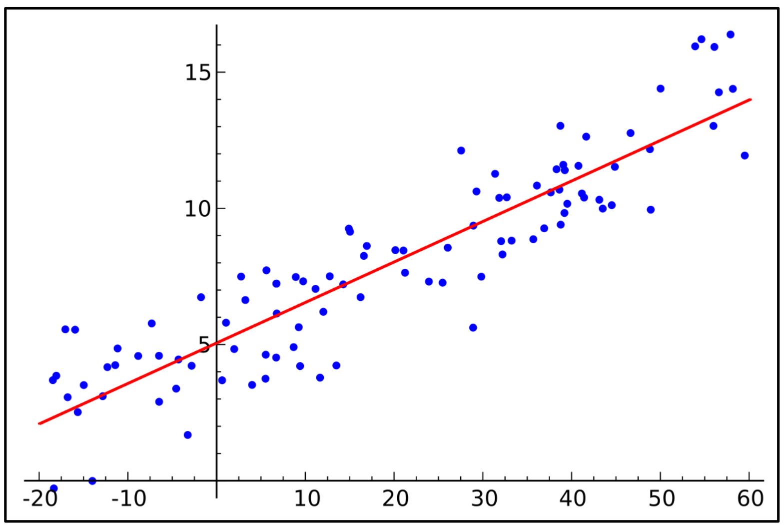
- Usually used in classification task
- Predicted probability distribution: $q(x)$
- Real probability distribution: $p(x)$
- Goal: minimize cross entropy

Step3: The best Function

国立清華大學
National Tsing Hua University

INNOVATION ENGINEERING LAB

Goal



- Assume y = wx + b
- w, b are coefficients we want to le
- Minimize loss function (EX: MSE)

$$L(w,b)=\sum_{n=1}^{k}\left(\hat{y}^n-(b+w\times x^n)\right)^2$$

Gradient

$$\nabla f = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \\ \vdots \end{bmatrix}$$

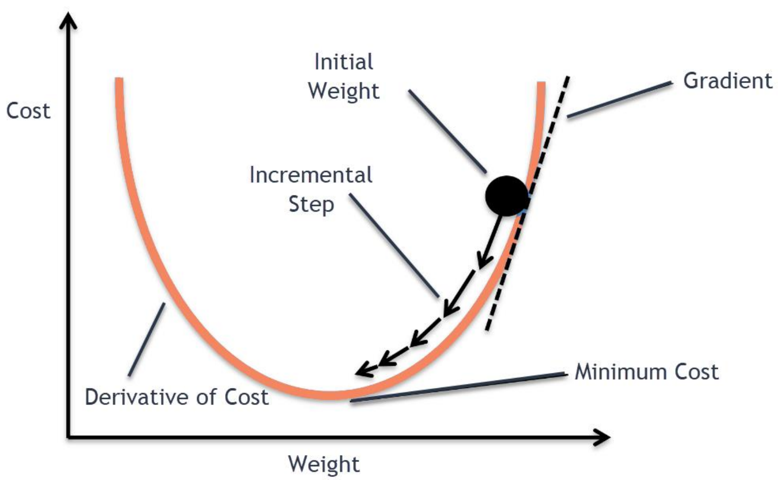
- Partial derivative of all variable
- The direction you should travel to increase the value of f most rapidly

$$L(w,b) = \sum_{n=1}^{k} \left(\hat{y}^n - (b + w \times x^n)\right)^2$$

國立清華大學
National Tsing Hua University

INNOVATION ENGINEERING LAB

Gradient Descent



$$L(w, b) = \sum_{n=1}^{k} \left(\hat{y}^n - (b + w \times x^n)\right)^2$$

- w, b are coefficients we want to le
- We want to minimize $L(w, b)$
  1. Choose an initial weight
  2. Calculate the gradient
  3. Move in the opposite directio

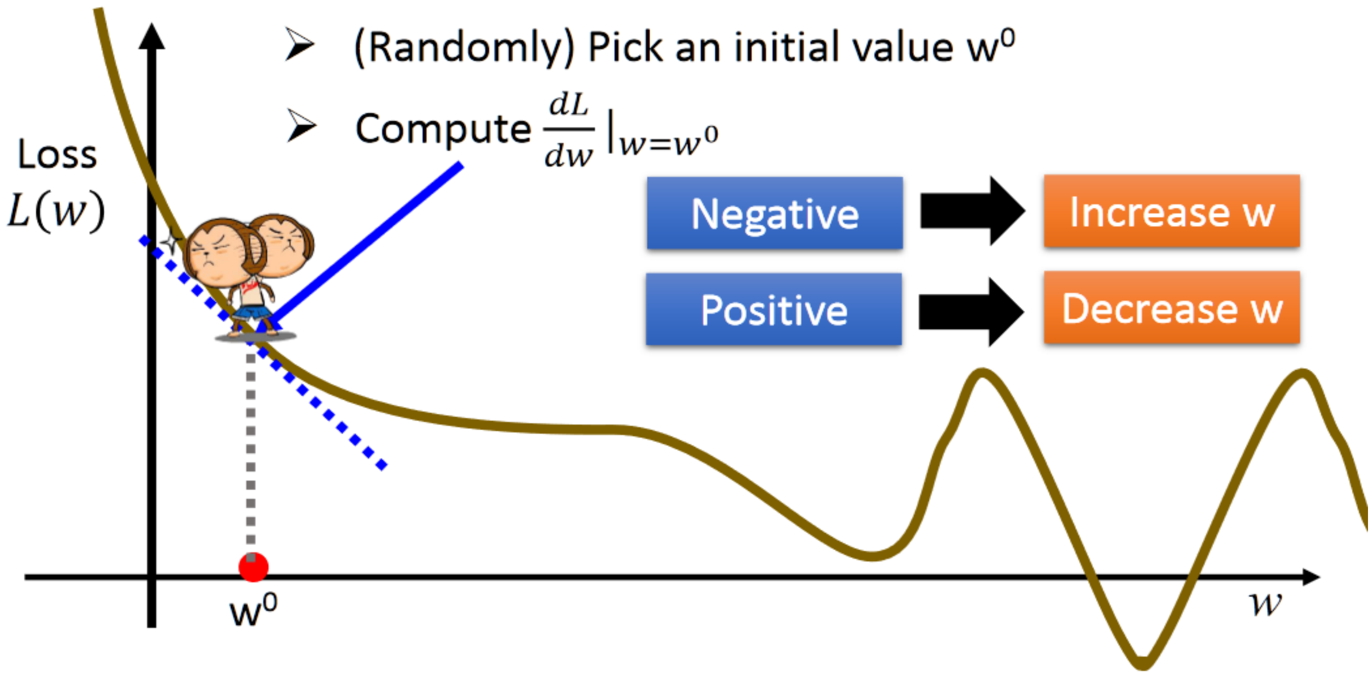
Gradient Descent

$$w^* = arg \min_w L(w)$$

- Consider loss function $L(w)$ with one parameter w:
  - ➢ (Randomly) Pick an initial value w^0
  - ➢ Compute $\frac{dL}{dw}|_{w=w^0}$

Negative → Increase w

Positive → Decrease w



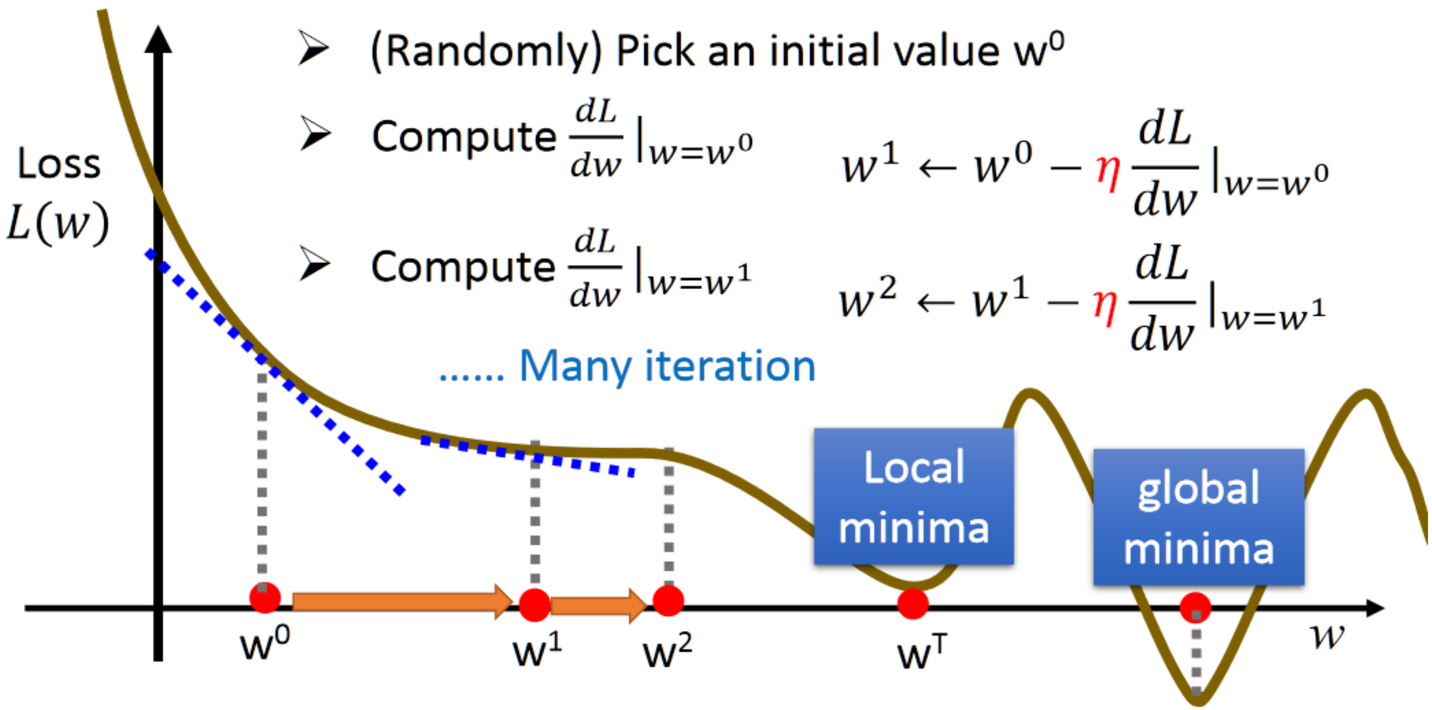
http://speech.ee.ntu.edu.tw/~tlkagk/courses/ML_2017/Lecture/Regression.pdf

Gradient Descent

$$w^* = arg \min_{w} L(w)$$

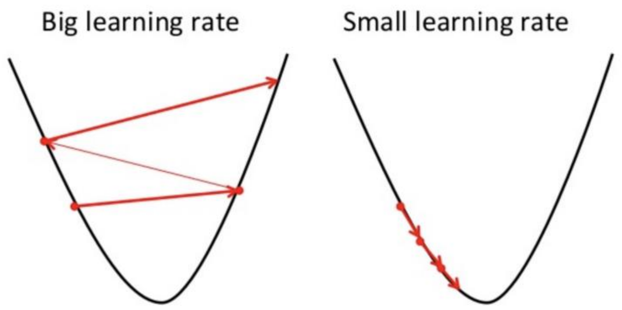
- Consider loss function $L(w)$ with one parameter w:
  - (Randomly) Pick an initial value w^0
  - Compute $\frac{dL}{dw}|_{w=w^0}$ $\quad w^1 \leftarrow w^0 - \eta \frac{dL}{dw}|_{w=w^0}$
  - Compute $\frac{dL}{dw}|_{w=w^1}$ $\quad w^2 \leftarrow w^1 - \eta \frac{dL}{dw}|_{w=w^1}$

…… Many iteration

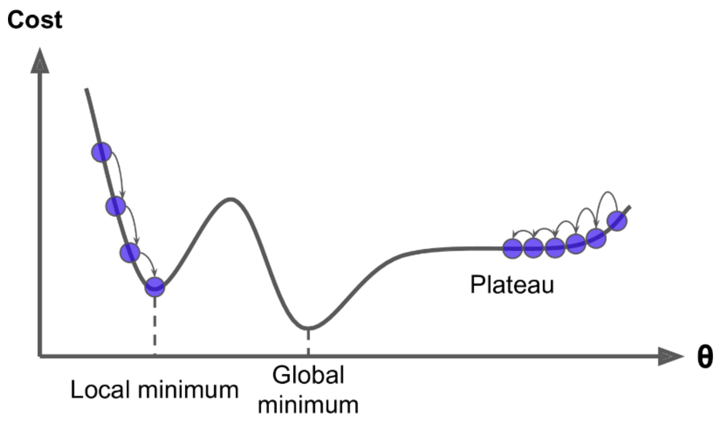


http://speech.ee.ntu.edu.tw/~tlkagk/courses/ML_2017/Lecture/Regression.pdf

Problems of Gradient Descent



- Small learning rate → slow training process
- Big learning rate → difficult to converge



- Training process may be hindered by local minimum or plateau
- We need to adjust learning rate to improve this process (Adagrad, Momentum, Adam…)

References

- Regression and gradient descent by Hung-Yi Lee
- Introduction of entropy
- Introduction of loss function

Class assignment

- Please search , read and learn the definition of the following terms.
- Write down your own introduction of these terms in your report.
- Turn in your report with the format of pdf
- (maximum: 3 pages).

  1. **Stochastic gradient descent (SGD)**
  2. **Momentum (in machine learning)**
  3. **Adagrad**
  4. **RMS prop**

Homework

- Please refer to the ‘gradient descent.ipynb’ and see how this algorithm updates parameters
- Please modify some hyper parameters of this notebook and illustrate of those modifications with corresponding effects in this notebook
- Turn in this homework with the format of ipynb