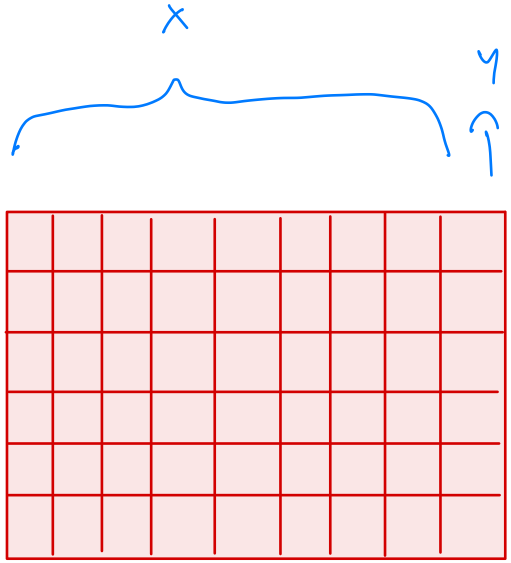
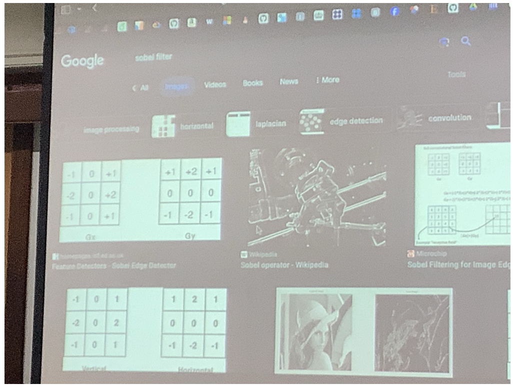
Classical



$\{x_1, x_2, x_3, \dots x_n\} \mid x \in \mathbb{R}$

Deeplearning



-1	0	+1
-2	0	+2
-1	0	+1

+1	+2	+1
0	0	0
-1	-2	-1

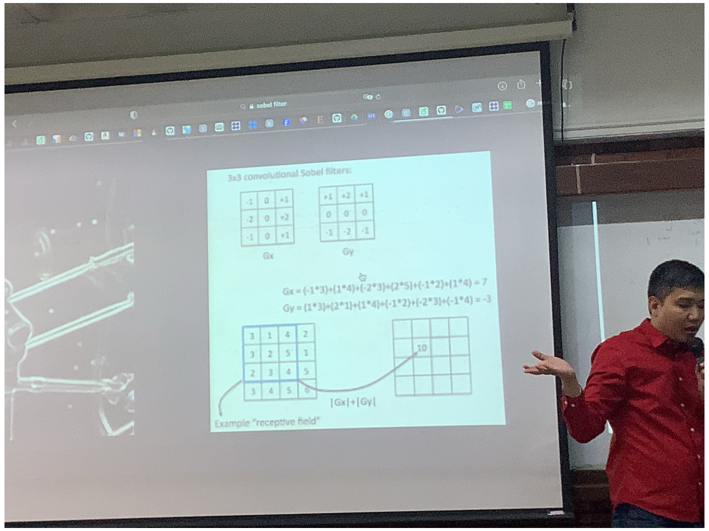
homepages.inf.ed.ac.uk
Feature Detectors - Sobel Edge Detector

Wikipedia
Sobel operator - Wikipedia

Microchip
Sobel Filtering for Image Edg

-1	0	1
-2	0	2
-1	0	1

1	2	1
0	0	0
-1	-2	-1



3x3 convolutional Sobel filters:

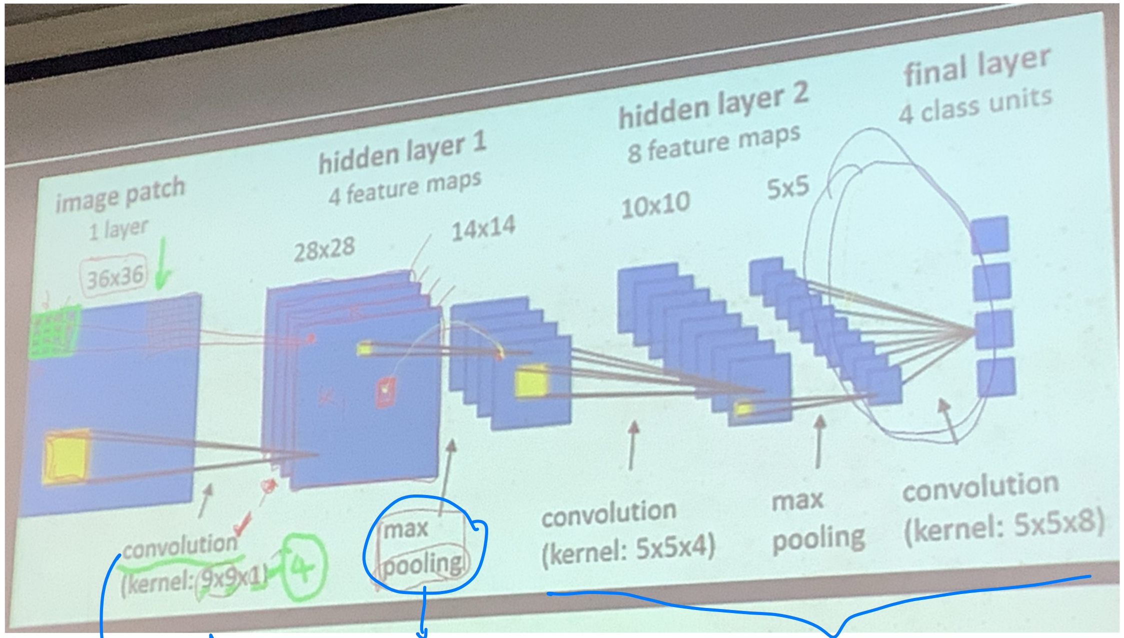
-1	0	+1
-2	0	+2
-1	0	+1

+1	+2	+1
0	0	0
-1	-2	-1

Gx = (-1*3)+(1*4)+(-2*3)+(2*5)+(-1*2)+(1*4) = 7

Gy = (1*3)+(2*1)+(1*4)+(-1*2)+(-2*3)+(-1*4) = -3

3	1	4	2
3	2	5	1
2	3	4	5
3	4	5	6



หาจุดเด่น

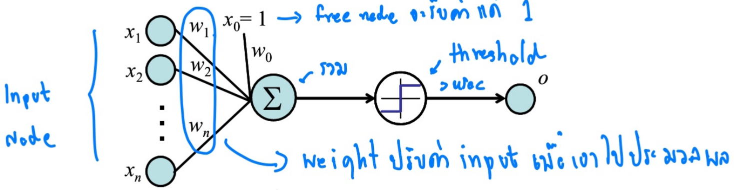
ดูค่าองค์รวม
เอาค่ามากที่สุด

ทำซ้ำ

6.7.1 เพอร์เซปตรอน

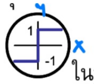
เพอร์เซปตรอน (perceptron) เป็นข่ายงานประสาทเทียมแบบง่ายมีหน่วยเดียวที่จำลองลักษณะของเซลล์ประสาทดังรูปที่ 6–35

สีผิว ส่วนสูง
x_1 x_2
สมมุติ ประมวลผล
$> 0 =$ ช
$< 0 =$ ญ

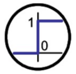


รูปที่ 6–35 เปอร์เซปตรอน

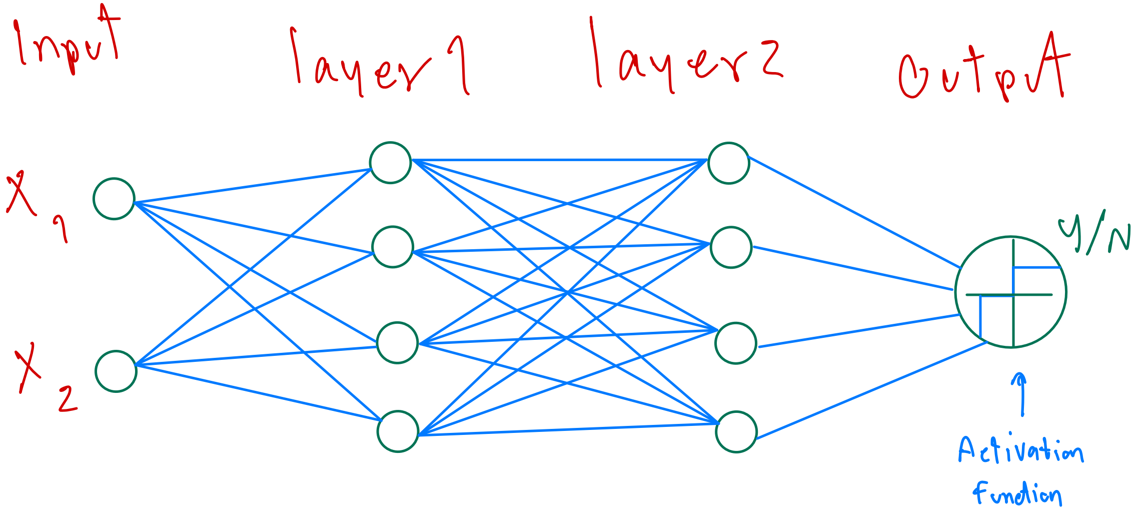
ฟังก์ชันกระตุ้น



x = input

ในรูปแสดง*ฟังก์ชันกระตุ้น (activation function)* ชนิดที่เรียกว่า*ฟังก์ชันสองขั้ว (bipolar function)* ซึ่งแสดงผลของเอาต์พุตเป็น 1 กับ -1 ฟังก์ชันกระตุ้นอื่นๆ ที่นิยมใช้ก็อย่างเช่น *ฟังก์ชันไบนารี (binary function)* ซึ่งแสดงผลของเอาต์พุตเป็น 1 กับ 0 และเขียนแทนด้วยรูป 

ถ้า x ผลรวม > 1 ผลลัพธ์จะเป็น 1
ถ้า x ผลรวม < 1 ——— 0

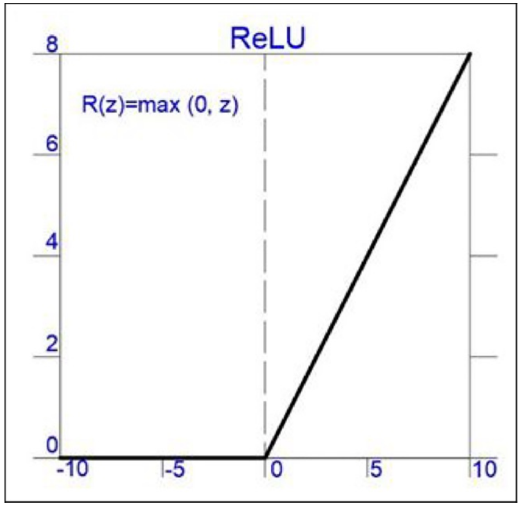


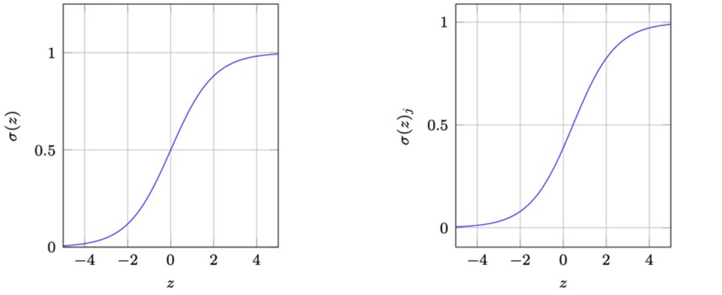
Fully Connected
Dense layer

Activation function

RElu

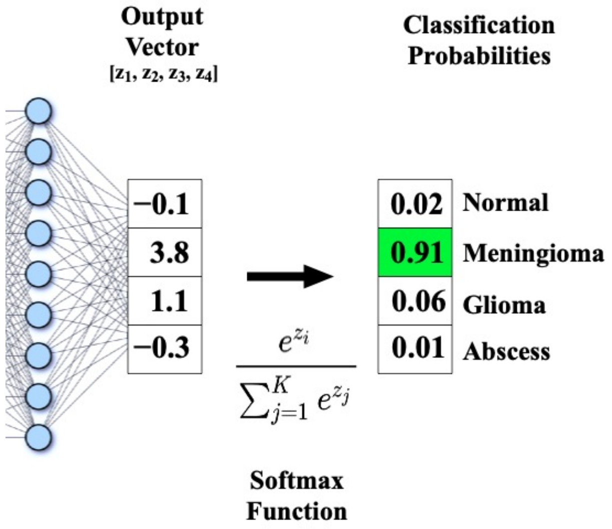
Sigmoid function $\xrightarrow{\text{คล้ายๆกับ}}$ softmax





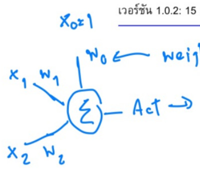
(a) Sigmoid activation function. (b) Softmax activation function.

Figure 1: Sigmoid and Softmax activation functions



$$\frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$$

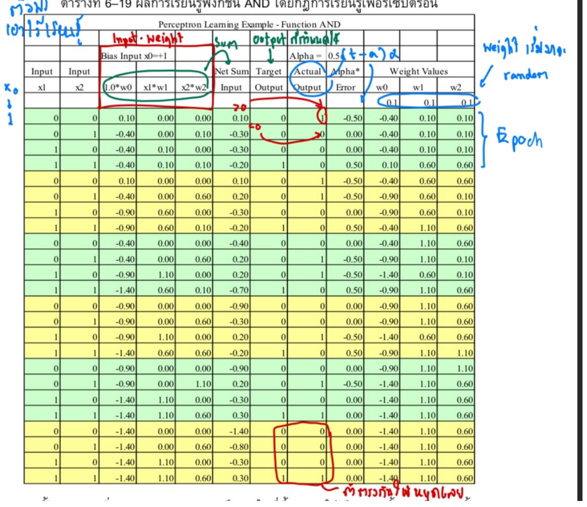
$$f(\varepsilon)\begin{cases} 1 ; & \varepsilon \geq 0 \\ 0 ; & \varepsilon < 0 \end{cases}$$



weight ตัวปร เอาไว้เรียนรู้

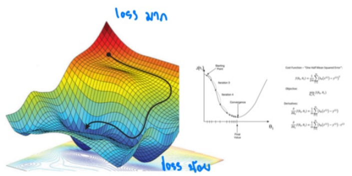
เวอร์ชัน 1.0.2: 15 มีค.2548 : 9:05 PM boonserm.k@chula.ac.th 6 การเรียนรู้ของเครื่อง 175

ตารางที่ 6–19 ผลการเรียนรู้ฟังก์ชัน AND โดยกฎการเรียนรู้เพอร์เซปตรอน


Input · weight
sum
output ที่กำหนดไว้
(t − a) α
x0
1
>0
<0
weight เริ่มจาก random
Epoch
ถ้าตรงกันให้หยุดเลย

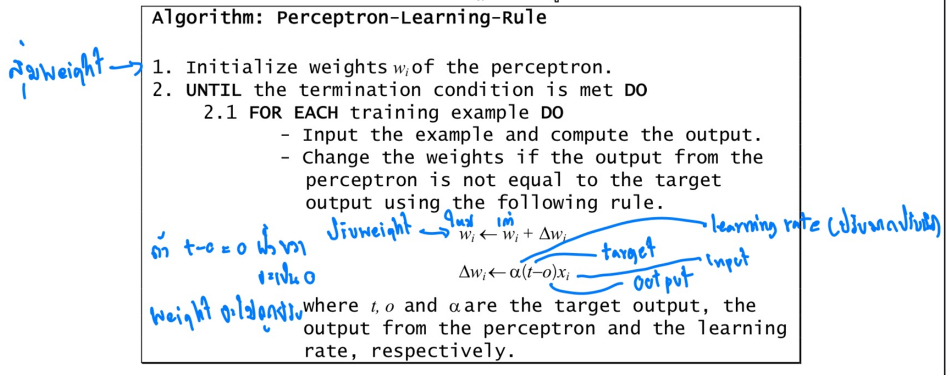
Perceptron Learning Example - Function AND											
		Bias Input x0=+1					Alpha =	0.5			
Input	Input				Net Sum	Target	Actual	Alpha*	Weight Values		
x1	x2	1.0*w0	x1*w1	x2*w2	Input	Output	Output	Error	w0	w1	w2
									0.1	0.1	0.1
0	0	0.10	0.00	0.00	0.10	0	1	-0.50	-0.40	0.10	0.10
0	1	-0.40	0.00	0.10	-0.30	0	0	0.00	-0.40	0.10	0.10
1	0	-0.40	0.10	0.00	-0.30	0	0	0.00	-0.40	0.10	0.10
1	1	-0.40	0.10	0.10	-0.20	1	0	0.50	0.10	0.60	0.60
0	0	0.10	0.00	0.00	0.10	0	1	-0.50	-0.40	0.60	0.60
0	1	-0.40	0.00	0.60	0.20	0	1	-0.50	-0.90	0.60	0.10
1	0	-0.90	0.60	0.00	-0.30	0	0	0.00	-0.90	0.60	0.10
1	1	-0.90	0.60	0.10	-0.20	1	0	0.50	-0.40	1.10	0.60
0	0	-0.40	0.00	0.00	-0.40	0	0	0.00	-0.40	1.10	0.60
0	1	-0.40	0.00	0.60	0.20	0	1	-0.50	-0.90	1.10	0.10
1	0	-0.90	1.10	0.00	0.20	0	1	-0.50	-1.40	0.60	0.10
1	1	-1.40	0.60	0.10	-0.70	1	0	0.50	-0.90	1.10	0.60
0	0	-0.90	0.00	0.00	-0.90	0	0	0.00	-0.90	1.10	0.60
0	1	-0.90	0.00	0.60	-0.30	0	0	0.00	-0.90	1.10	0.60
1	0	-0.90	1.10	0.00	0.20	0	1	-0.50	-1.40	0.60	0.60
1	1	-1.40	0.60	0.60	-0.20	1	0	0.50	-0.90	1.10	1.10
0	0	-0.90	0.00	0.00	-0.90	0	0	0.00	-0.90	1.10	1.10
0	1	-0.90	0.00	1.10	0.20	0	1	-0.50	-1.40	1.10	0.60
1	0	-1.40	1.10	0.00	-0.30	0	0	0.00	-1.40	1.10	0.60
1	1	-1.40	1.10	0.60	0.30	1	1	0.00	-1.40	1.10	0.60
0	0	-1.40	0.00	0.00	-1.40	0	0	0.00	-1.40	1.10	0.60
0	1	-1.40	0.00	0.60	-0.80	0	0	0.00	-1.40	1.10	0.60
1	0	-1.40	1.10	0.00	-0.30	0	0	0.00	-1.40	1.10	0.60
1	1	-1.40	1.10	0.60	0.30	1	1	0.00	-1.40	1.10	0.60

$$\text{Loss} = -\frac{1}{\text{output size}} \sum_{i=1}^{\text{output size}} \overset{\text{คำตอบ}}{y_i} \cdot \log \overset{\text{ของจริง}}{\hat{y}_i} + (1 - y_i) \cdot \log\,(1 - \hat{y}_i)$$



เวอร์ชัน 1.0.2: 15 มีค.2548 : 9:05 PM boonserm.k@chula.ac.th 6 การเรียนรู้ของเครื่อง 173

ตารางที่ 6–17 อัลกอริทึมกฎการเรียนรู้เพอร์เซปตรอน

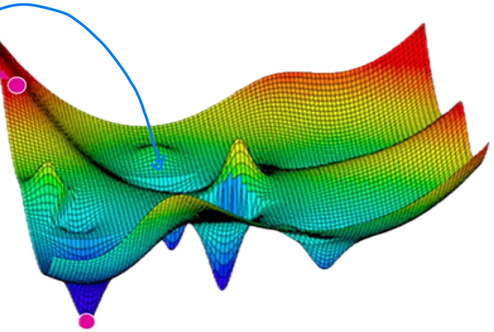


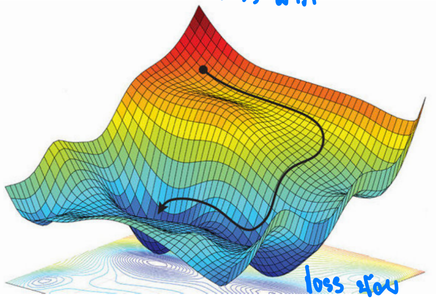
```
Algorithm: Perceptron-Learning-Rule

1. Initialize weights $w_i$ of the perceptron.
2. UNTIL the termination condition is met DO
    2.1 FOR EACH training example DO
        - Input the example and compute the output.
        - Change the weights if the output from the
          perceptron is not equal to the target
          output using the following rule.
                $w_i \leftarrow w_i + \Delta w_i$
                $\Delta w_i \leftarrow \alpha(t-o)x_i$
          where $t$, $o$ and $\alpha$ are the target output, the
          output from the perceptron and the learning
          rate, respectively.
```

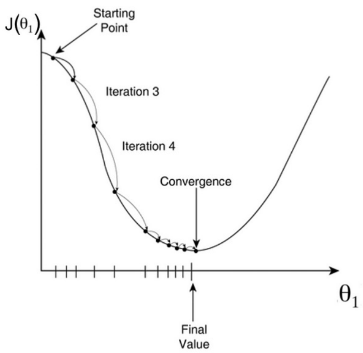
Loss function

$$\text{Loss} = -\frac{1}{\substack{\text{output}\\ \text{size}}} \sum_{i=1}^{\substack{\text{output}\\ \text{size}}} \overset{\text{คำตอบ}}{\overset{\downarrow}{y_i}} \cdot \log \overset{\text{ของจริง}}{\overset{\downarrow}{\hat{y}_i}} + (1 - y_i) \cdot \log\,(1 - \hat{y}_i)$$

learning rate
ต่ำ จะทำให้เจอ
หลุมหลอก







Cost Function – "One Half Mean Squared Error":

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

Objective:

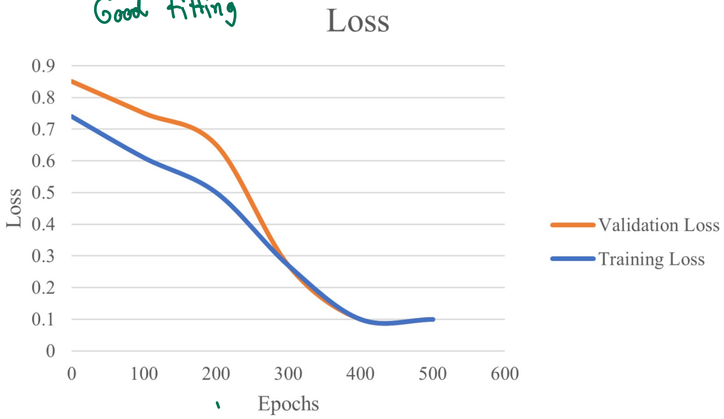
$$\min_{\theta_0, \theta_1} J(\theta_0, \theta_1)$$

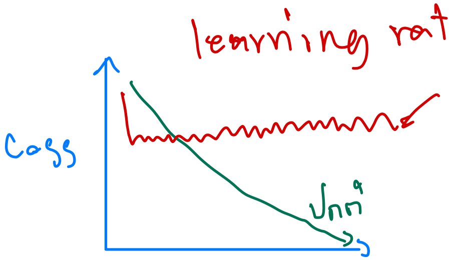
Derivatives:

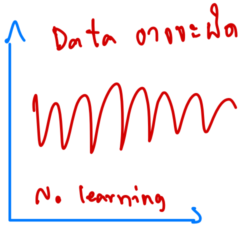
$$\frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1) = \frac{1}{m}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)$$

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

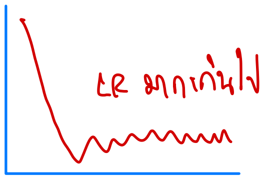
Good fitting

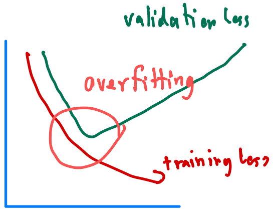




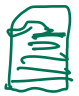


Epoch

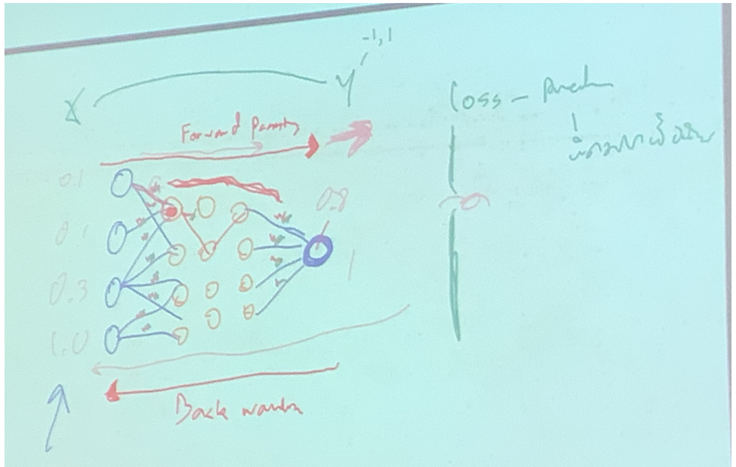




Batch



การทรบด้วย Batch
ต่อการผยหาม จะให้ตอบ
ที่เด้ยอมรับกันหมดเลย

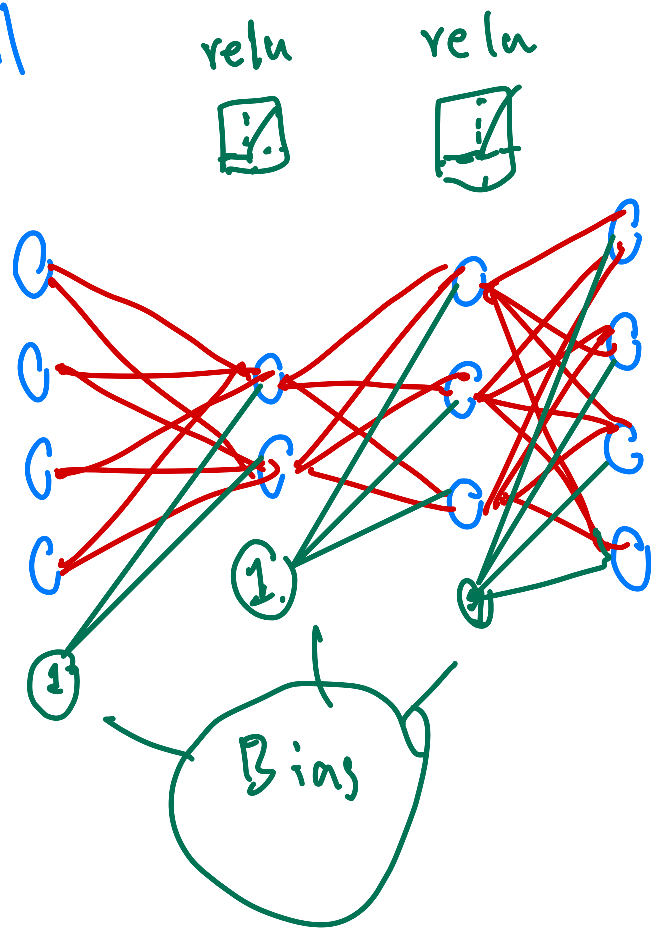


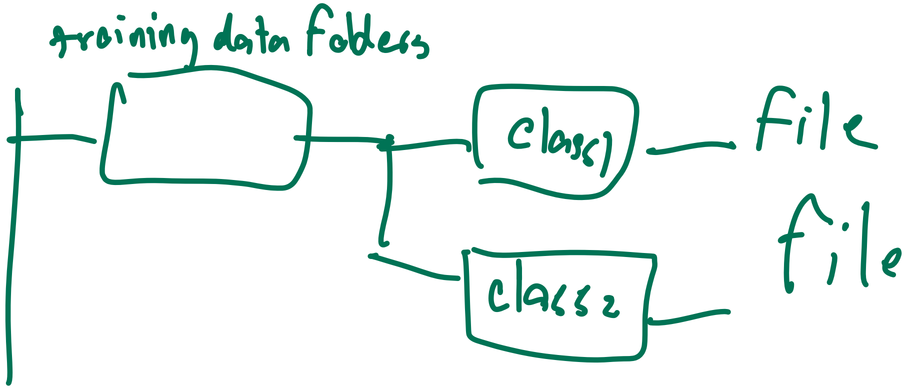
Deep learning

Neural Network

مكتبة Tensor flow

sequential


relu
relu
1
1
Bias


training data folders
class1
class2
file
file