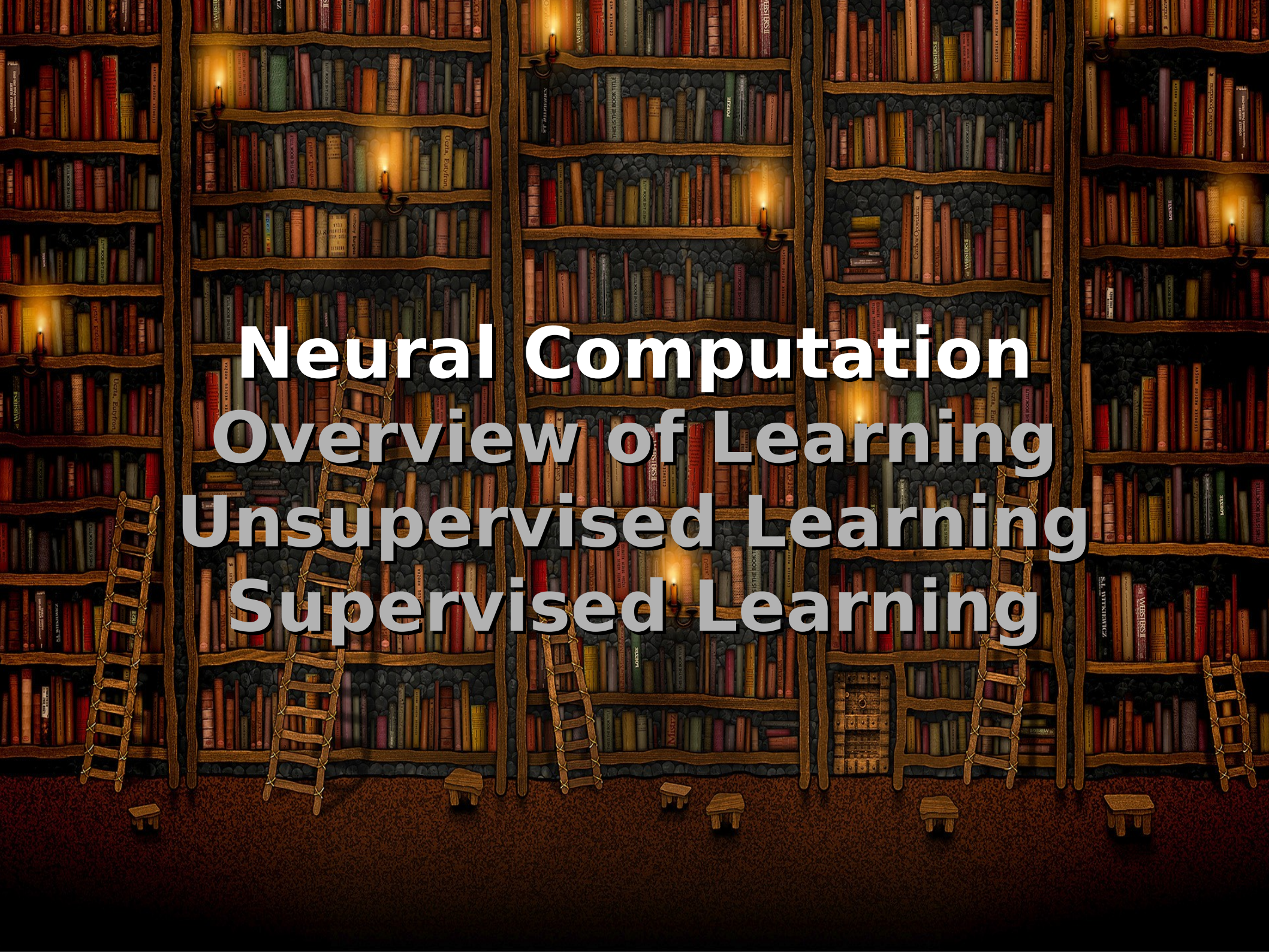
神經與行為模型建構
(Neural & Behavioral Modeling)

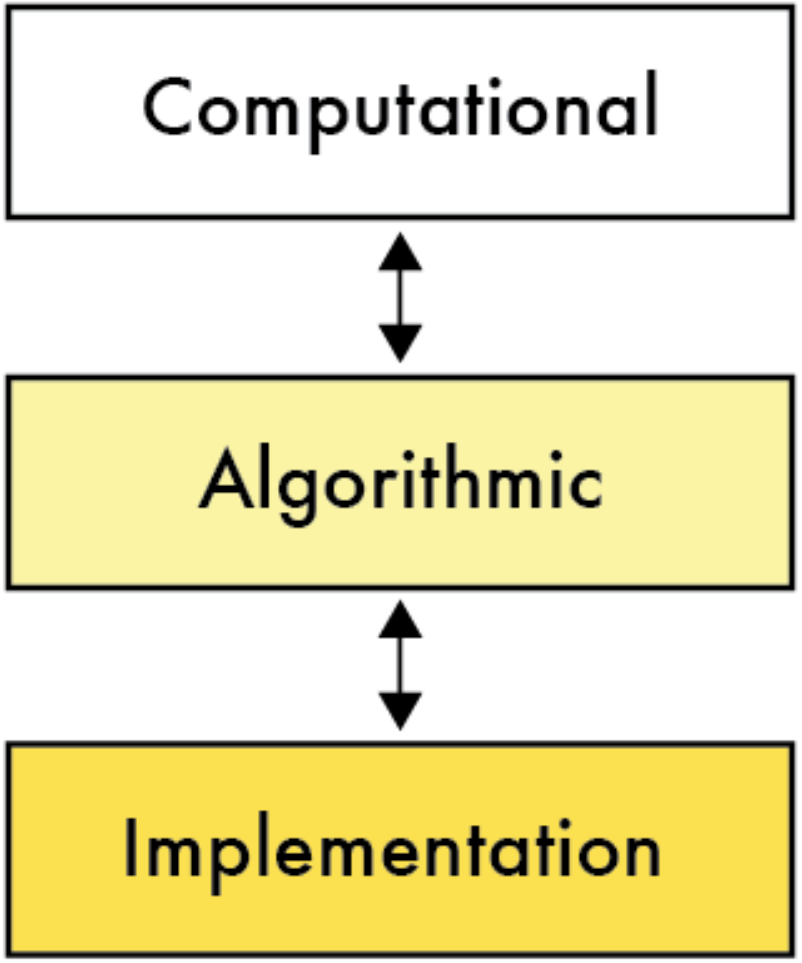
課號：Psy7277　　教室： 北 206
識別碼：227M9280　　時間： 五 234


學習的實驗 / 模型較複雜
!


Neural Computation
Overview of Learning
Unsupervised Learning
Supervised Learning

了解一個認知系統的特性與極限

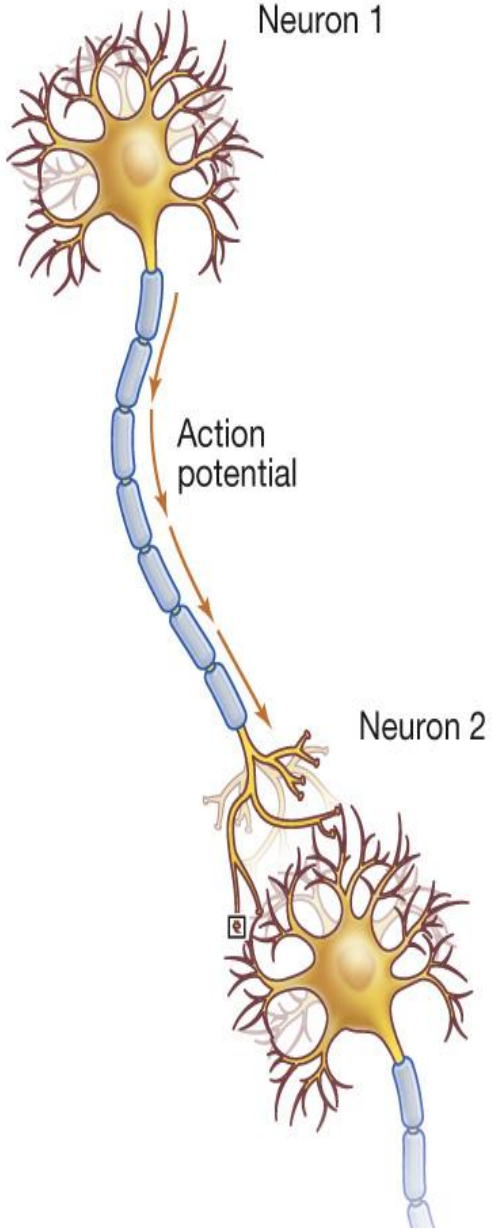
David Marr's 3 levels of analysis

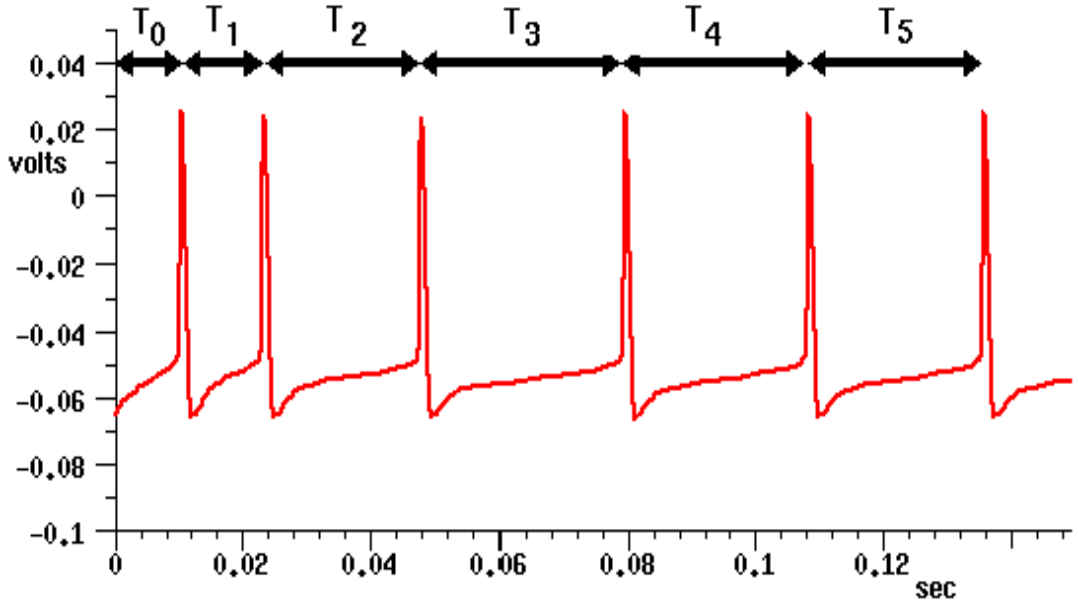


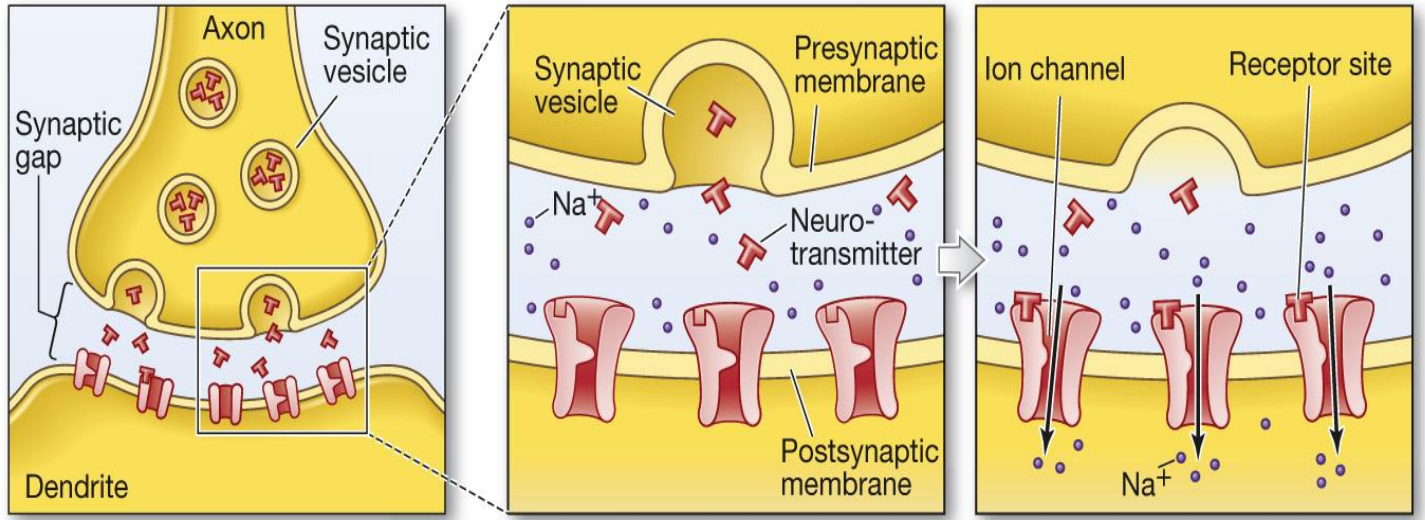
- 計算上的問題是什麼？
  x 是什麼, y 是什麼, 找質因素
- 能夠用什麼演算法來解決？
  暴力解、XX演算法
- 要透過什麼硬體來實現算法？
  手算、算盤、電腦、量子電腦?!

Implementation: 有 0 與 1 的神經元

刺激越強，firing rate 變強 (單位時間內的次數變多)
但是 spike (action potential) 還是一樣的 (0 or 1)

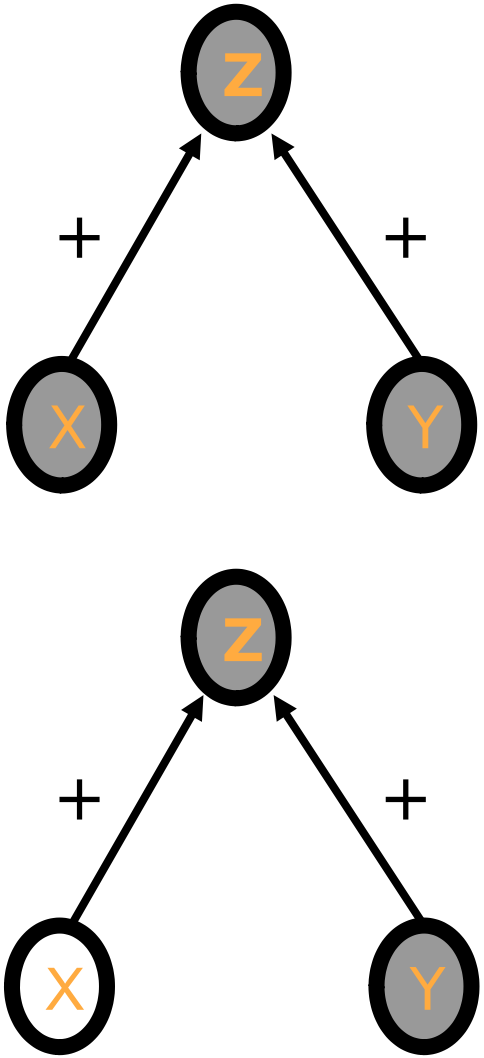






Implementation: 神經元做加法

如果接受刺激的神經元有個低閾值

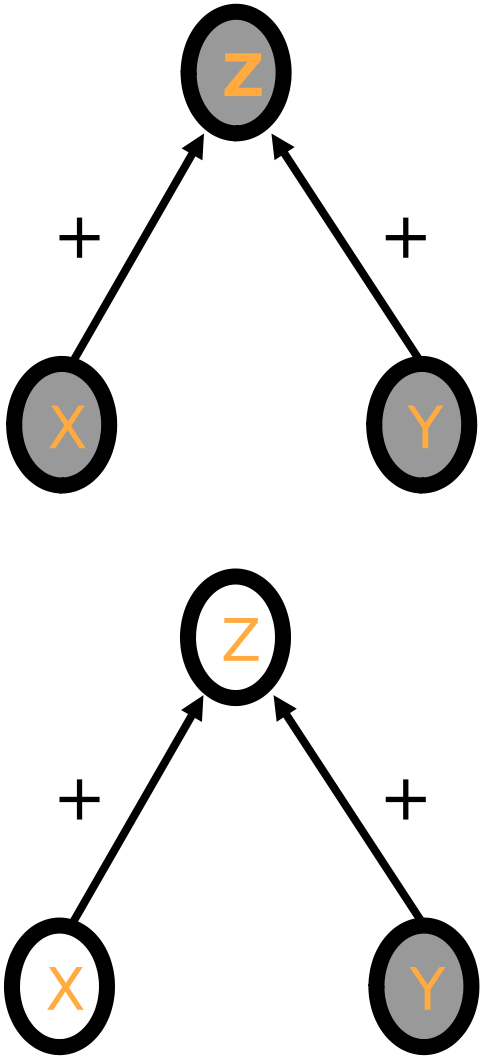


X	Y	Z
0	0	0
0	1	1
1	0	1
1	1	1

Z=X+Y (OR)

Implementation: 神經元做乘法

如果接受刺激的神經元有個高閾值

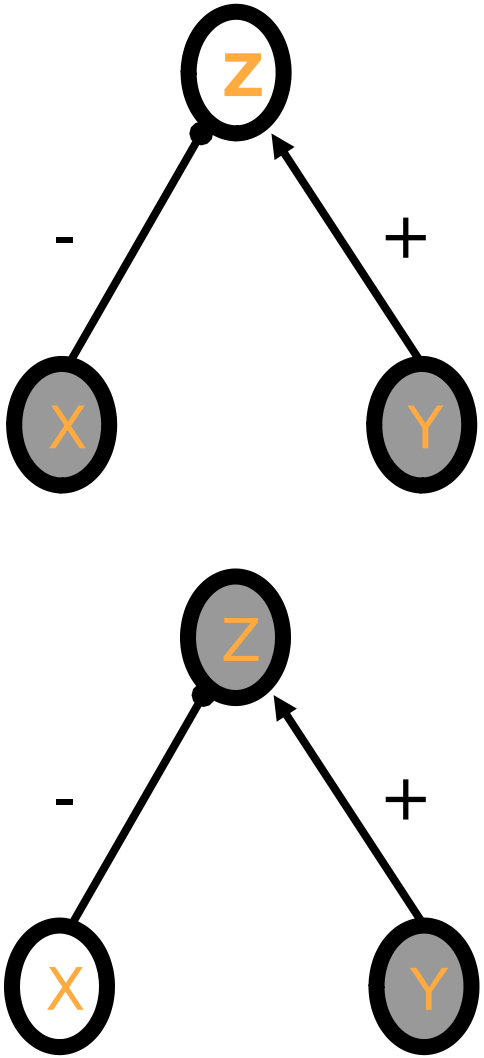


X	Y	Z
0	0	0
0	1	0
1	0	0
1	1	1

Z=X*Y (AND)

Implementation: 神經元做減法

如果開始考慮神經元彼此的抑制關係

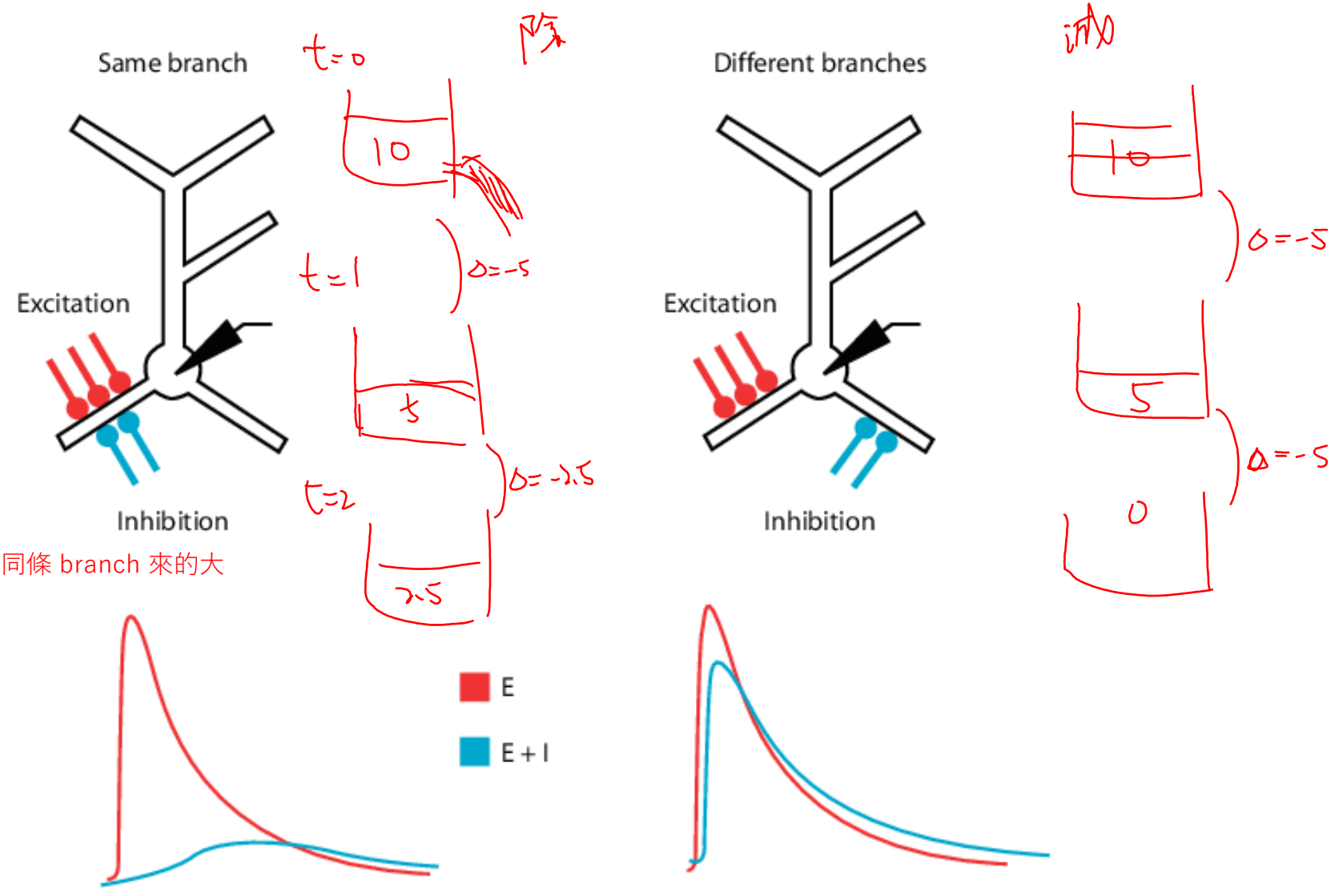


X	Z
0	1
1	0

Z=1-X (NOT)

Implementation: 神經元做除法

神經元間的抑制可以導致減法或除法



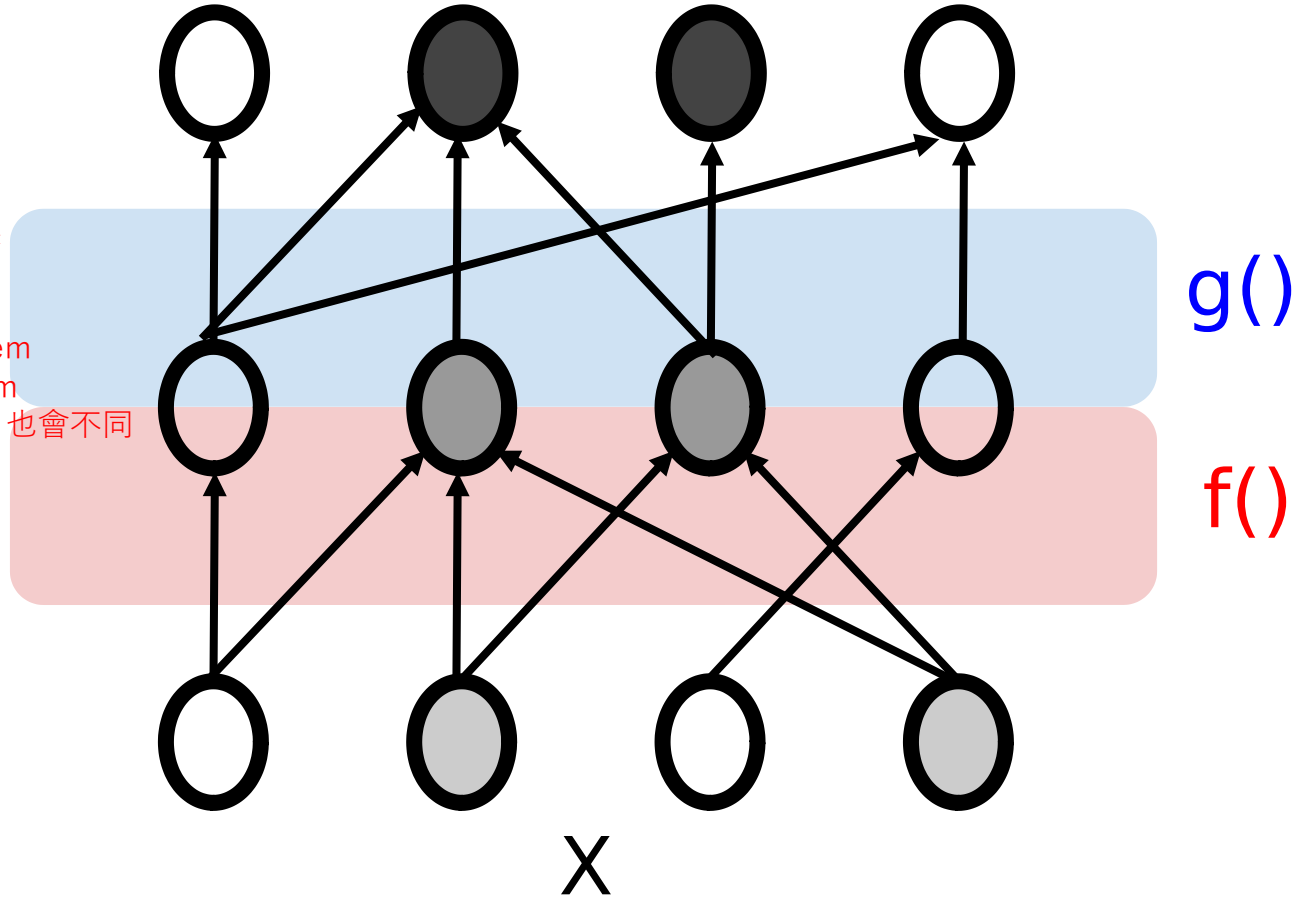
同條 branch 上
拉下來的效果比不同條 branch 來的大

Algorithm: 基本計算的組合

一個神經網路透過組合基本的計算來實現演算法

如果 f() g() 都是 linear function, 那其實 g.f() 也是 linea function

$$Y=g(f(X))$$



internal representation
透過學習來知曉這些連結的關係
有些 weighted 高，有些低

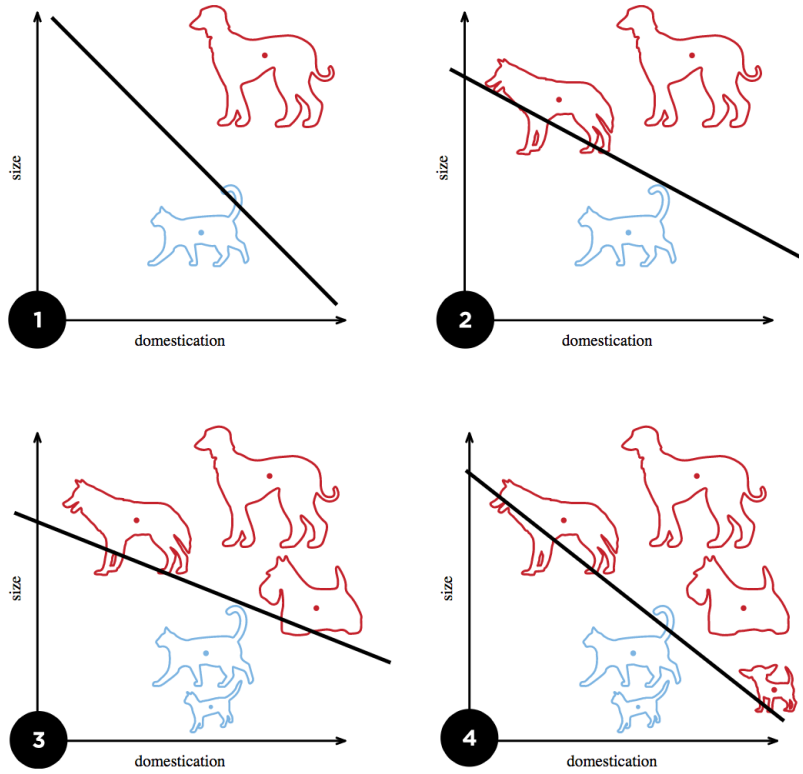
在同樣的 computational problem
不同的 computational algorithm
找到的 internal representation 也會不同

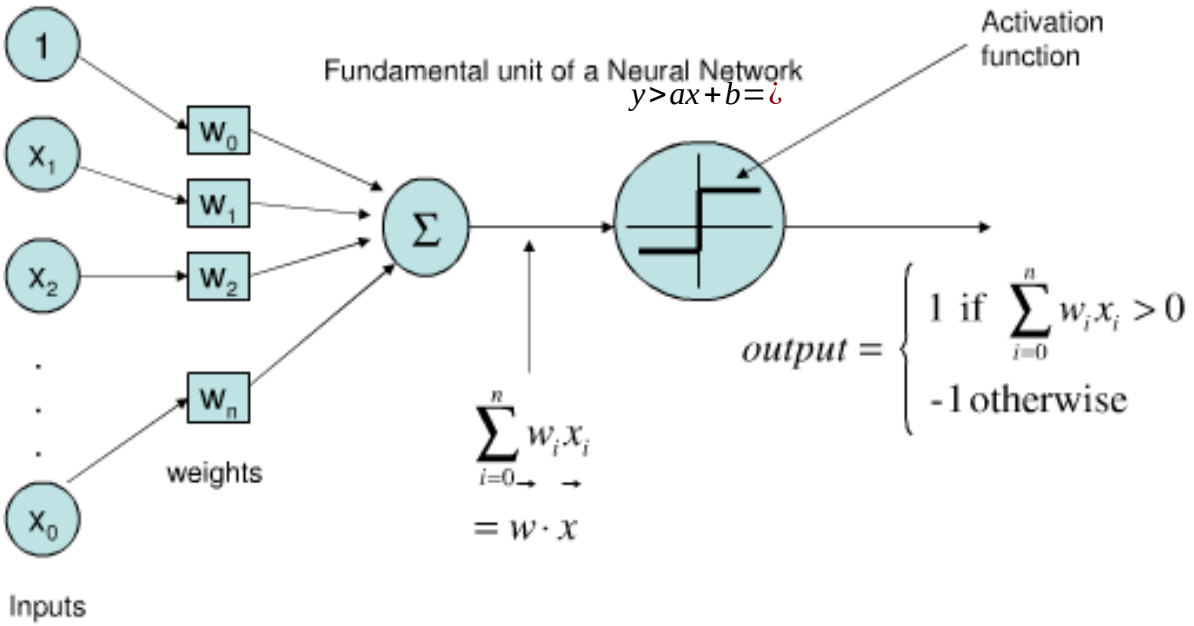
Computational Problem: 區分貓狗

Perceptron (Rosenblatt, 1958)
Cognitron (Fukushima, 1975)

$$y > ax + b \Rightarrow y - ax - b > 0 \Rightarrow w_2 x_2 + w_1 x_1 + w_0 1 > 0$$

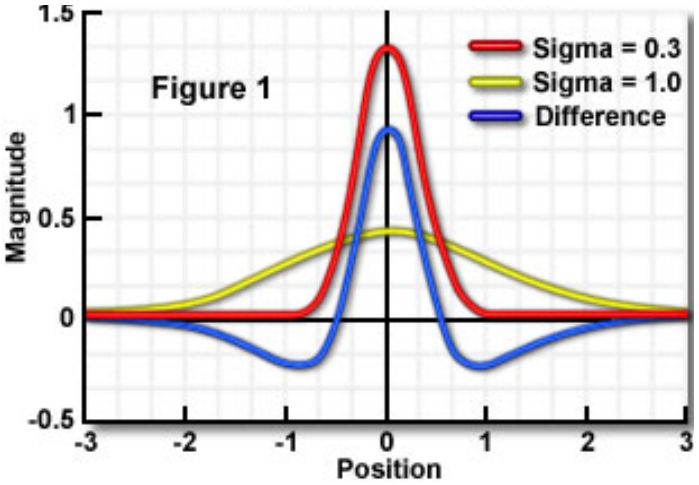
pattern recogniter
區分貓狗可以視為區分是不是狗

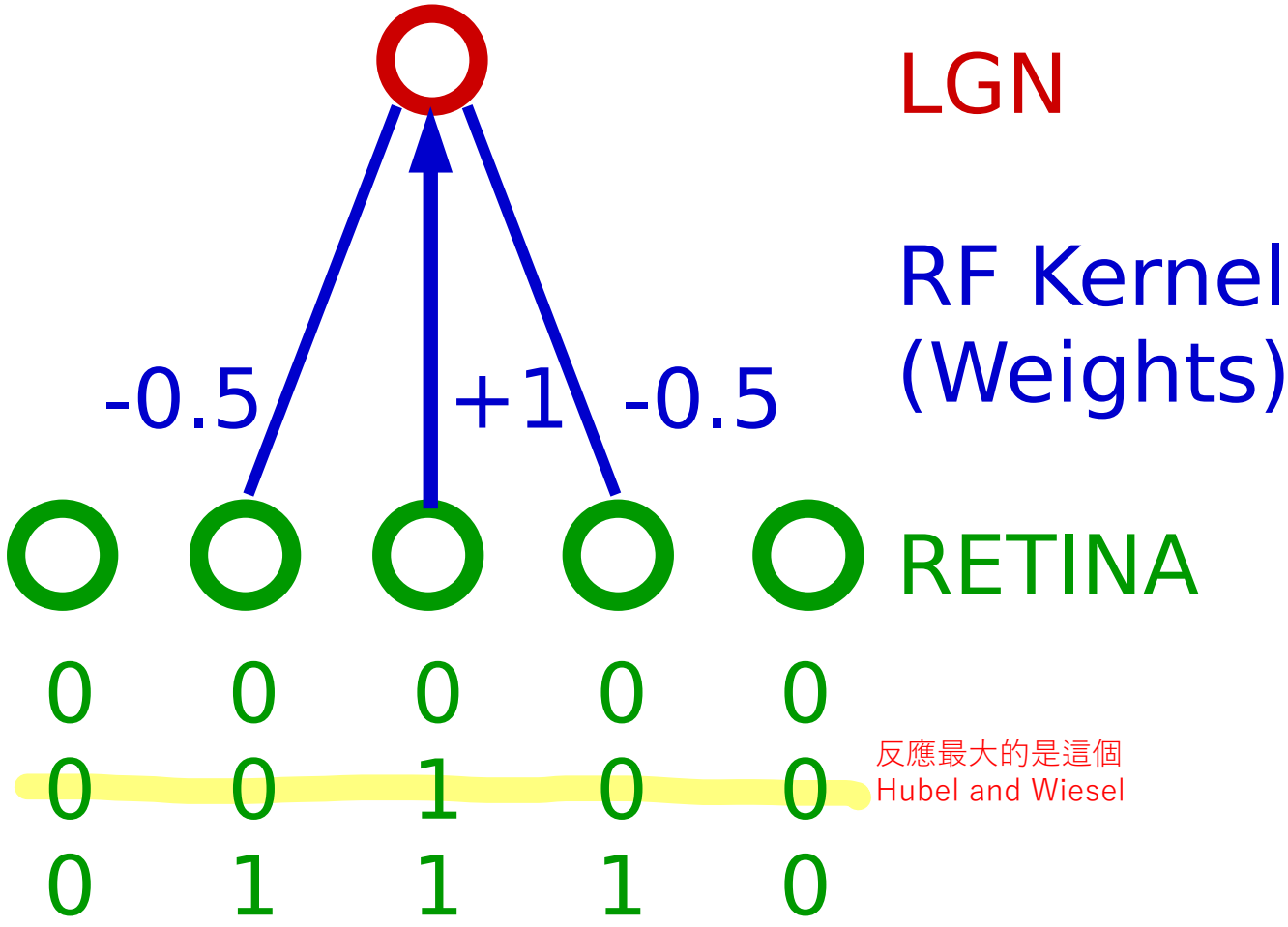




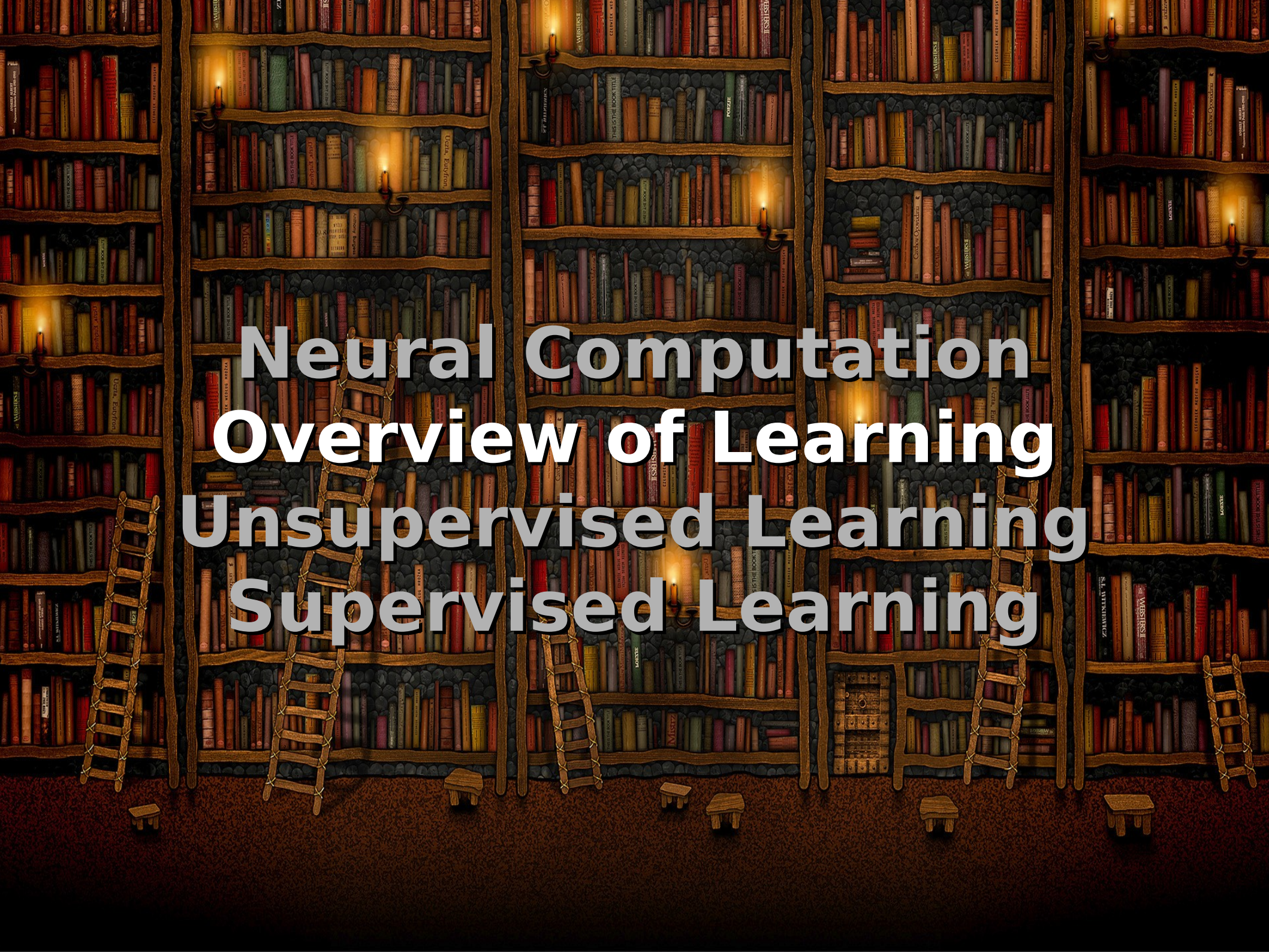
$$y < ax + b \Rightarrow y - a - b < 0 \Rightarrow w_2 x_2 + w_1 x_1 + w_0 1 < 0$$

Neuron as a Pattern Recognizer



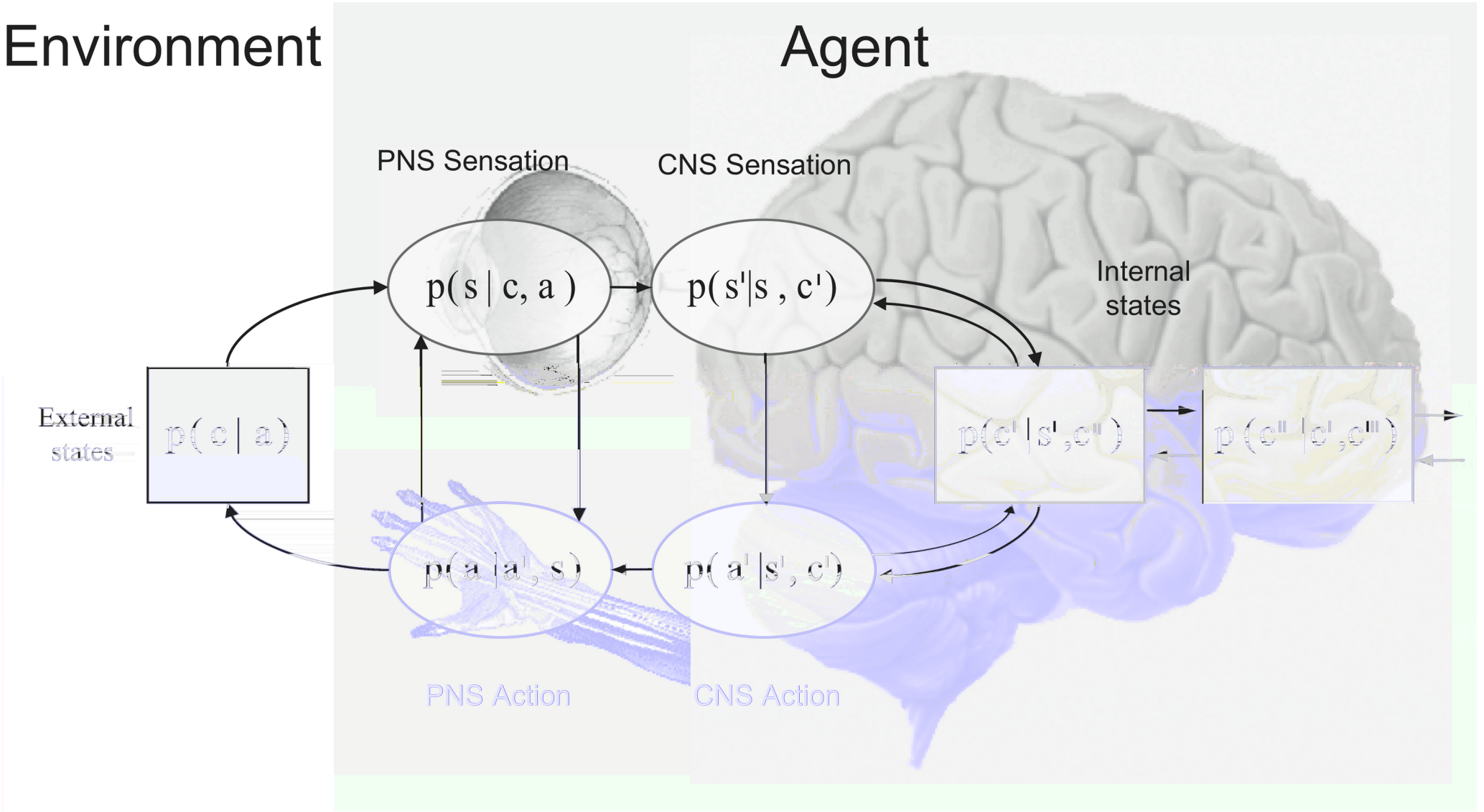



```
RETINA=matrix('0 0 0 0 0;0 0 1 0 0;0 1 1 1 0')
LGN_RF=matrix('0;-0.5;1;-0.5;0')
LGN_RESPONSE=RETINA*LGN_RF
```


Neural Computation
Overview of Learning
Unsupervised Learning
Supervised Learning

學習 / 記憶的目的：預測與控制

學習和演化的目的都是更適應環境



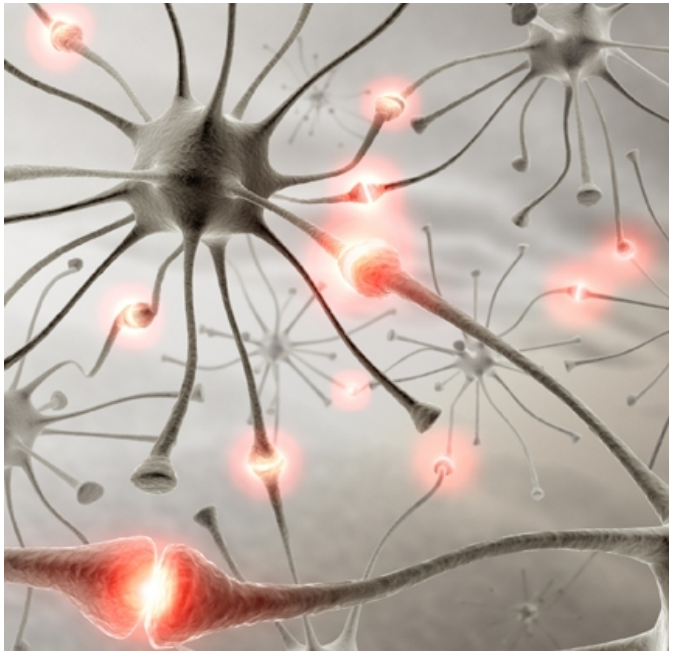
學習中的兩難 (Dilemma)

Learning vs. Performance

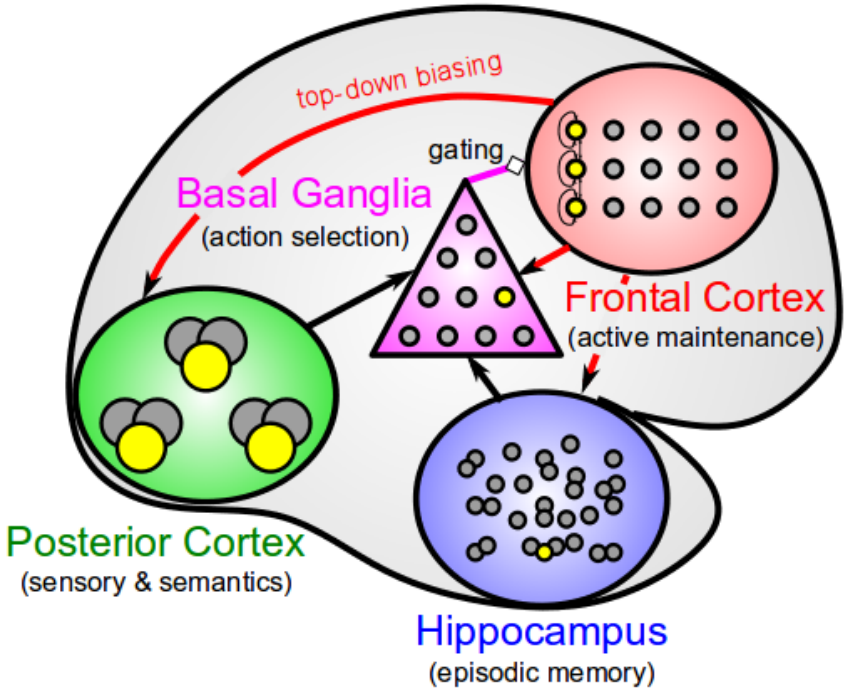
行為上，正在學習一件事時表現通常不好，表現好時表示其實已相當熟練沒學到什麼。

Stability vs. Plasticity

大腦中，神經結構的變化雖可幫助學習新事物，但這些結構上的改變也意味著對舊事物的遺忘。



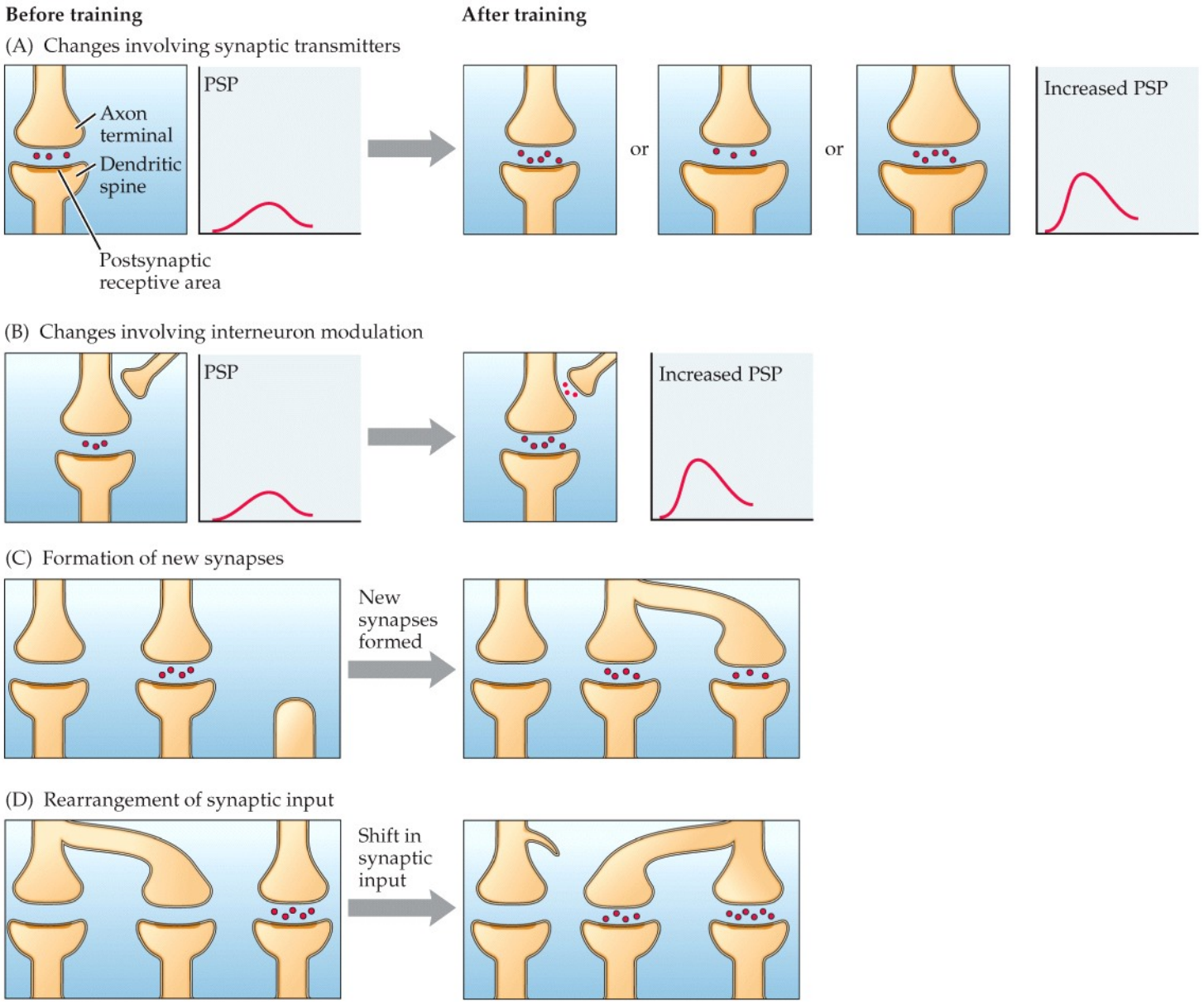
不同腦區的計算 / 學習性質



	Learning Signal			*Dynamics*		
Area	**Reward**	**Error**	**Self Org**	**Separator**	**Integrator**	**Attractor**
Basal Ganglia	+++	---	---	++	-	---
Cerebellum	---	+++	---	+++	---	---
Hippocampus	+	+	+++	+++	---	+++
Neocortex	++	+++	++	---	+++	+++

突觸連結變強的生物機制

雖有不同機制但數學上都是 ΔW_{ij} 變大



人與機器的歸納學習

皆有以下三種情境

非監督式學習：事物沒有標籤（對錯或名字）
小孩發現有些生物會動，有些不會動

監督式學習：事物有明確標籤或該怎麼做
爸媽教小孩會動的叫動物，不會動的叫植物

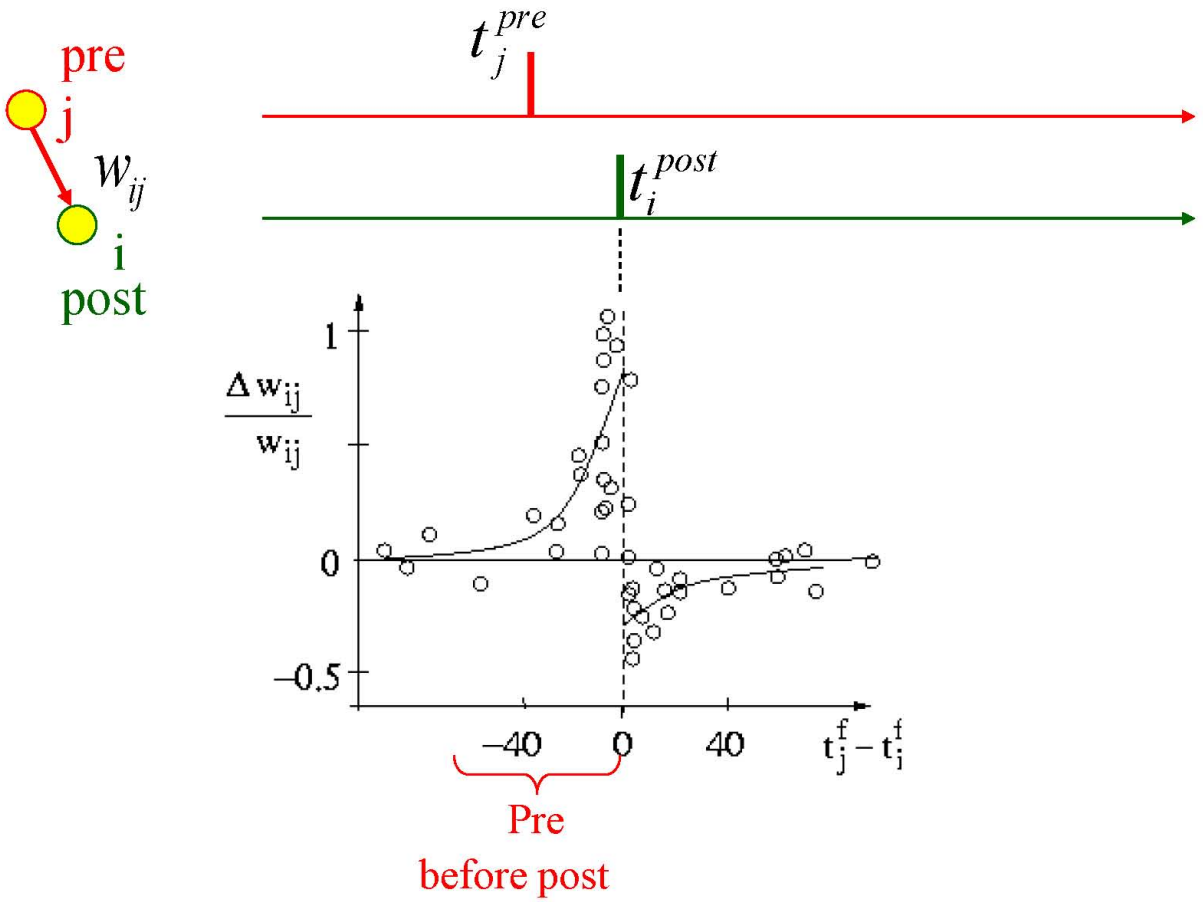
強化式學習：只知道特定事件後會有賞罰
小孩發現打動物會被咬，打植物卻沒事


Neural Computation
Overview of Learning
Unsupervised Learning
Supervised Learning

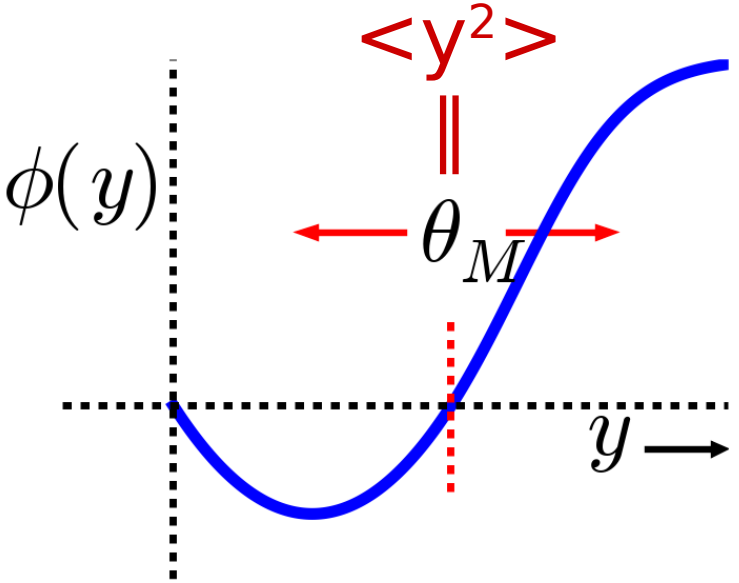
Hebbian vs. Anti-Hebbian

$$\Delta w_{ij} = x_i y_j \quad \text{vs.} \quad \Delta w_{ij} = -x_i y_j$$

生物上發現兩者並存：STDP（左）& BCM rule（右）

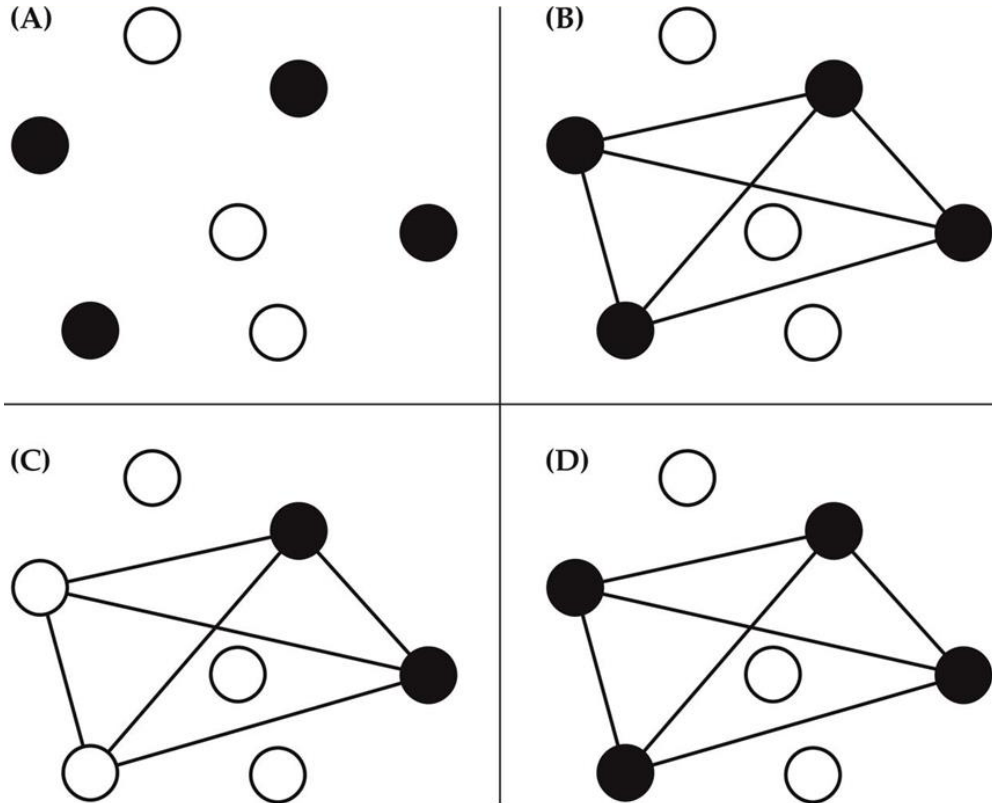


$$\Delta w_{ij} = x_i \Phi(y_j)$$
$$= x_i [y_j (y_j - \theta)]$$



Hopfield Network

類 hippocampus 的 auto-associative memory



Memorize:

$$w_{ij} = \frac{1}{n}\sum_{p=1}^{n} \Delta w_{ij} = \frac{1}{n}\sum_{p=1}^{n} x_i^p x_j^p$$

Cued Recall:

$$x_i = \begin{cases} +1, \sum_j w_{ij} x_j \geq \theta_i \\ -1, otherwise \end{cases}$$

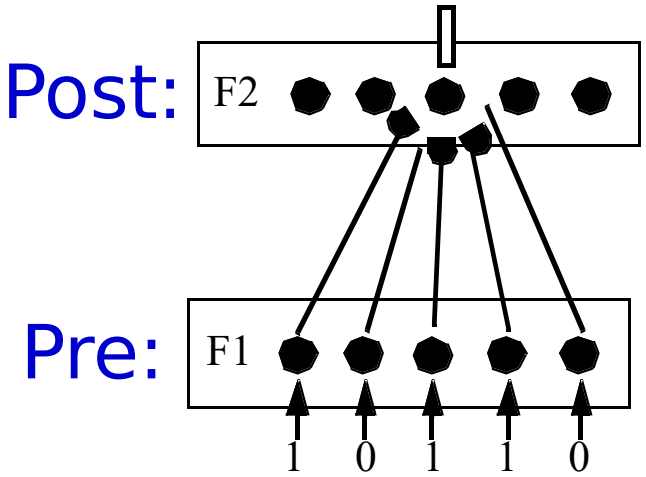
本質上是用大家來投票來實行

constraint satisfaction / convergent thinking

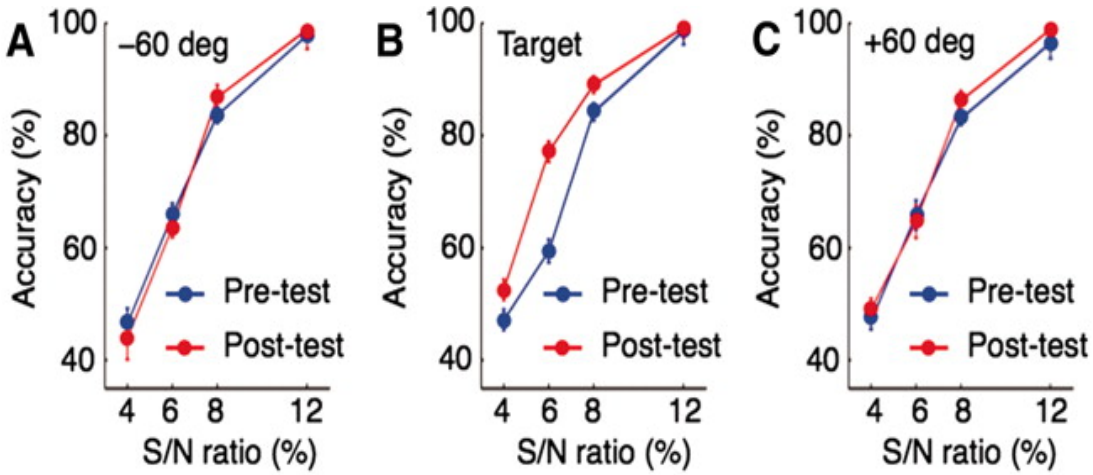
Boltzmann Machine是 Hopfield Net 的隨機版

InStar Learning

InStar: 突觸後神經元學習辨認突觸前的樣式



$$\Delta \vec{W}_i^{1\to 2} = y_i\left(\vec{x} - \vec{w}_i^{1\to 2}\right)$$



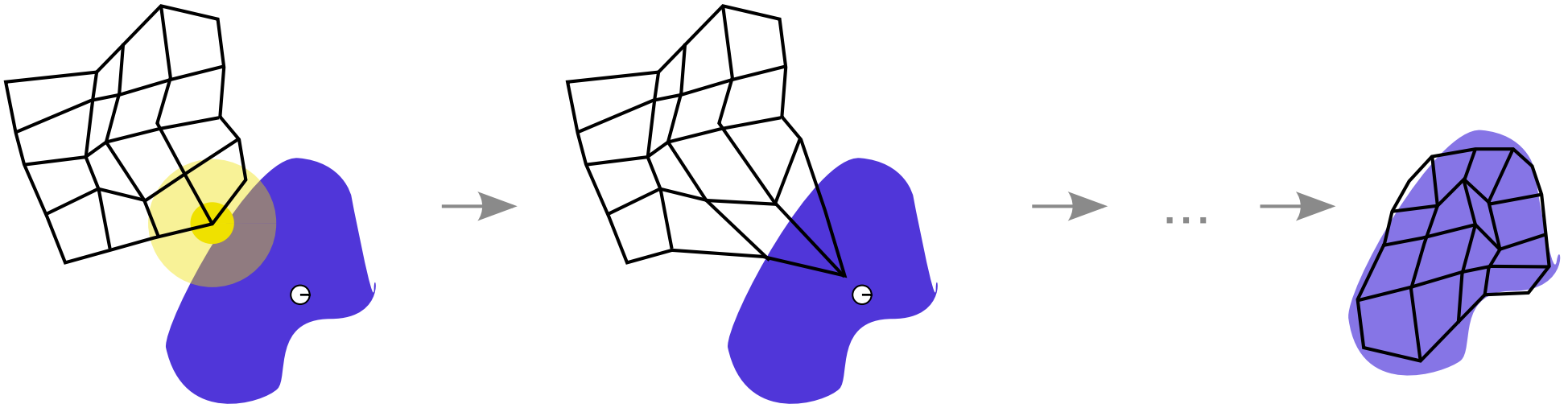

```
x=array([0,1,0]) # repeated stimulus
W=random.rand(3) # random initial weights
for i in range(10): # 10 trials
    y=dot(W,x) # the only postsynaptic neuron
    W=W+y*(x-W) # postsynaptically gated InStar
    print(W,y)
```

Self-Organizing Map

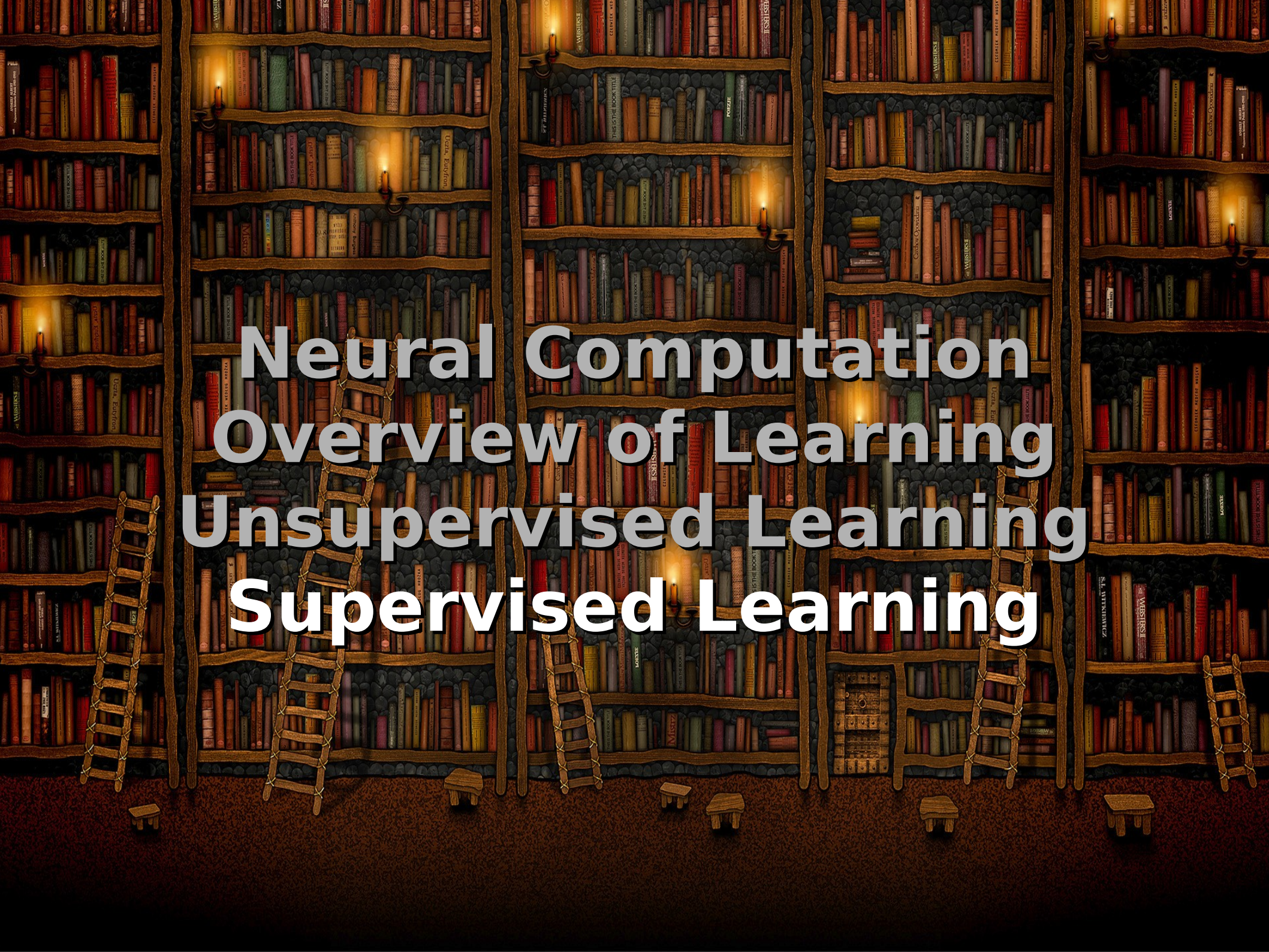
只是 InStar 的變形：每次學習的是一個贏者 y_i 的近鄰

$$\Delta \vec{w}_i = N(y_j, y_i)(\vec{x} - \vec{w}_i)$$

N 是某種 neighborhood function（如常態分佈）



學辨認刺激但保留神經元間的拓撲關係 (V1/A1/SMC)

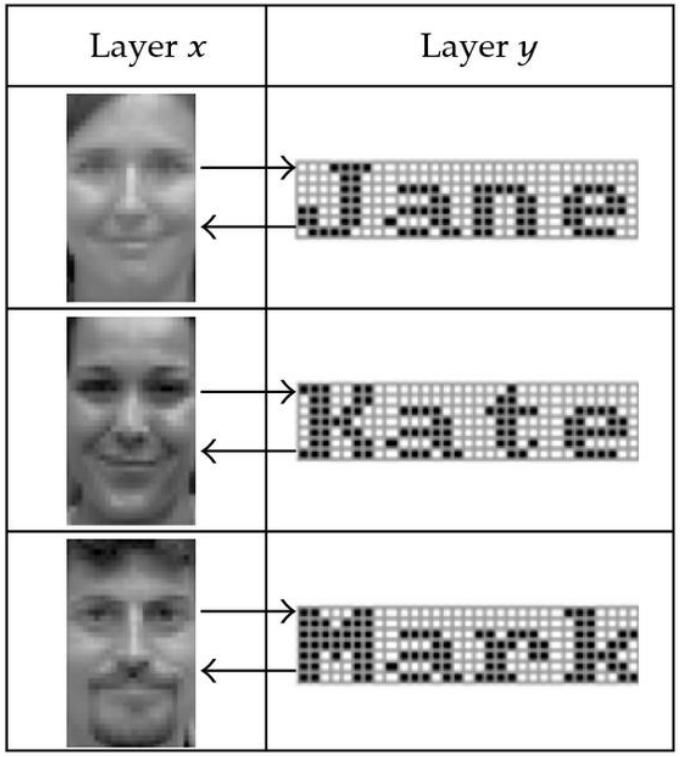

Neural Computation
Overview of Learning
Unsupervised Learning
Supervised Learning

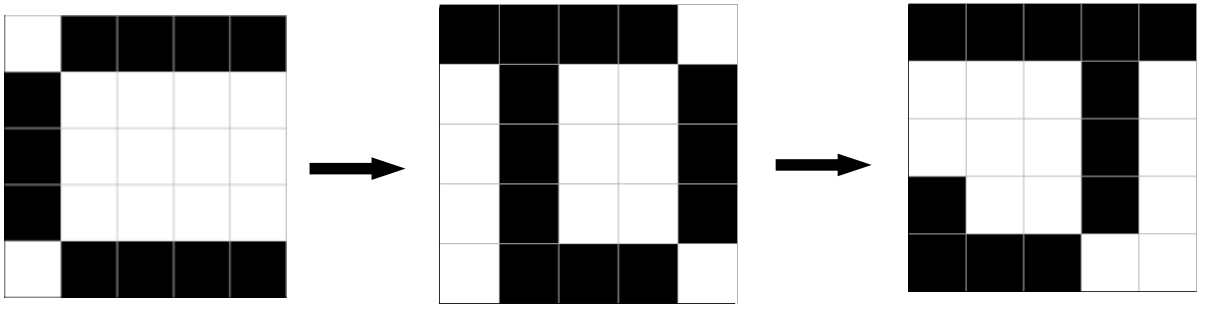
Supervised Hebbian Learning

輸入層的 x_i 和輸出層的 y_j 都 clamp 住來改變 w_{ij}

$$\Delta w_{ij} = x_i y_j \qquad w_{ij} = \frac{1}{n}\sum_{p=1}^{n} \Delta w_{ij} = \sum_{p=1}^{n} x_i^p y_j^p$$

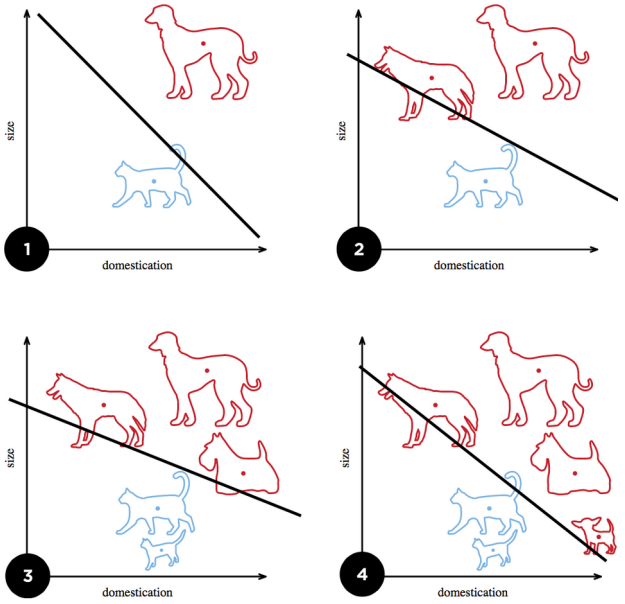
Hopfield 網路可改為 Hetero-associative Memory
來連結不同本質 / 維度的 x 與 y 或產生 x 的序列

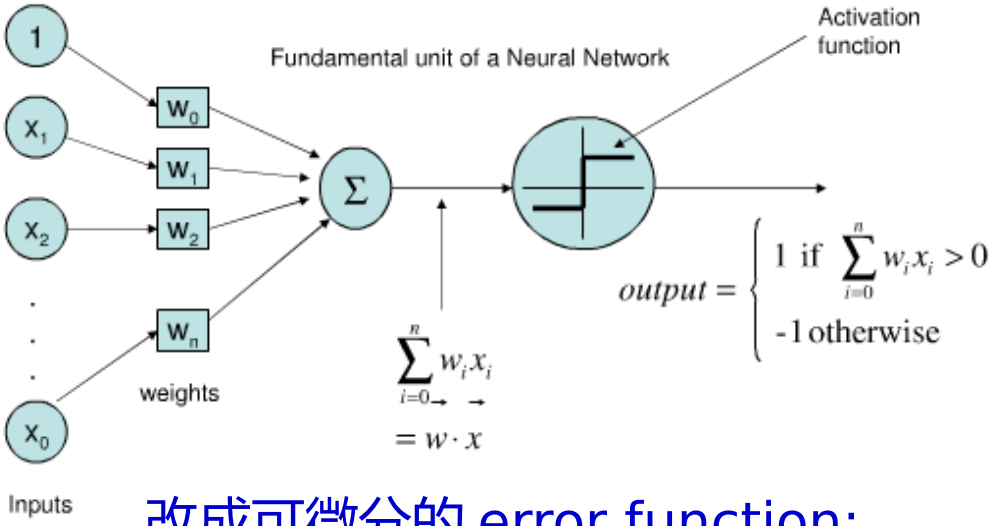




Perceptron: 分類

學習規則通常由 gradient descent 推導而來

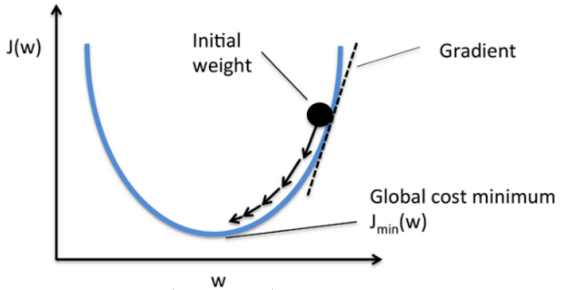




改成可微分的 error function:
相減變相乘

$$E=\left(Y^p-sgn\left(WX^p\right)\right)^2\approx-\left(WX^p\right)Y^p$$

但階梯函數 sgn 不可微分



這不就是 Supervised Hebbian 嗎 !?

$$W^{new}=W^{old}+\Delta W=W^{old}-\partial E\left(W\right)/\partial W=W^{old}+X^pY^p$$

Delta Rule：迴歸

x_i 驅使 y_j 產生的錯誤 $\delta_j=(t_j-y_j)$ 要透過修改 w_{ij} 來減小

$$y_j=\sum_i w_{ij}x_i$$

$$\Delta w_{ij}=x_i\delta_j=x_i(t_j-y_j)$$

上式只適用兩層且線性的網路

對任意的激發函數 f 則可由最小平方誤差法推得：

$$y_j=f\left(\sum_i w_{ij}x_i\right)$$

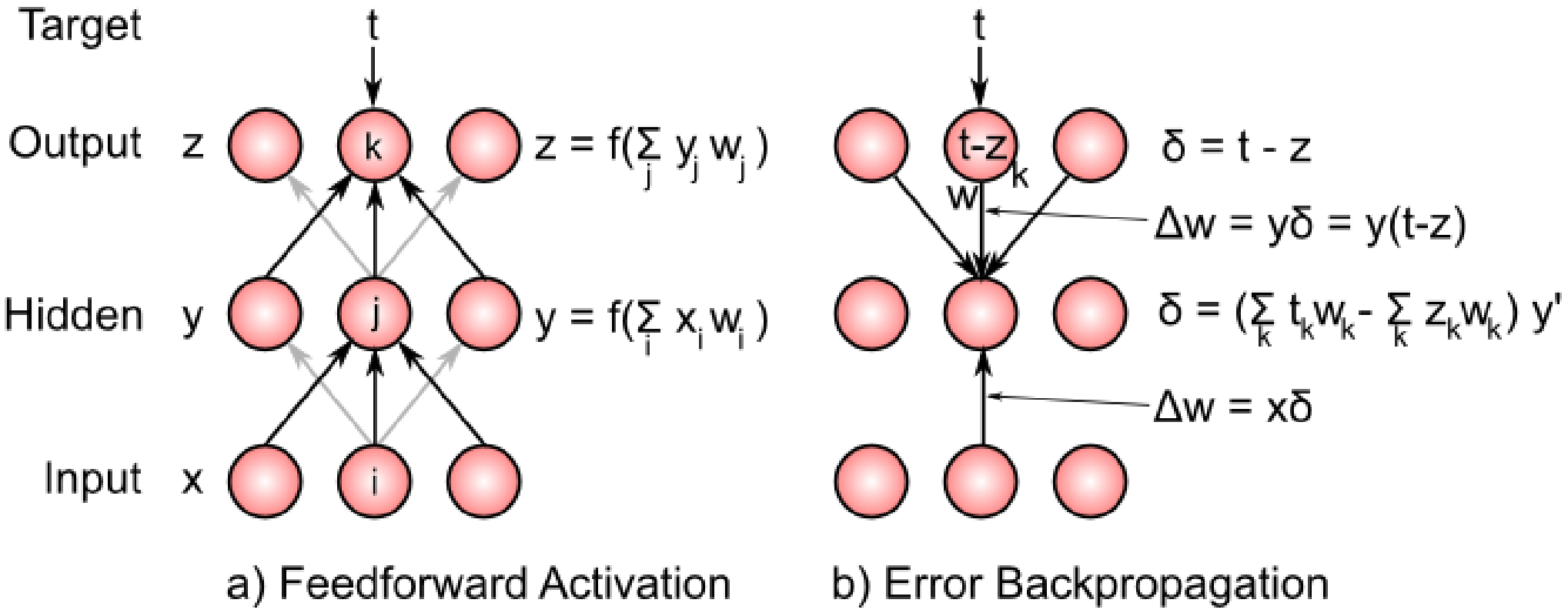
可視為非線性網路的 δ_j

$$\Delta w_{ij}=x_i\delta_j f'(h_j)=x_i(t_j-y_j)f'(h_j)$$

注意：若 $x_i=0$（不在場），則 $\Delta W_{ij}=0$（不是它的錯）

Backpropagation

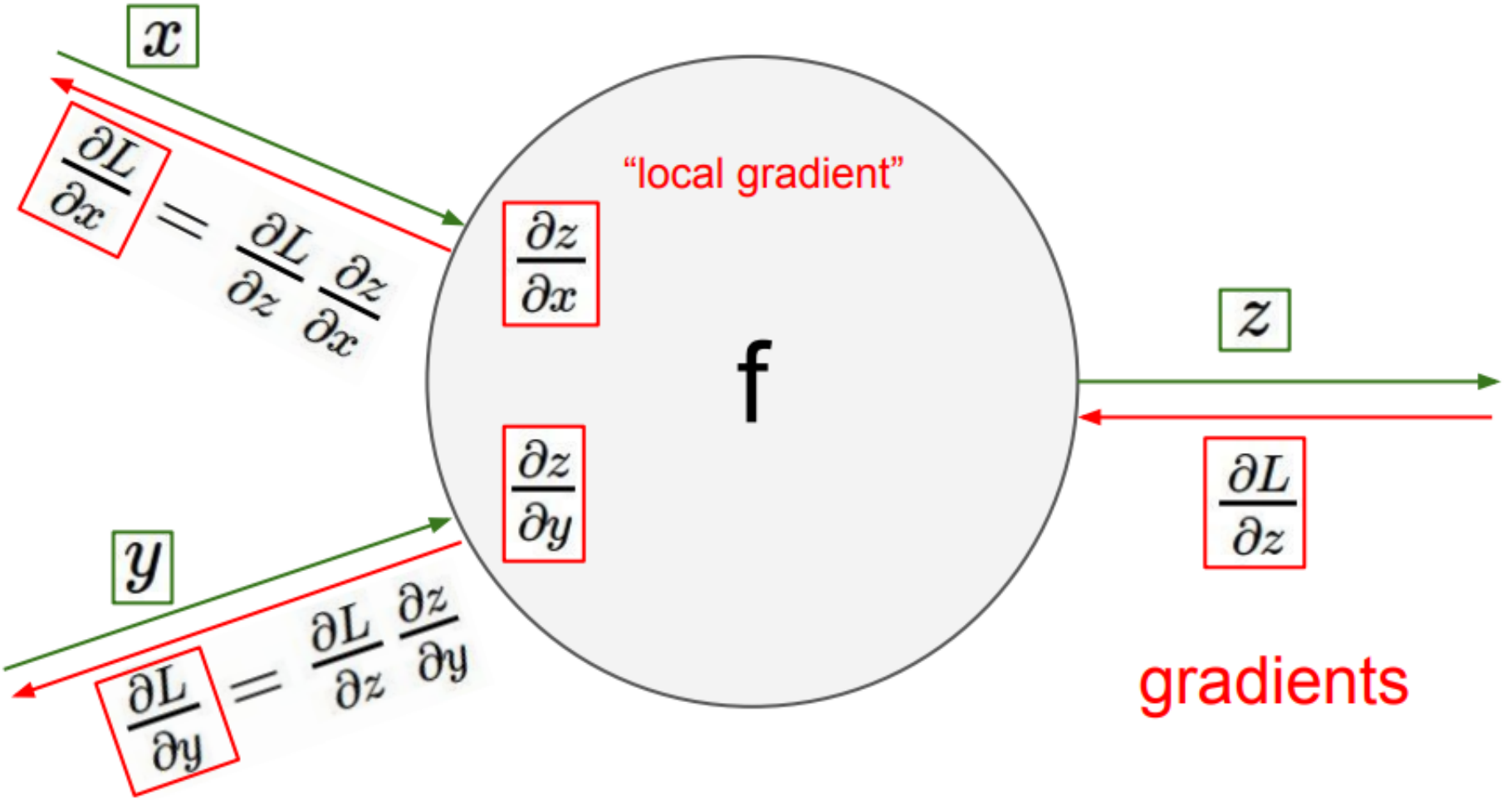
多層 NNs 常用的非生物性演算法（因違反 locality）



當激發函數是 sigmoid 時 BP 的推討請參考這裡

Deep Backpropagation

看似複雜但每個映射只要管好計算自己的梯度即可



Revisiting Biological Plausibility

大腦內可能有接近計算 Back Prop 的方法

Review

Theories of Error Back-Propagation in the Brain

James C.R. Whittington[1,2] and Rafal Bogacz[1,*]

This review article summarises recently proposed theories on how neural circuits in the brain could approximate the error back-propagation algorithm used by artificial neural networks. Computational models implementing these theories achieve learning as efficient as artificial neural networks, but they use simple synaptic plasticity rules based on activity of presynaptic and postsynaptic neurons. The models have similarities, such as including both feedforward and feedback connections, allowing information about error to propagate throughout the network. Furthermore, they incorporate experimental evidence on neural connectivity, responses, and plasticity. These models provide insights on how brain networks might be organised such that modification of synaptic weights on multiple levels of cortical hierarchy leads to improved performance on tasks.

Highlights

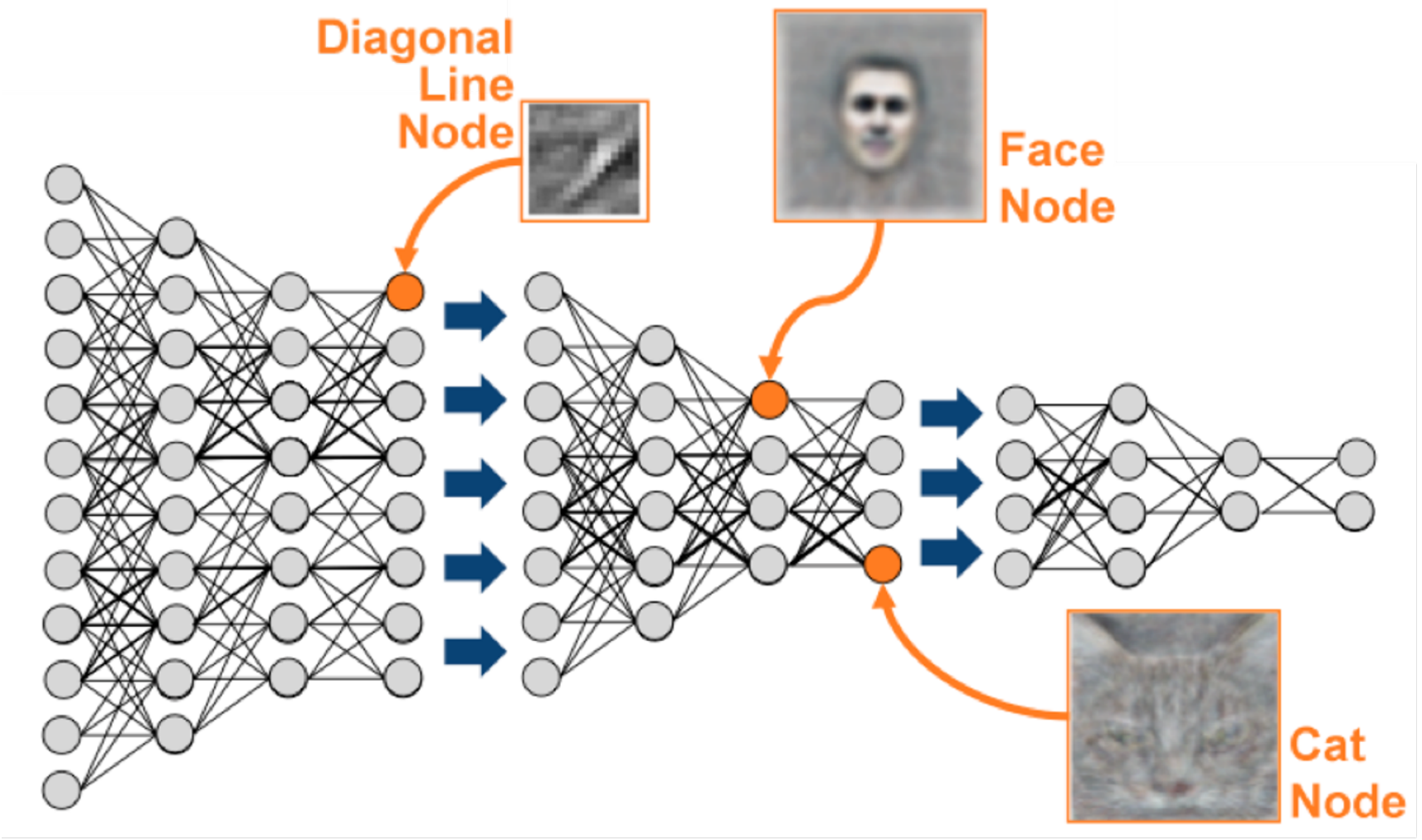
The error back-propagation algorithm can be approximated in networks of neurons, in which plasticity only depends on the activity of presynaptic and postsynaptic neurons.

These biologically plausible deep learning models include both feedforward and feedback connections, allowing the errors made by the network to propagate through the layers.

The learning rules in different biologically plausible models can be implemented with different types of spike-time-dependent plasticity.

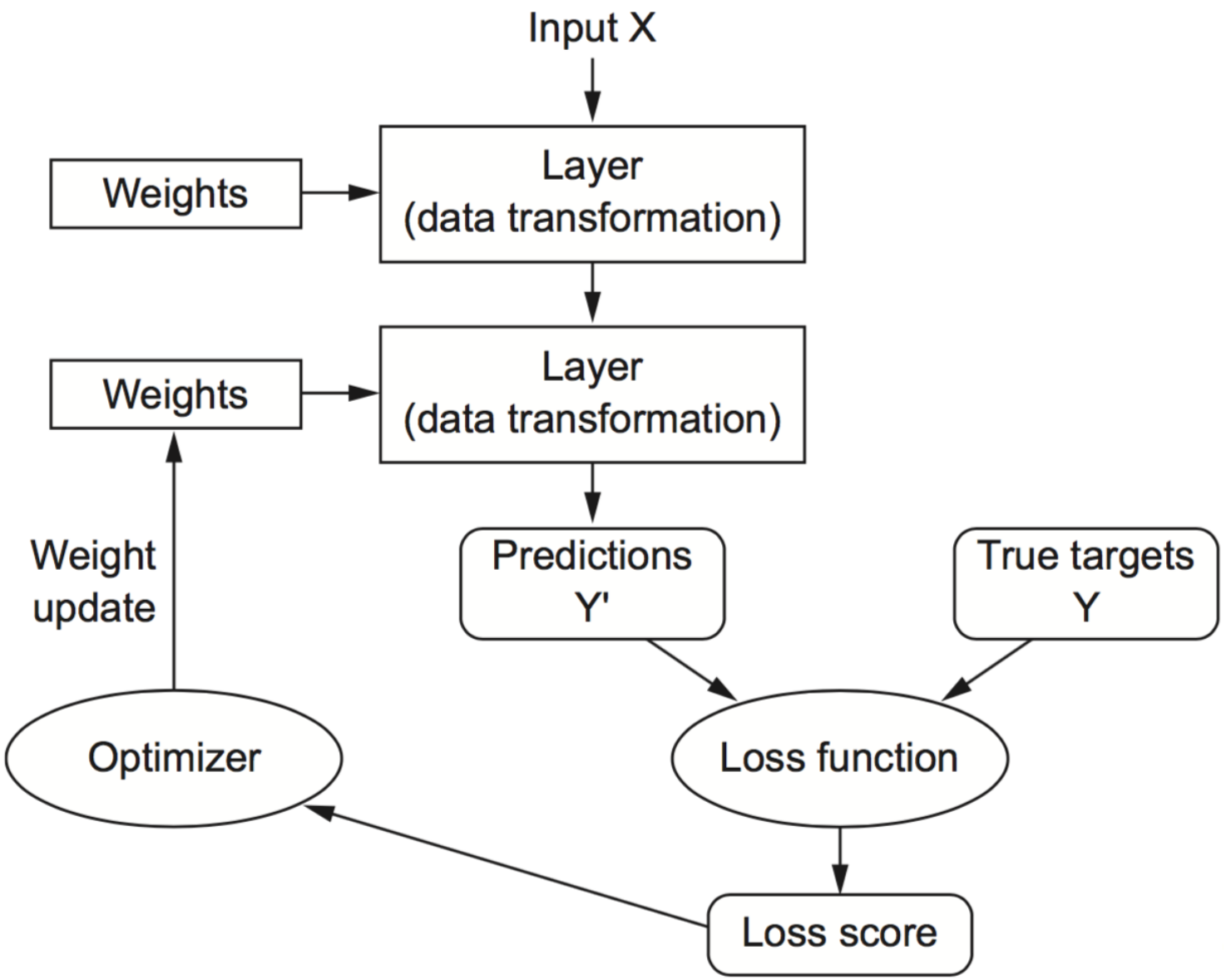
Deep Neural Network

DNN 只是層數比較多的 Neural Net



Deep Supervised Learning

和 Shallow Supervised Learning 程序一樣



Game Over