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

課號：Psy5352
識別碼：227U2810

教室： 普 101
時間： 一 234

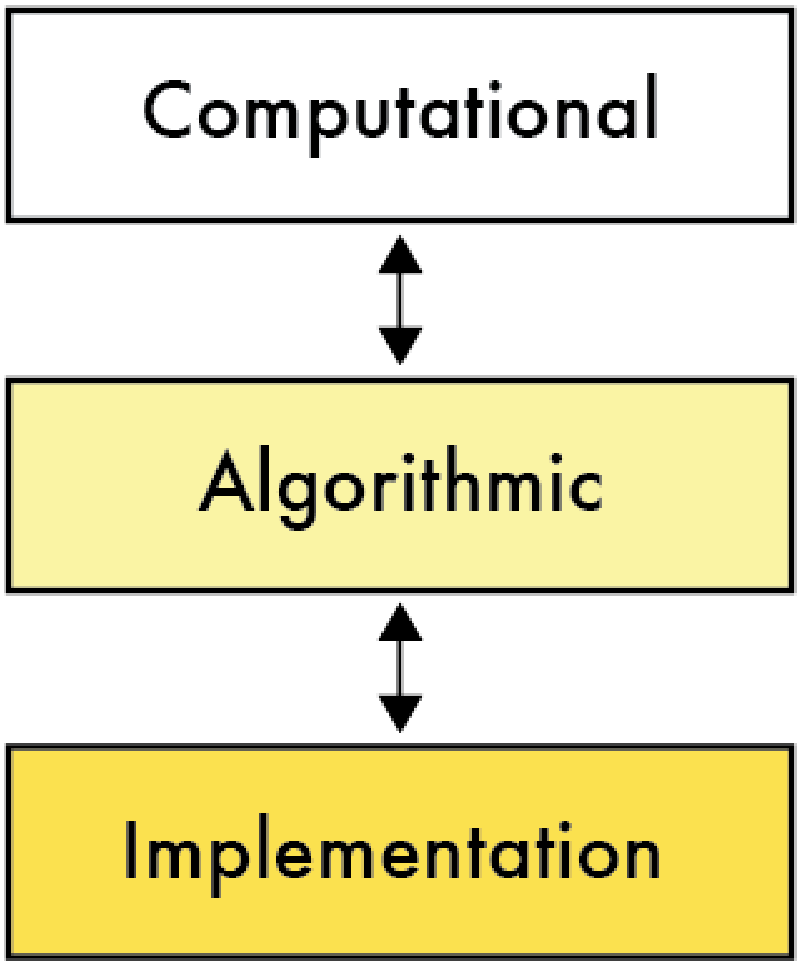

學習的實驗 / 模型較複雜！

Neural Computation

Overview of Learning
Unsupervised Learning
Supervised Learning (1/2)

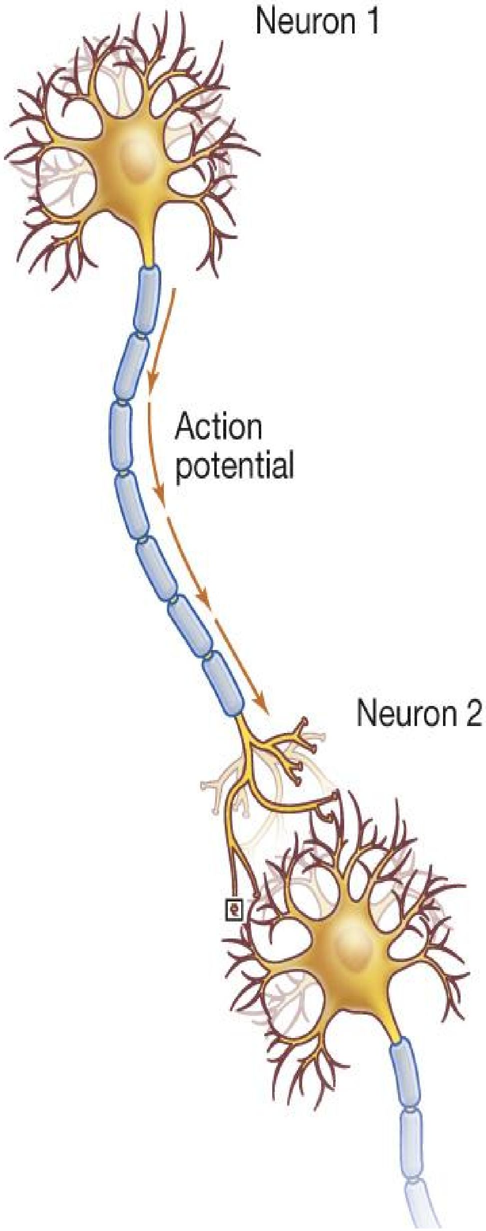
了解一個認知系統的特性與極限

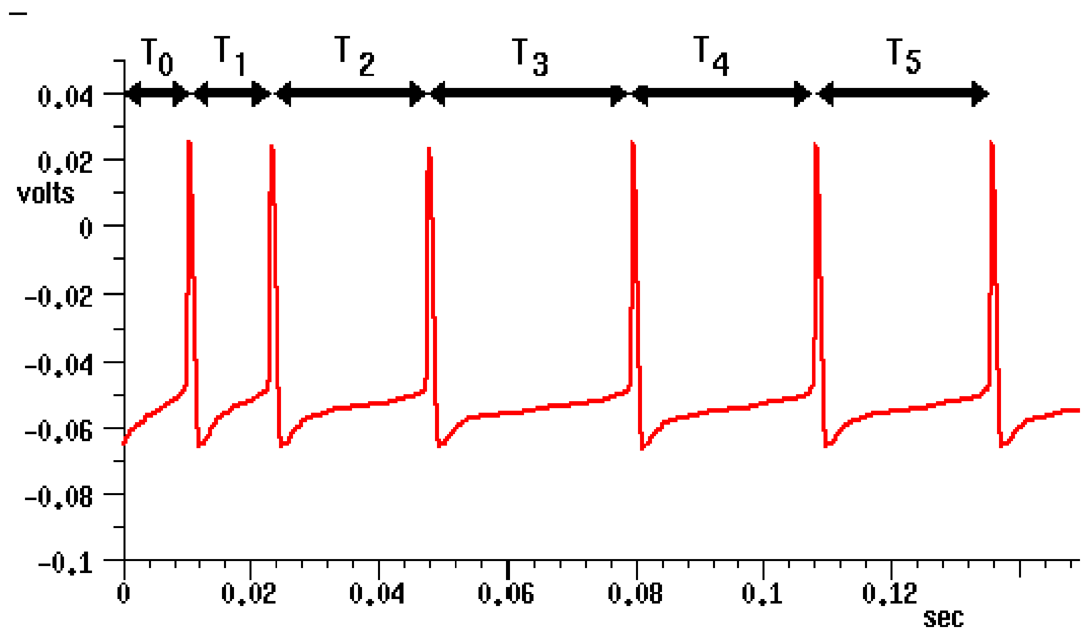
David Marr's 3 levels of analysis

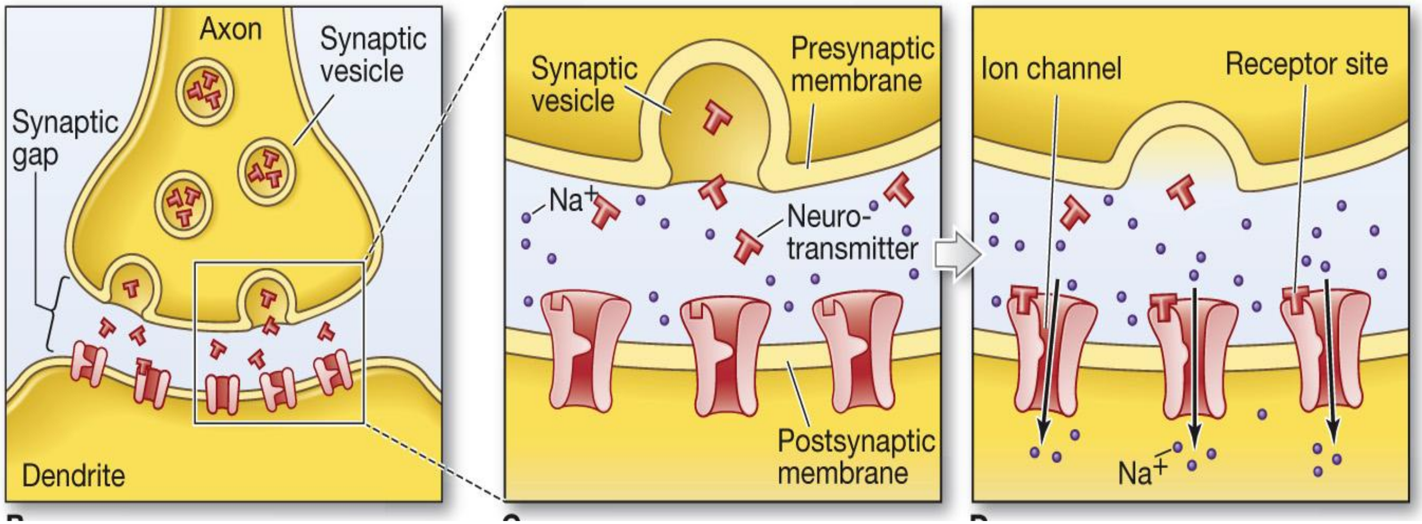


- 計算上的問題是什麼？
- 能夠用什麼演算法來解決？
- 要透過什麼硬體來實現算法？

Implementation: 有 0 與 1 的神經元

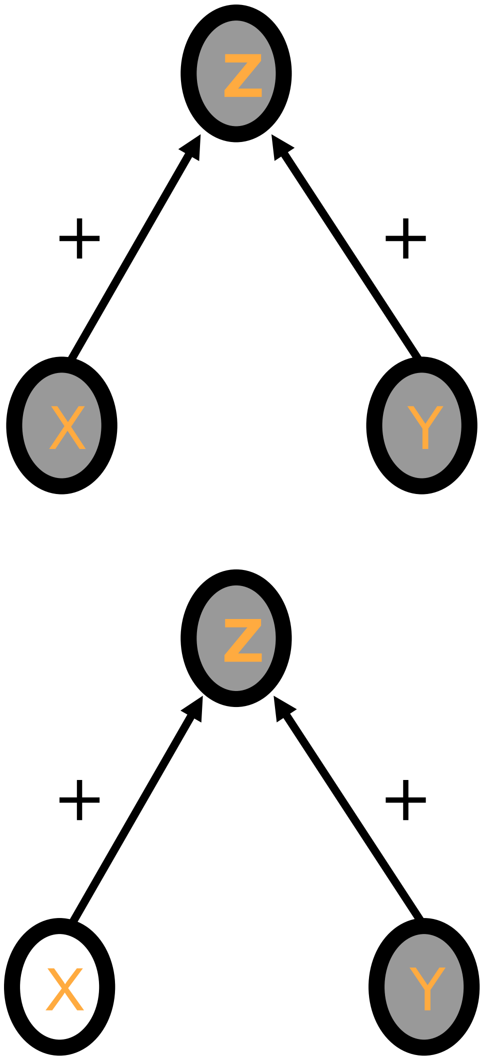






Implementation: 神經元做加法

如果接受刺激的神經元有個低閾值

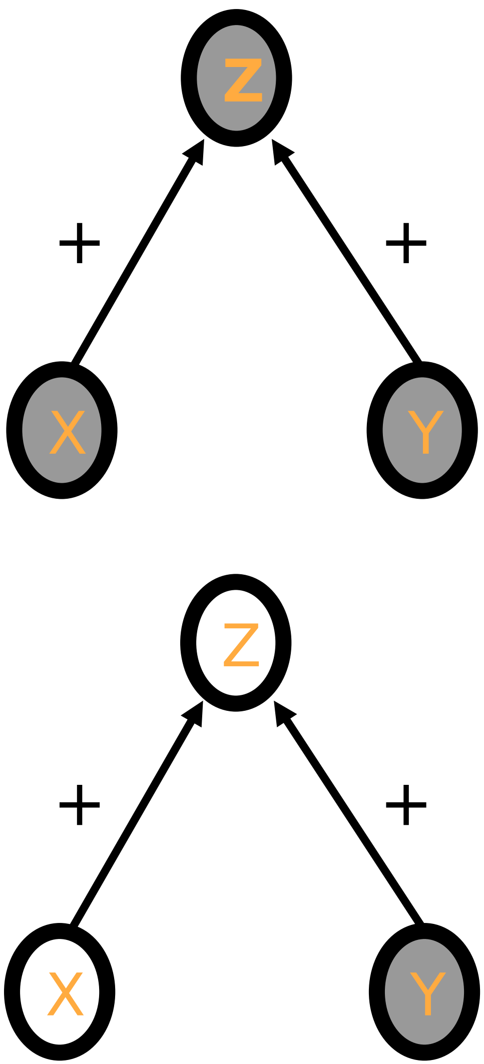


X	Y	Z
0	0	0
0	1	1
1	0	1
1	1	1

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

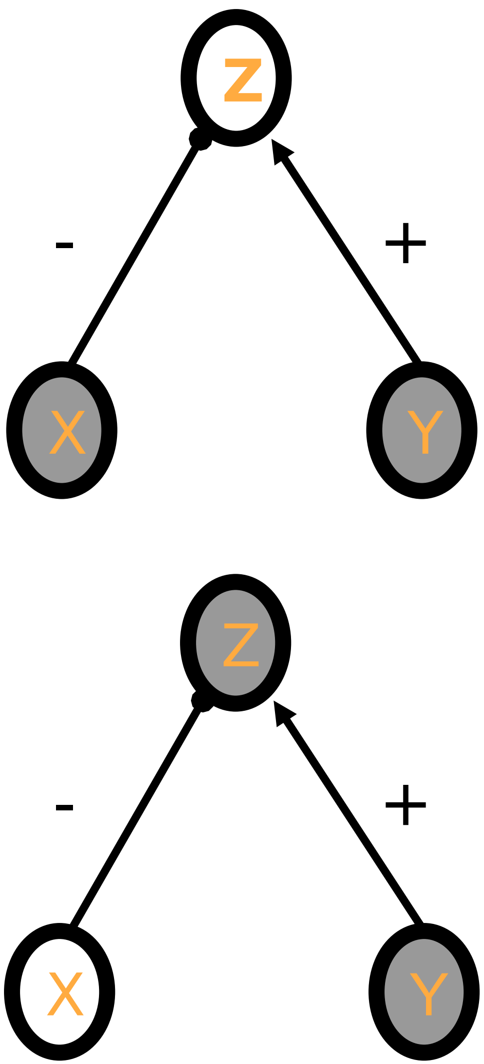


X	Y	Z
0	0	0
0	1	0
1	0	0
1	1	1

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

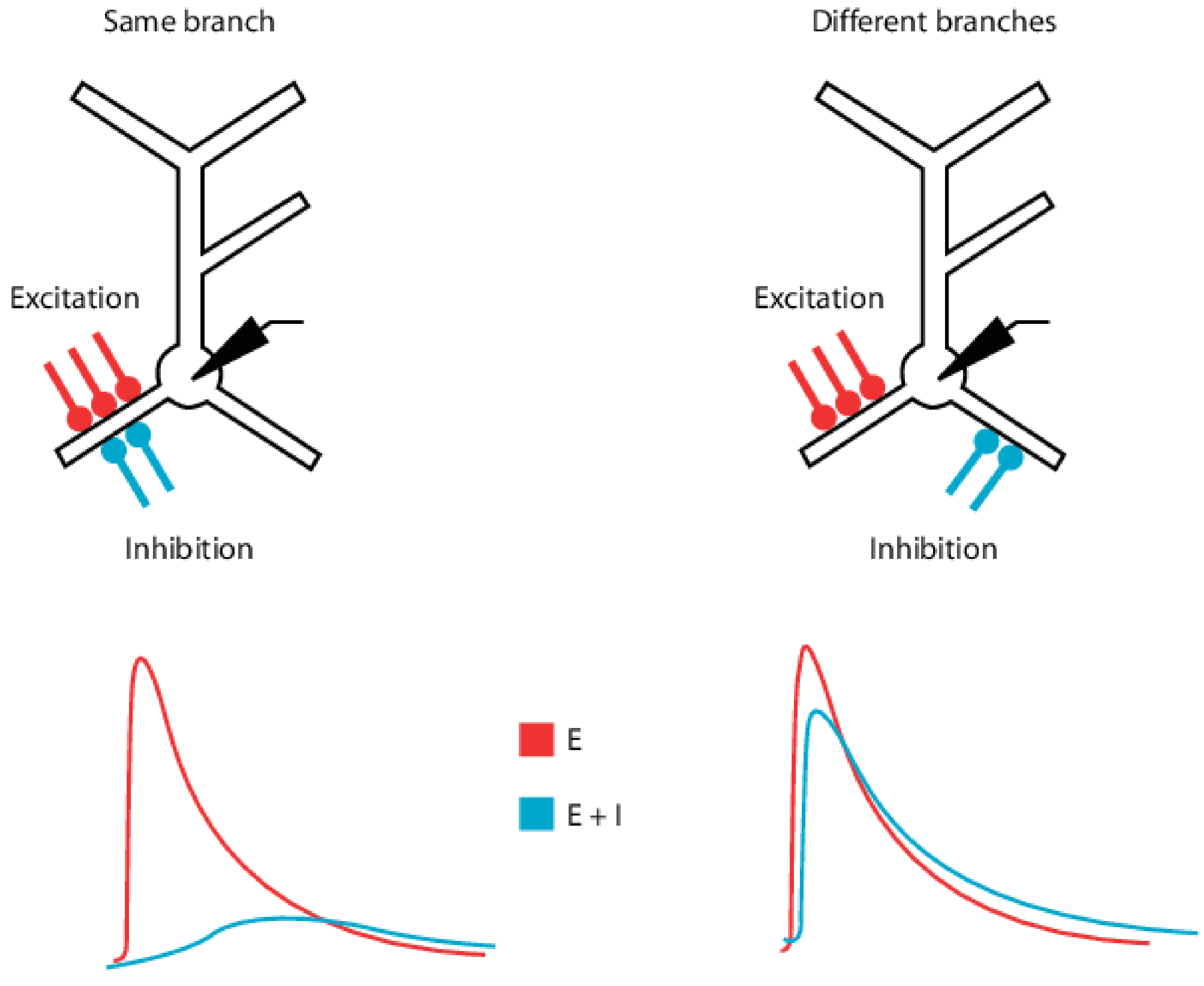


X	Z
0	1
1	0

Z=1-X (NOT)

Implementation: 神經元做除法

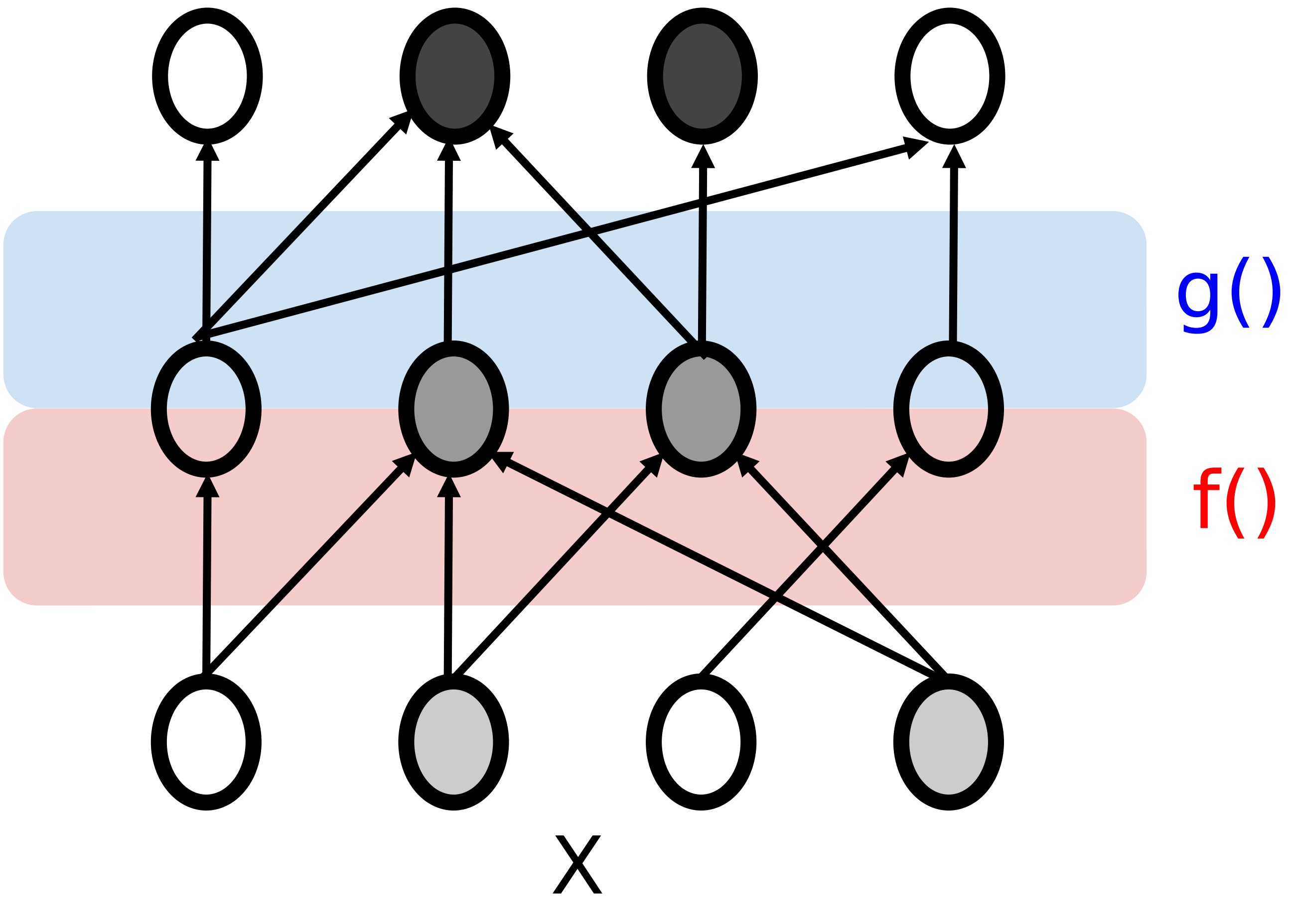
神經元間的抑制可以導致減法或除法



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

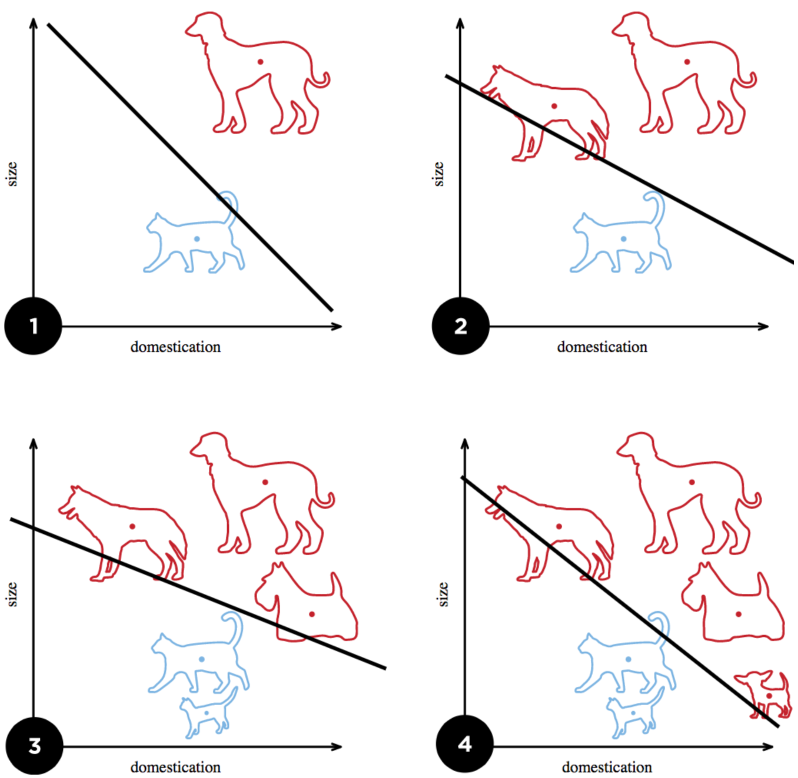
$$Y=g(f(X))$$

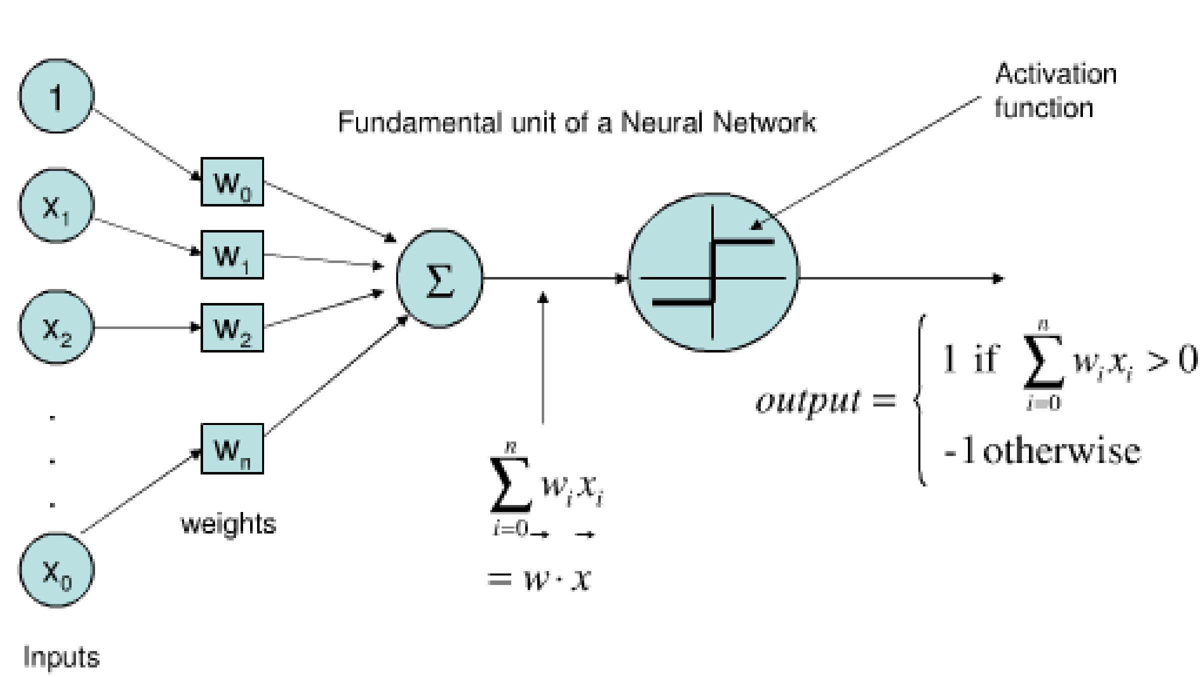


Computational Problem: 區分貓狗

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

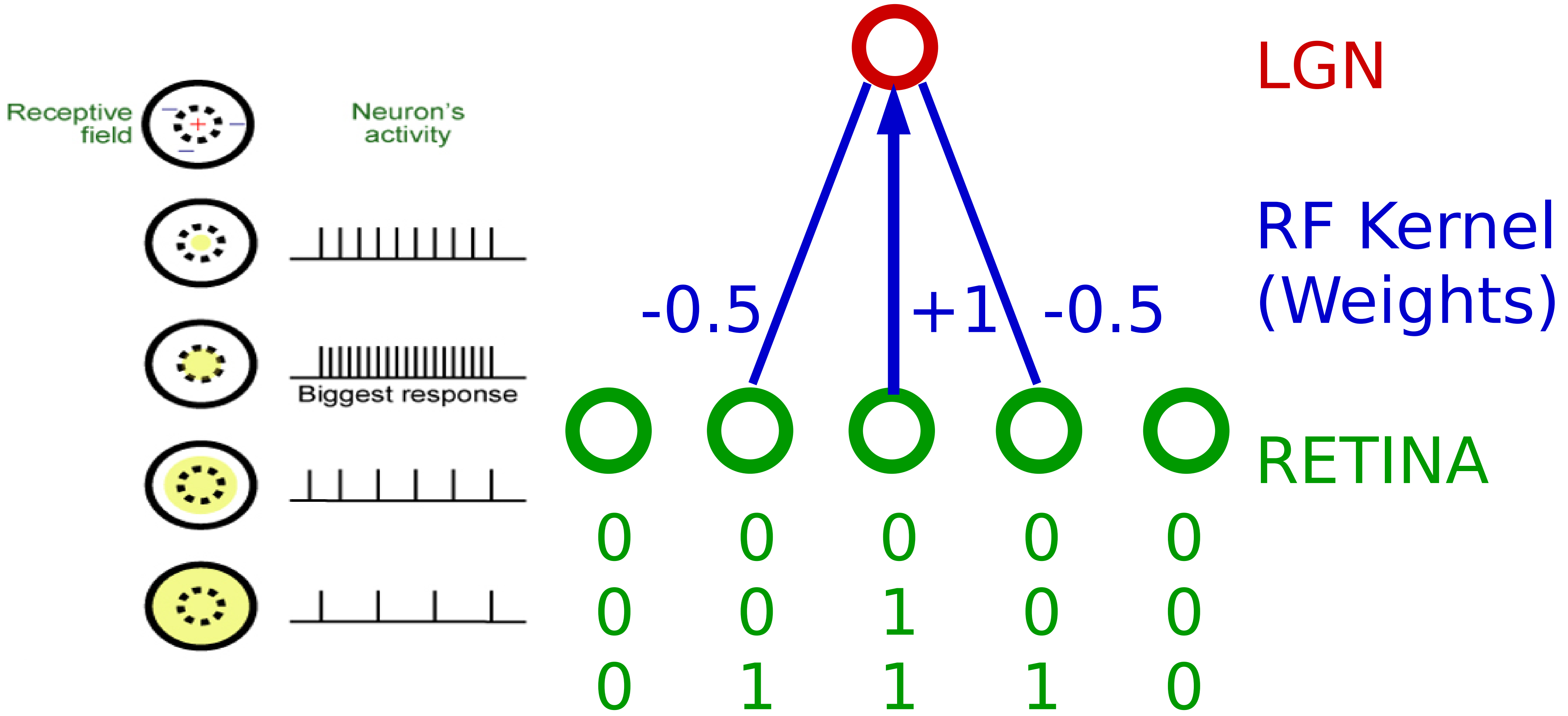
$$y>ax+b \Rightarrow y-ax-b>0 \Rightarrow w_2 x_2+w_1 x_1+w_0 1>0$$





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

Neuron as a Pattern Recognizer

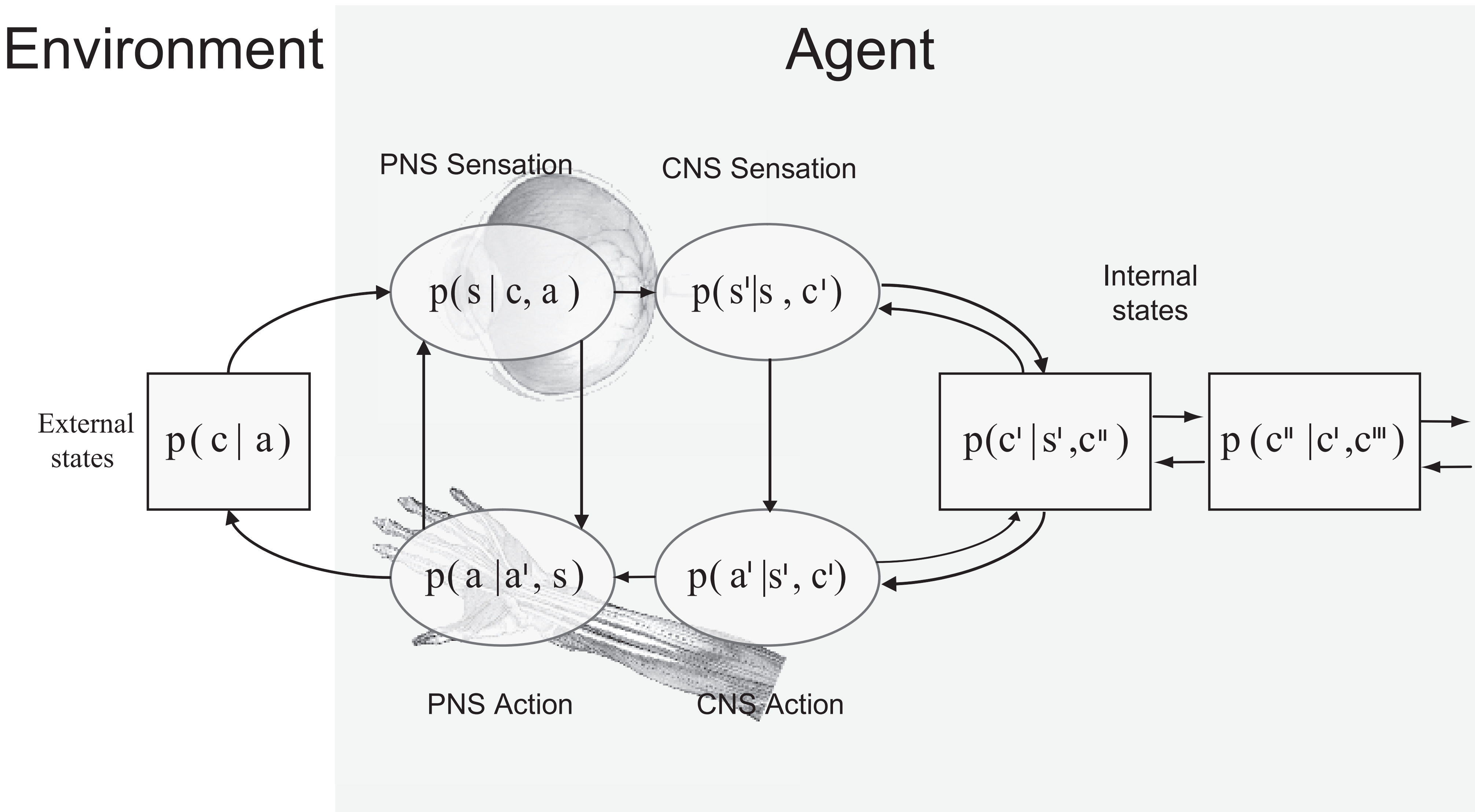



```
RETINA=matrix('0 0 0 0 0;0 0 1 0 0;0 1 1 1 0')
LGN_RF=matrix('0;-0.5;1;-0.5;0')
LGN_RESPONSE=RETINA*LGN_RF
```

Neural Computation
Overview of Learning
Unsupervised Learning
Supervised Learning (1/2)

學習 / 記憶的目的：預測與控制

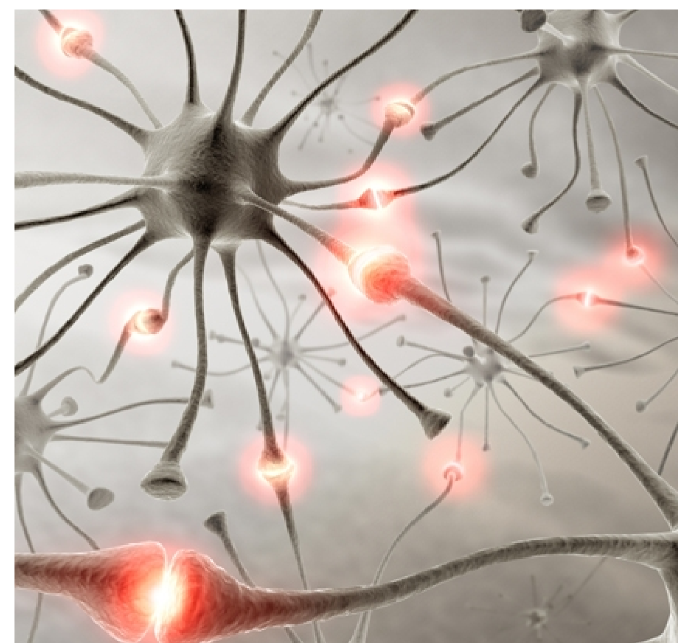
學習和演化的目的都是更適應環境



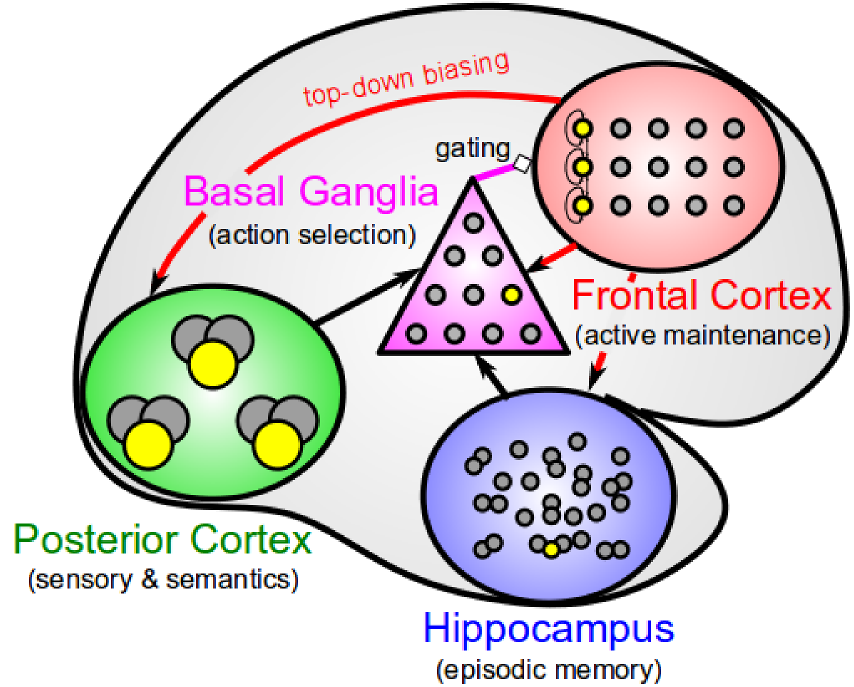
學習中的兩難 (Dilemma)

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。



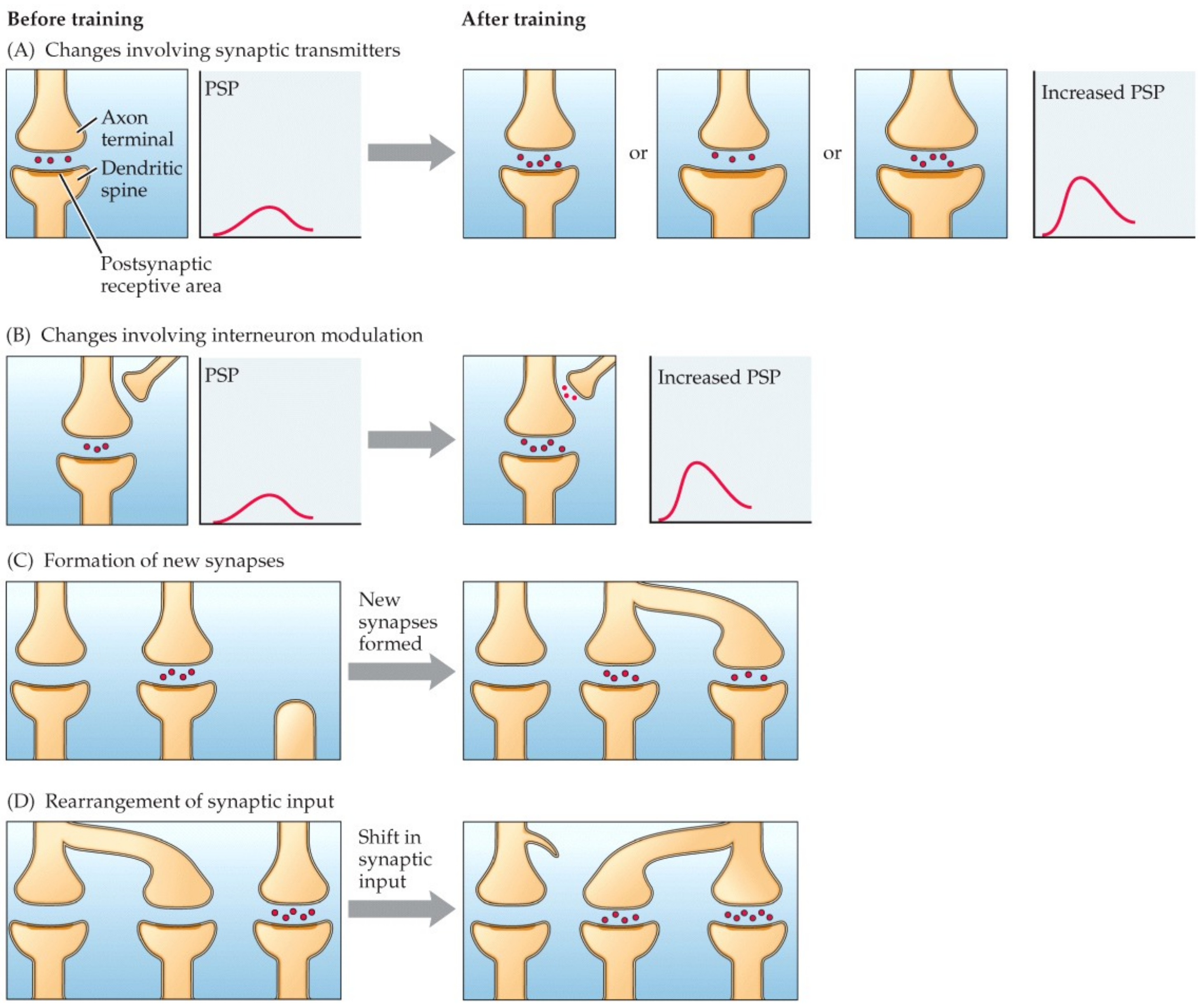
不同腦區的計算 / 學習性質



	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

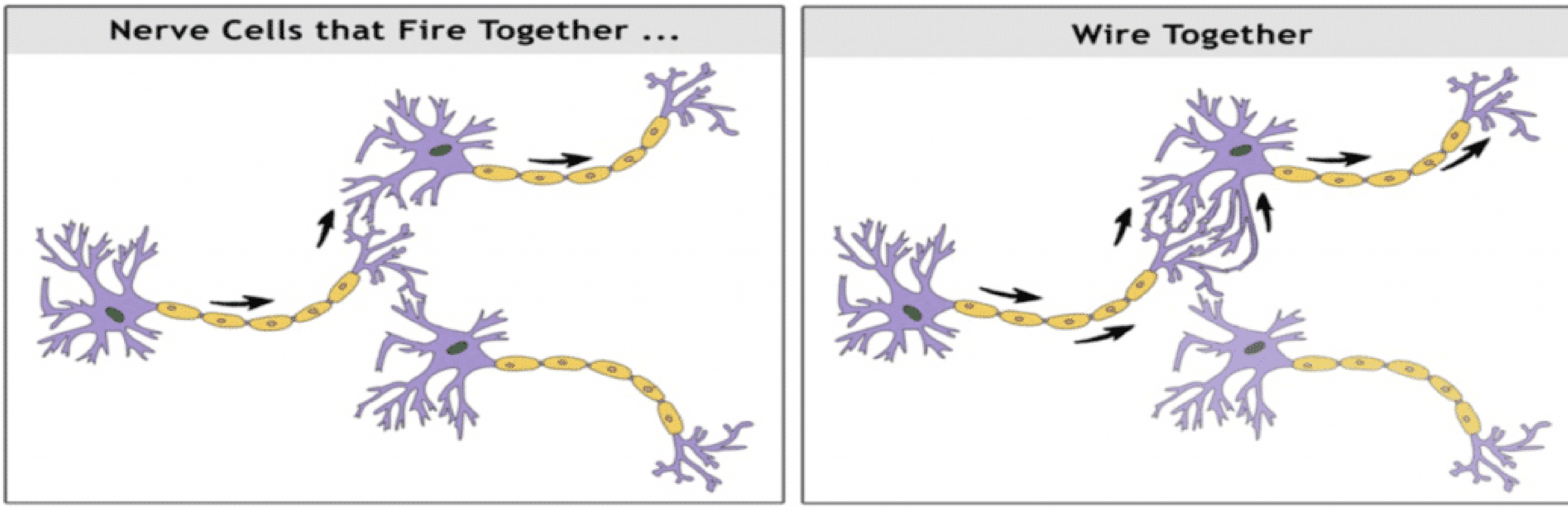
監督式學習：事物有明確標籤或該怎麼做
爸媽教小孩會動的叫動物，不會動的叫植物

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 (1/2)

Hebbian Learning Rule

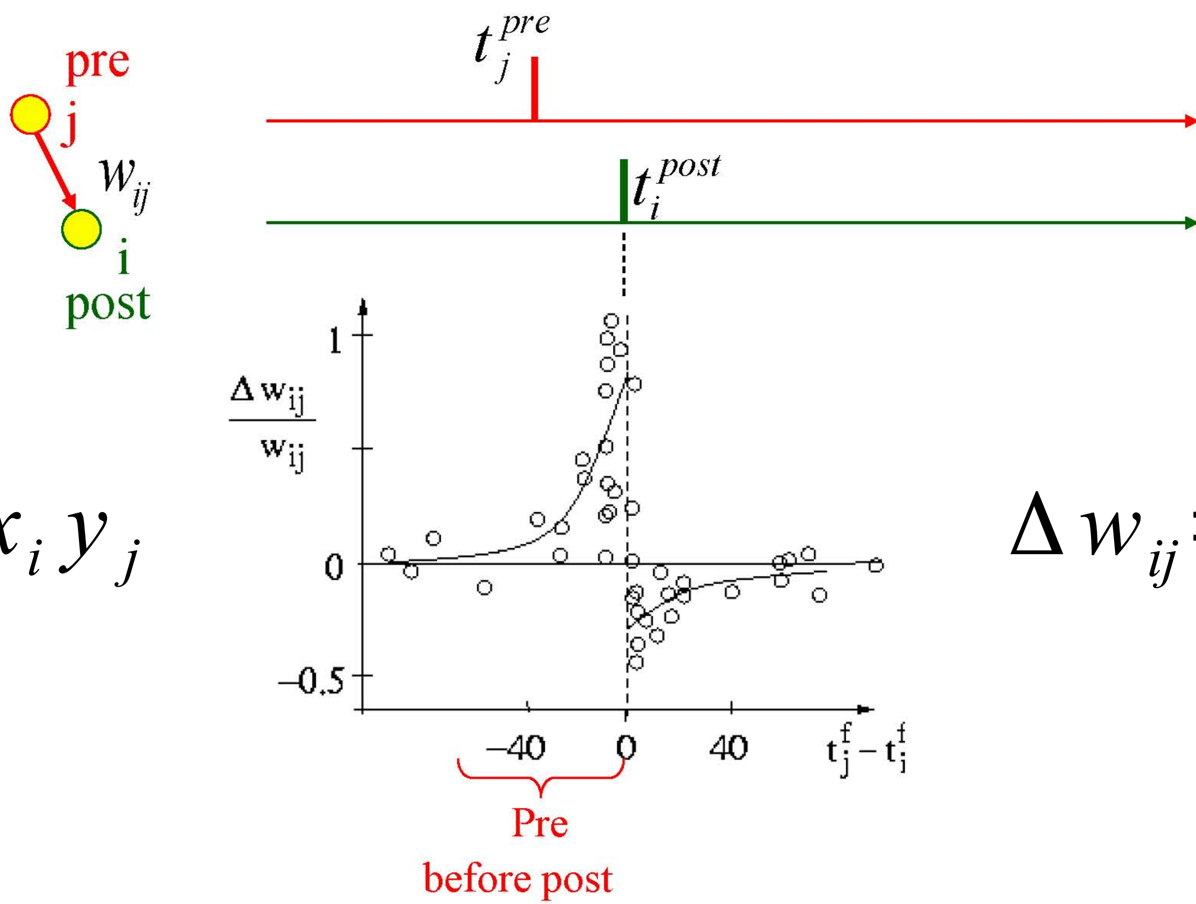
$$w_{ij}^{new} = w_{ij}^{old} + x_i y_j \Rightarrow \Delta w_{ij} = x_i y_j$$



但一神經連接一直強化到後來會不會爆掉？

Hebbian vs. Anti-Hebbian

生物上發現 Spike Timing Dependent Plasticity (STDP) 中兩者並存：

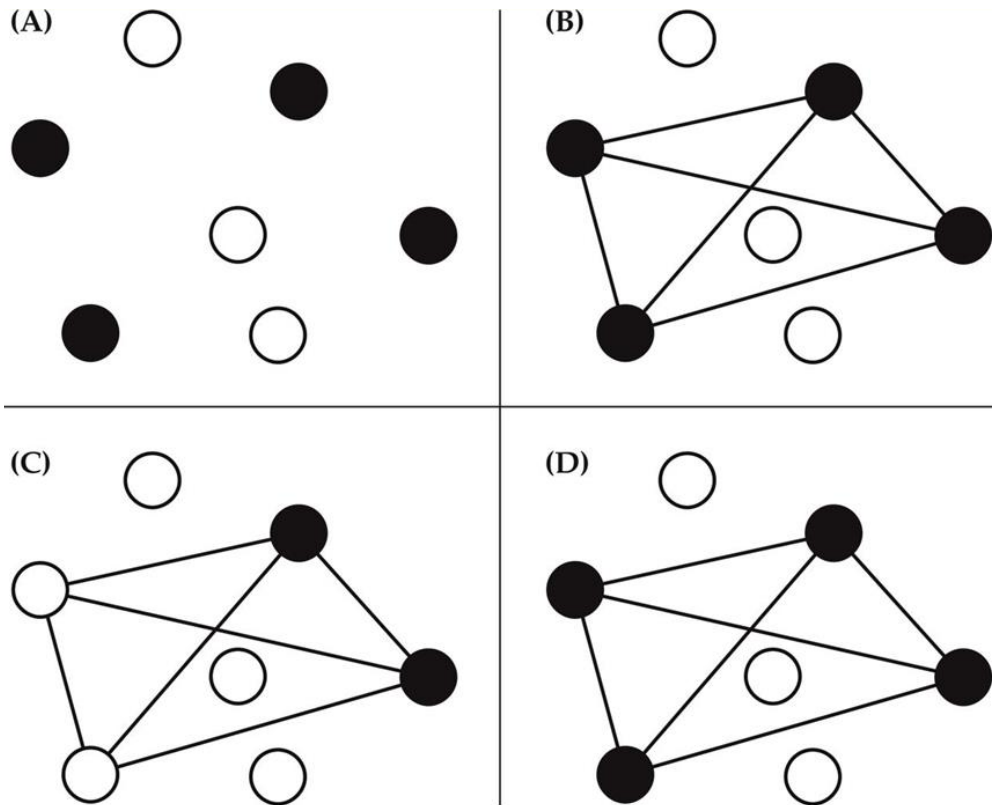


$$\Delta w_{ij} = x_i y_j$$

$$\Delta w_{ij} = -x_i y_j$$

Hopfield Network

學習時是 Hebbian + Anti-Hebbian Rules
回想時是透過證據彼此間的正 / 負相關來投票



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:

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

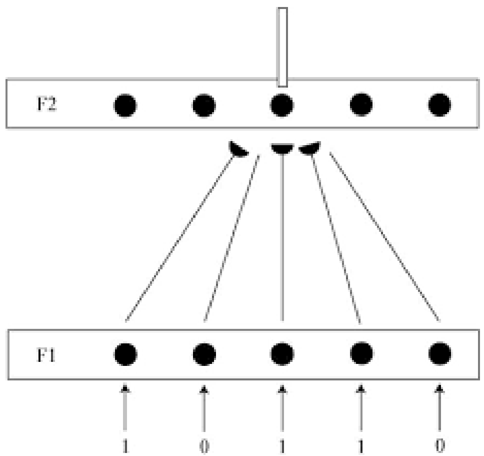
特徵似 hippocampus 的 auto-associative memory

InStar Learning

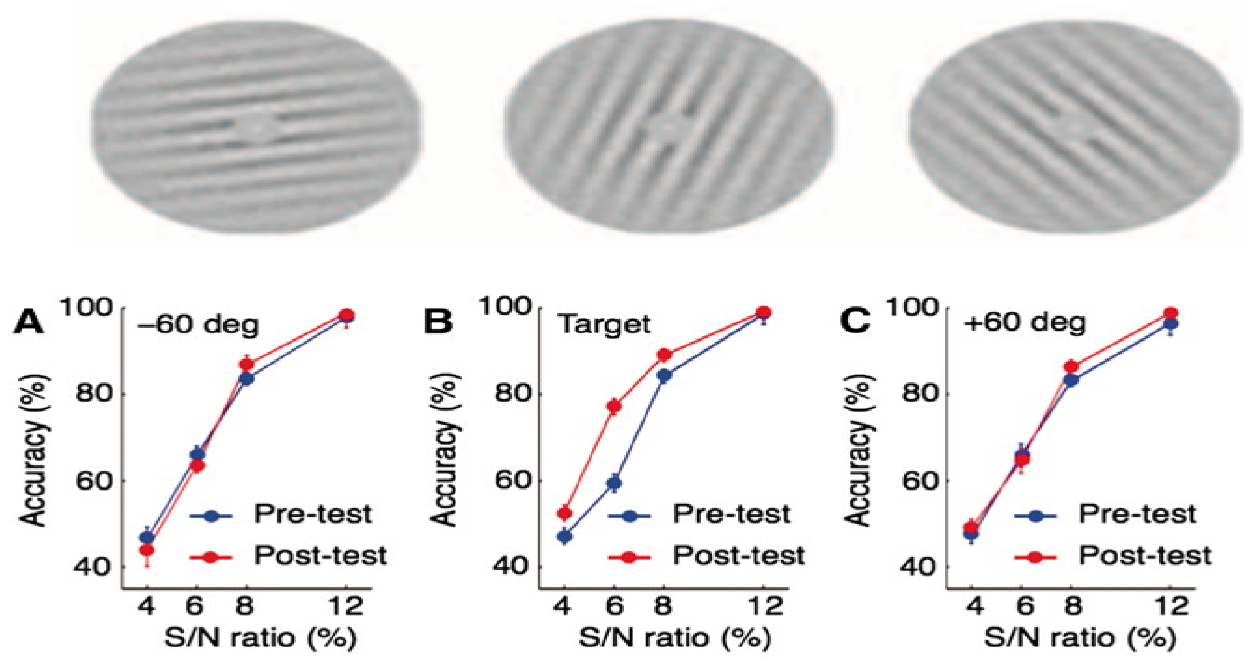
InStar: 突觸後神經元學習辨認突觸前的樣式

Post:

Pre:

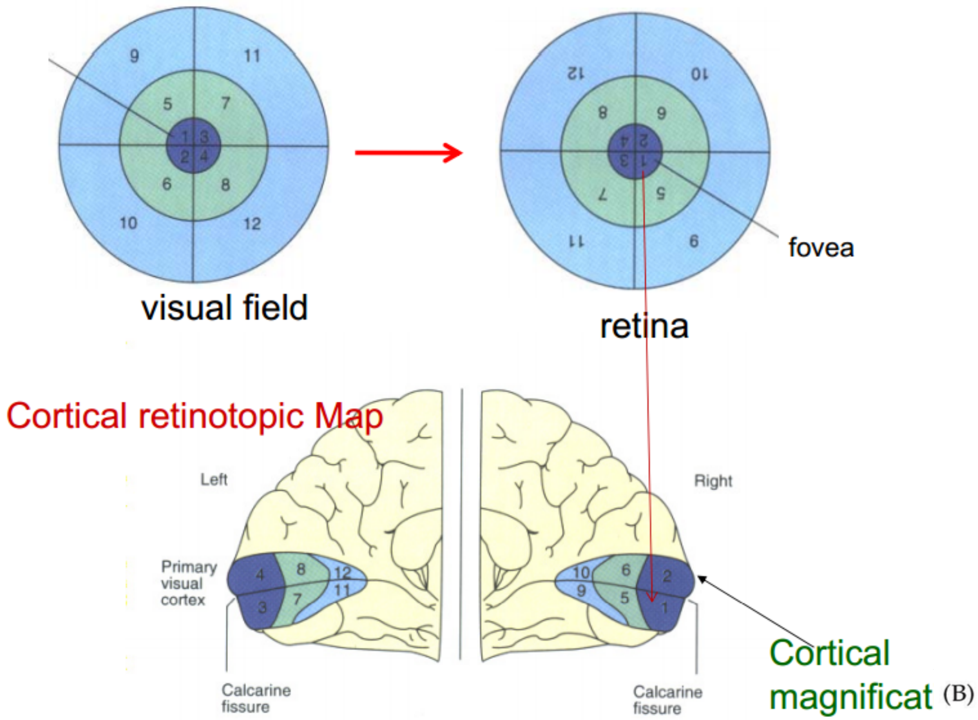


$$\Delta \vec{W}_j^{1\to 2} = y_j\left(\vec{x} - \vec{w}_j^{1\to 2}\right)$$

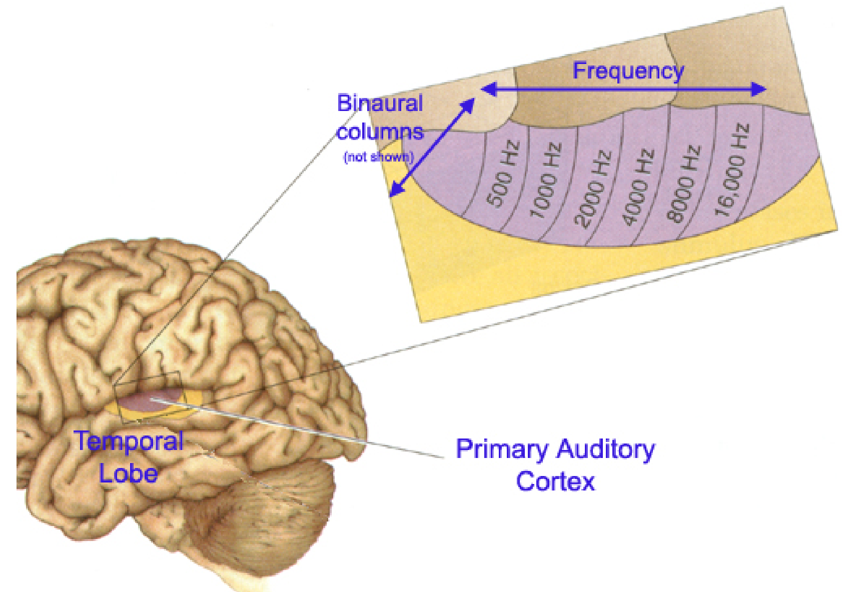


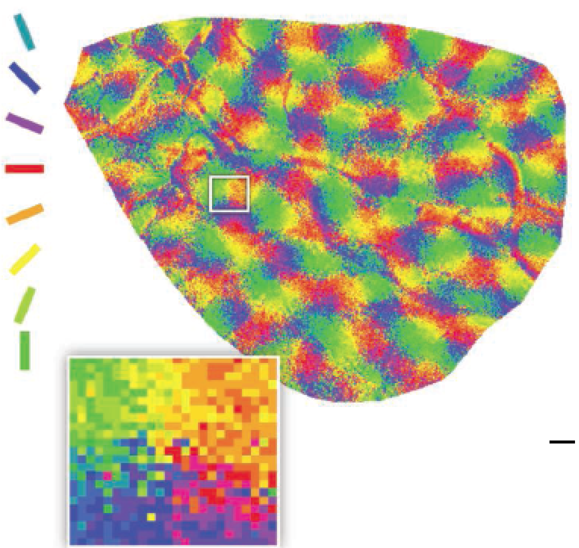

```
W=random.rand(3) # random initial weights
for i in range(10): # 10 trials
  x=array([0,1,0])+0.1*random.rand(3)#same S.+noise
  y=dot(W,x) # the only postsynaptic neuron
  W=W+y*(x-W) # postsynaptically gated InStar
  print(W,y)
```

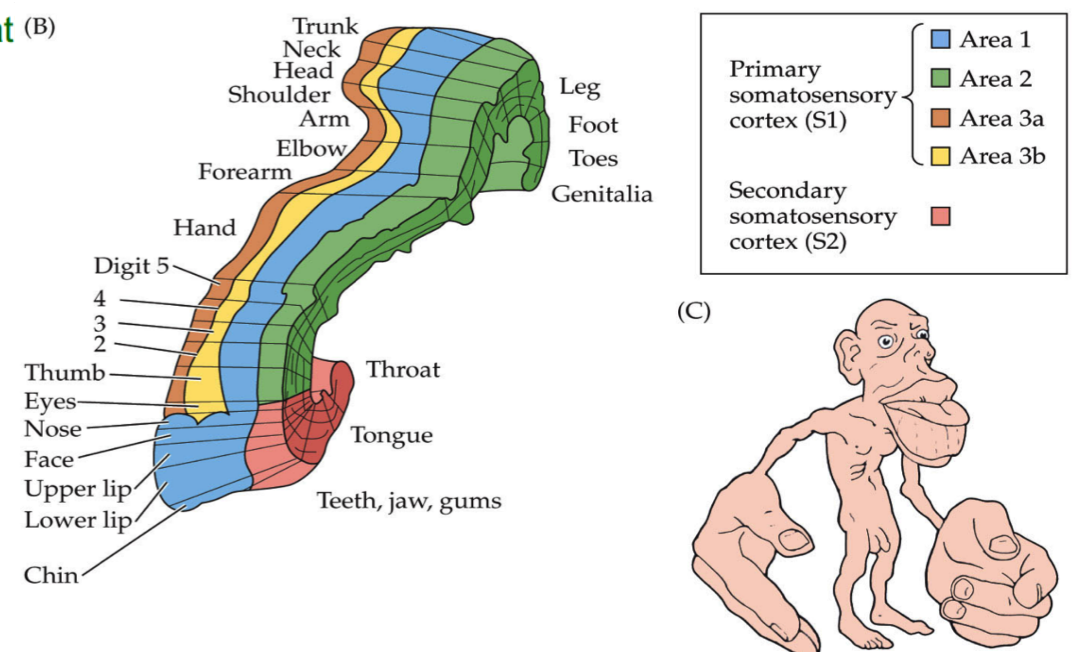
Topological Maps in the Brain



Tonotopic Map Has Columnar Organization





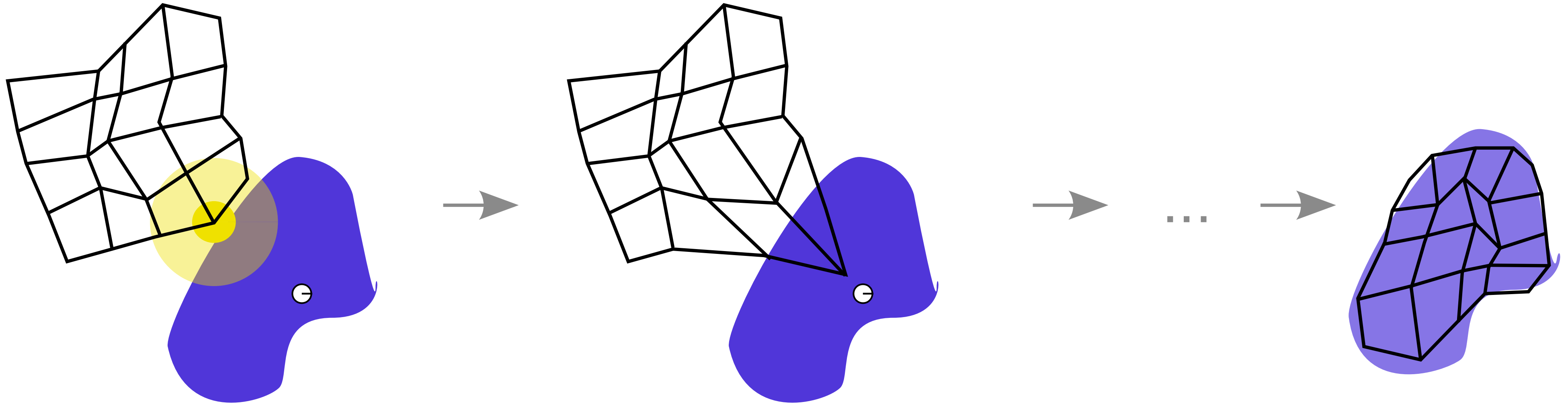


Self-Organizing Map

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)

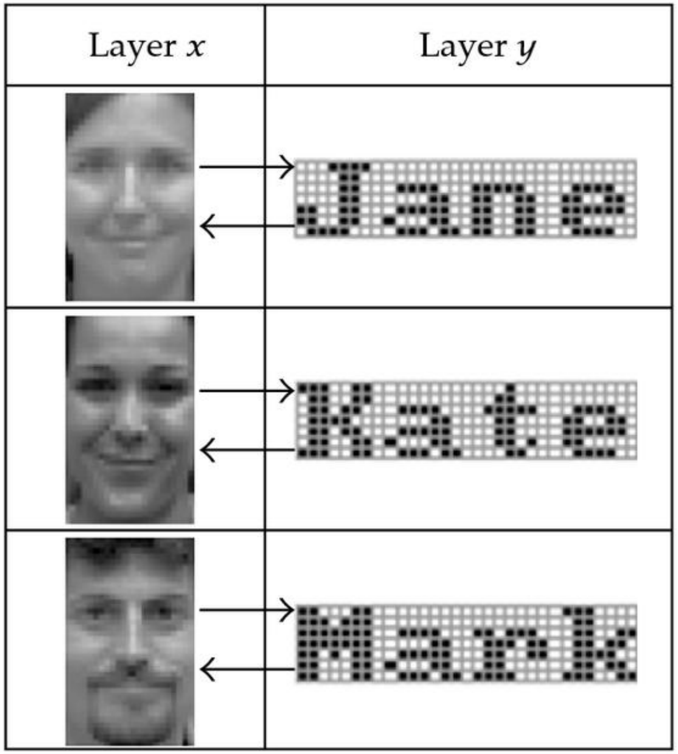
Neural Computation
Overview of Learning
Unsupervised Learning
Supervised Learning (1/2)

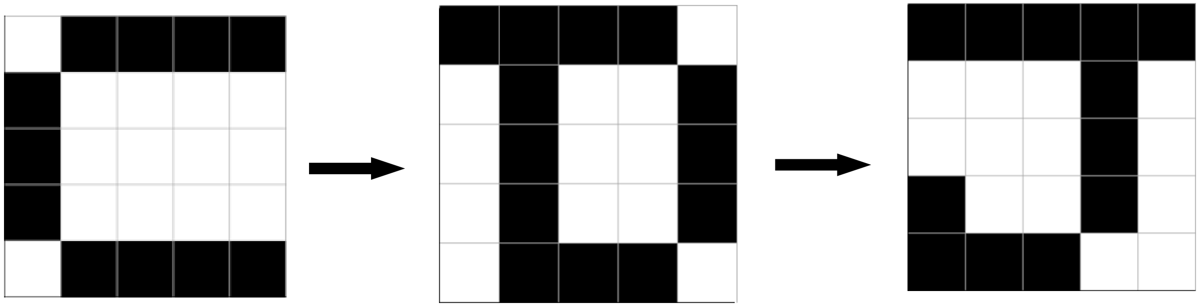
Supervised Hebbian Learning

精神和 Unsupervised Hebbian Learning 一樣：

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

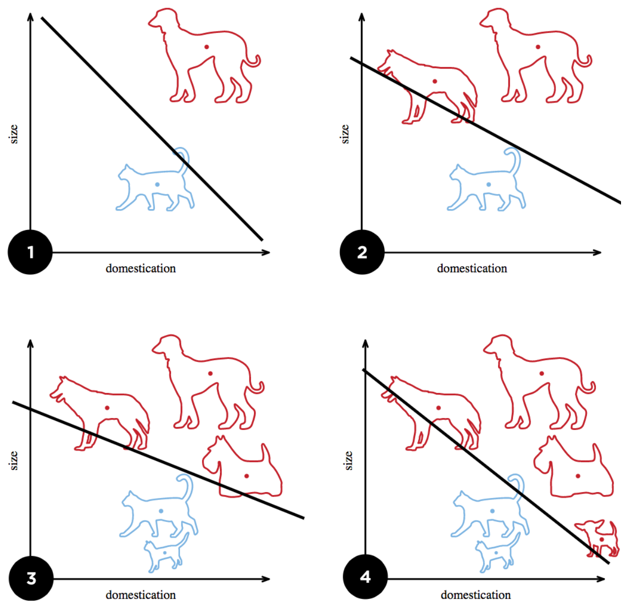
Hopfield 網路可改為 Hetero-associative Memory
來連結不同本質 / 維度的 x 與 y 或產生 x 的序列

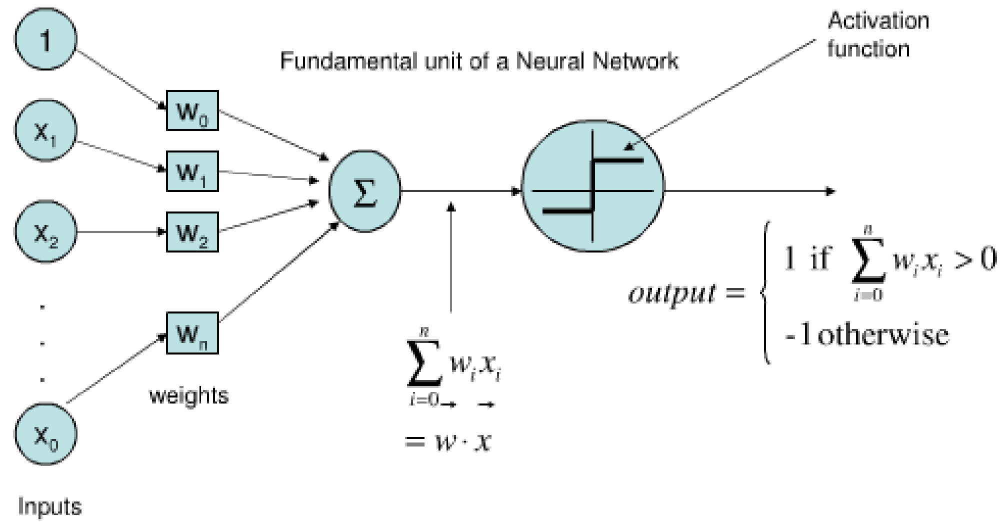




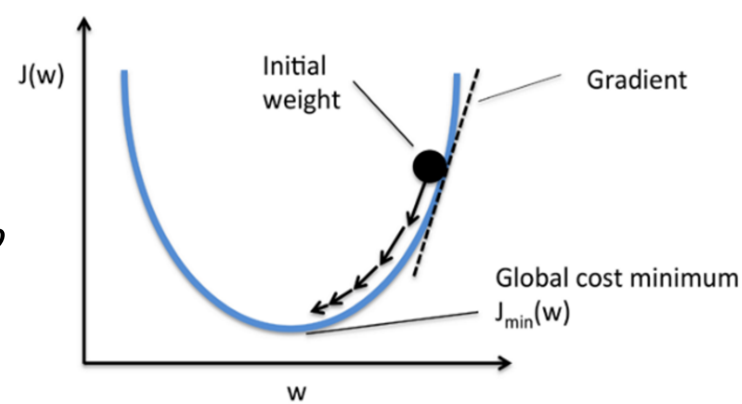
Perceptron：分類

學習規則通常由 gradient descent 推導而來





$$E = -\sum_p (WX^p) Y^p$$



這不就是 Supervised Hebbian 嗎 !?

$$W^{new} = W^{old} + \Delta W = W^{old} - \partial E(W)/\partial W = W^{old} + \sum_p \underline{X^p Y^p}$$

Game Over