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

黃從仁

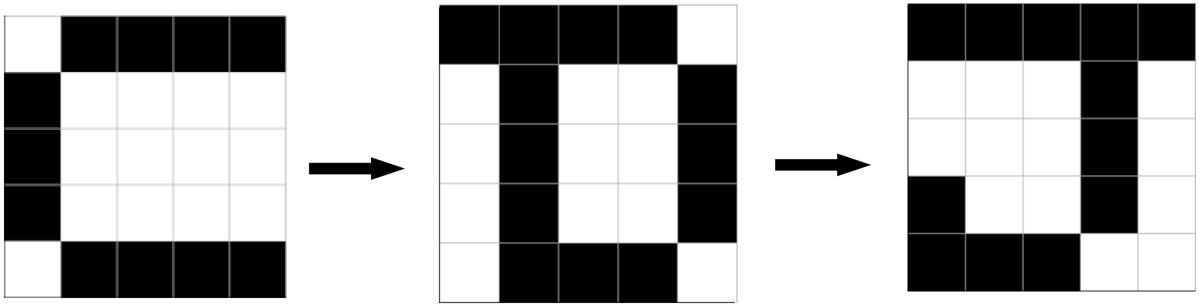
今天討論有回饋的學習！

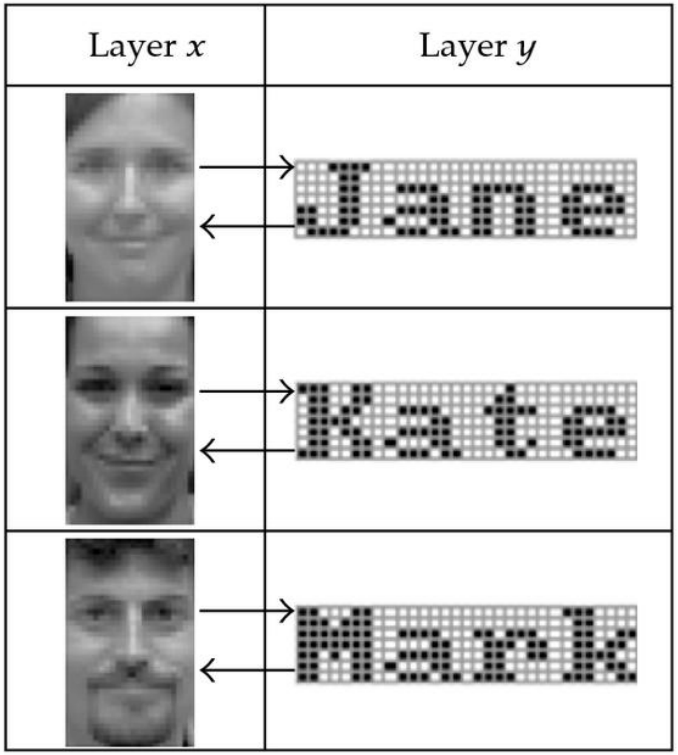
Supervised Learning
Reinforcement Learning

Supervised Hebbian Learning

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

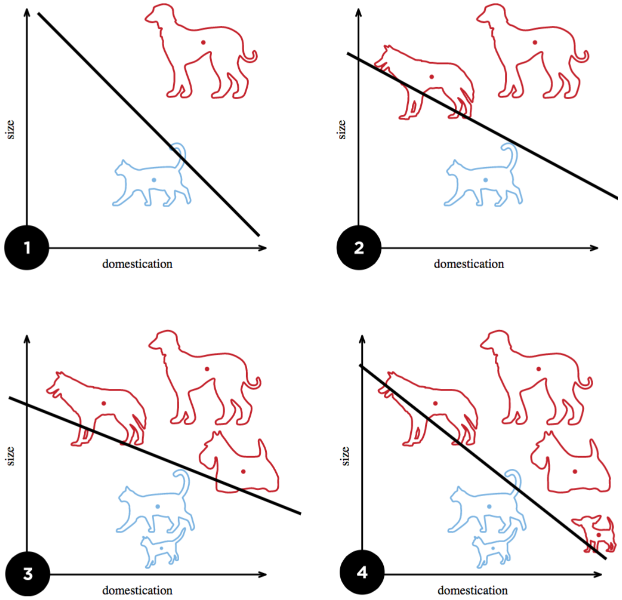
$$w_{ij} = w_{ij} + \frac{1}{n}\sum_{p=1}^{n}\Delta w_{ij} = 0 + \frac{1}{n}\sum_{p=1}^{n} x_i^p x_j^p \rightarrow \frac{1}{n}\sum_{p=1}^{n} x_i^p y_j^p$$

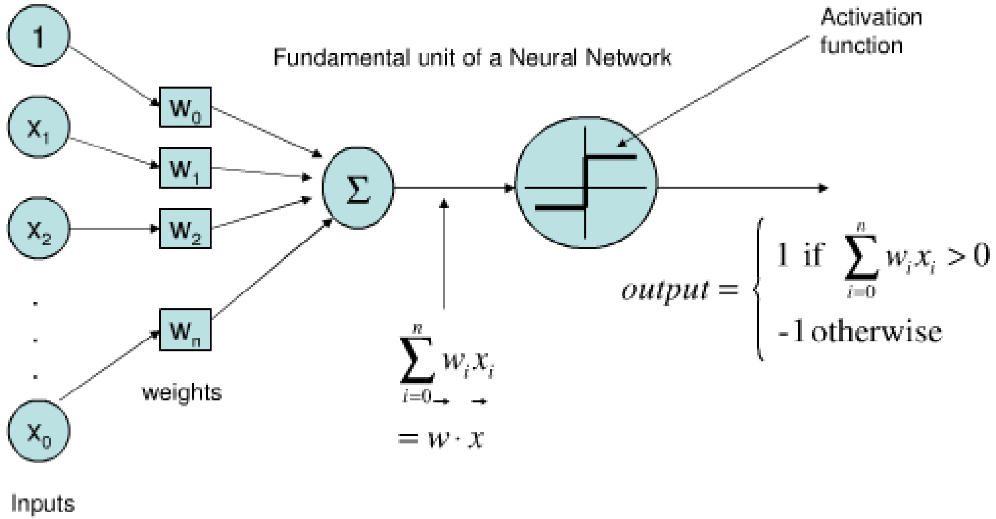




分類問題：Perceptron

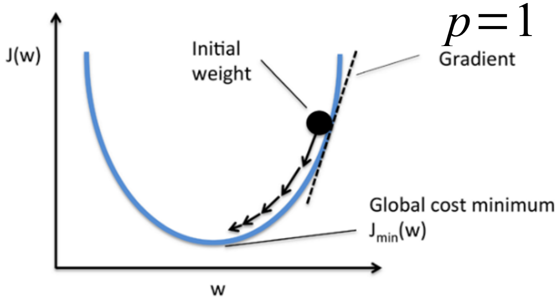
學習規則通常由 gradient descent 推導而來




Fundamental unit of a Neural Network
Activation function
1
x_1
x_2
x_0
Inputs
w_0
w_1
w_2
w_n
weights
Σ
$\sum_{i=0}^{n} w_i x_i = \vec{w} \cdot \vec{x}$
$output = \begin{cases} 1 \text{ if } \sum_{i=0}^{n} w_i x_i > 0 \\ -1 \text{ otherwise} \end{cases}$

因階梯函數不可微分，改成可微分的形式：

$$E = -\sum_{p=1}^{n} SIGN\left(WX^{p}\right)Y^{p} \approx -\sum_{p=1}^{n}\left(WX^{p}\right)Y^{p}$$



這不就是 Supervised Hebbian 嗎 !?

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

迴歸問題：Delta Rule

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

$$y_j=\sum_i w_{ij} x_i, E=\sum_j (y_j-t_j)^2/2$$

$$\Delta w_{ij}=-\partial E(w_{ij})/\partial w_{ij}=-(y_j-t_j)*x_i=\delta_j x_i$$

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：

$$y_j=f(\sum_i w_{ij} x_i), E=\sum_j (y_j-t_j)^2/2$$

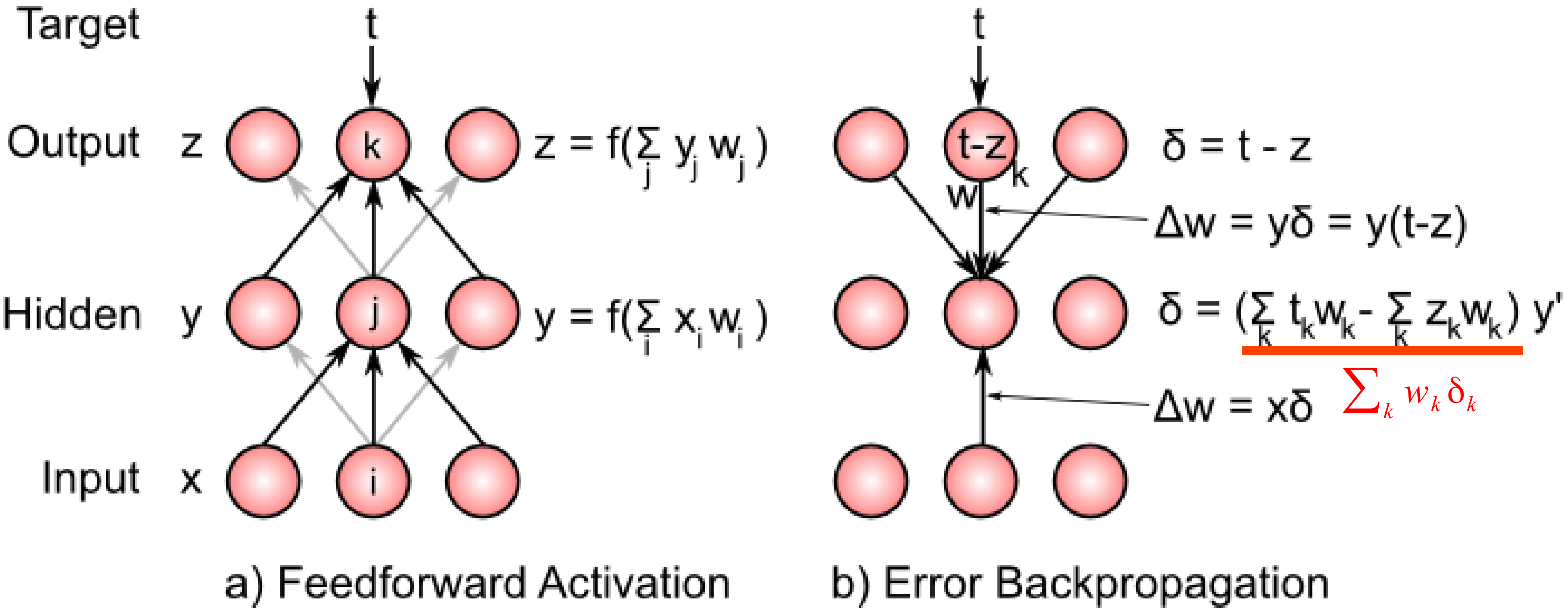
可視為非線性網路的 δ_j

$$\Delta w_{ij}=-\partial E(w_{ij})/\partial w_{ij}=\underline{-(y_j-t_j) f'(\sum_i w_{ij} x_i)}x_i$$

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 directionality & locality)



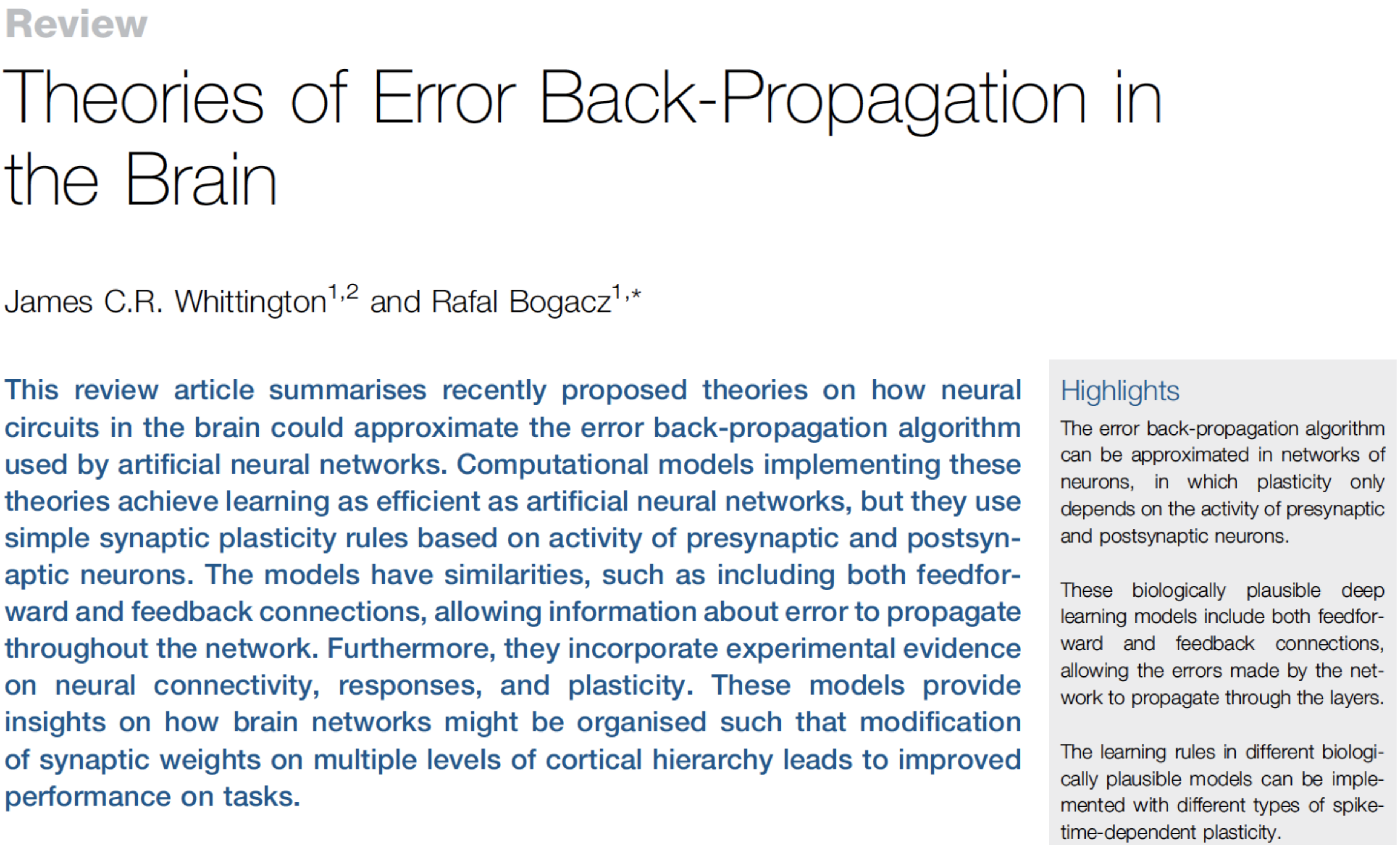
a) Feedforward Activation

b) Error Backpropagation

當激發函數是 sigmoid 時 BP 的推討請參考這裡

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.

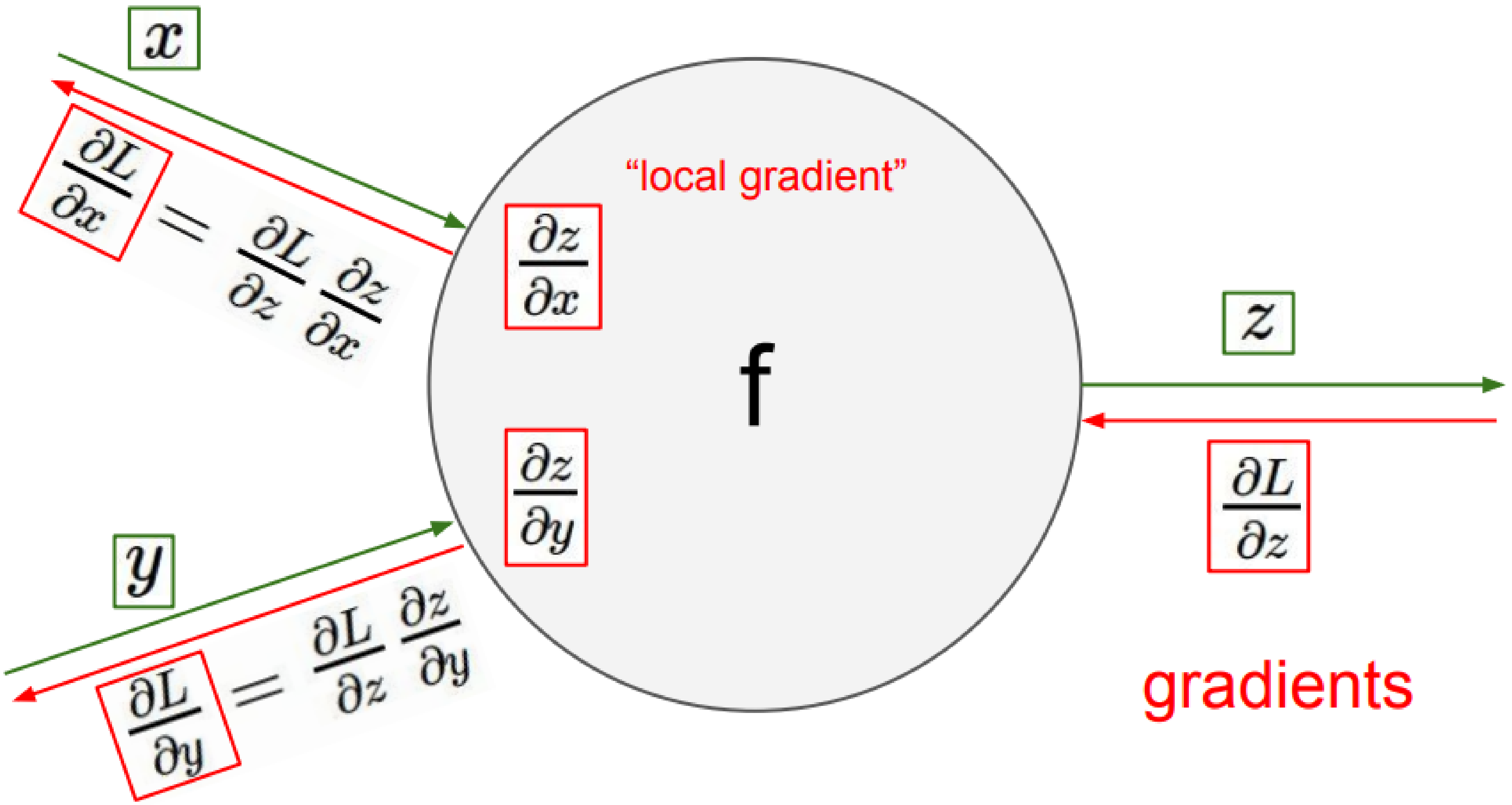
The learning rules in different biologically plausible models can be implemented with different types of spike-time-dependent plasticity.

Deep Backpropagation

看似複雜但每個映射只要管好計算自己的梯度即可

可視為非線性網路的 δ_j

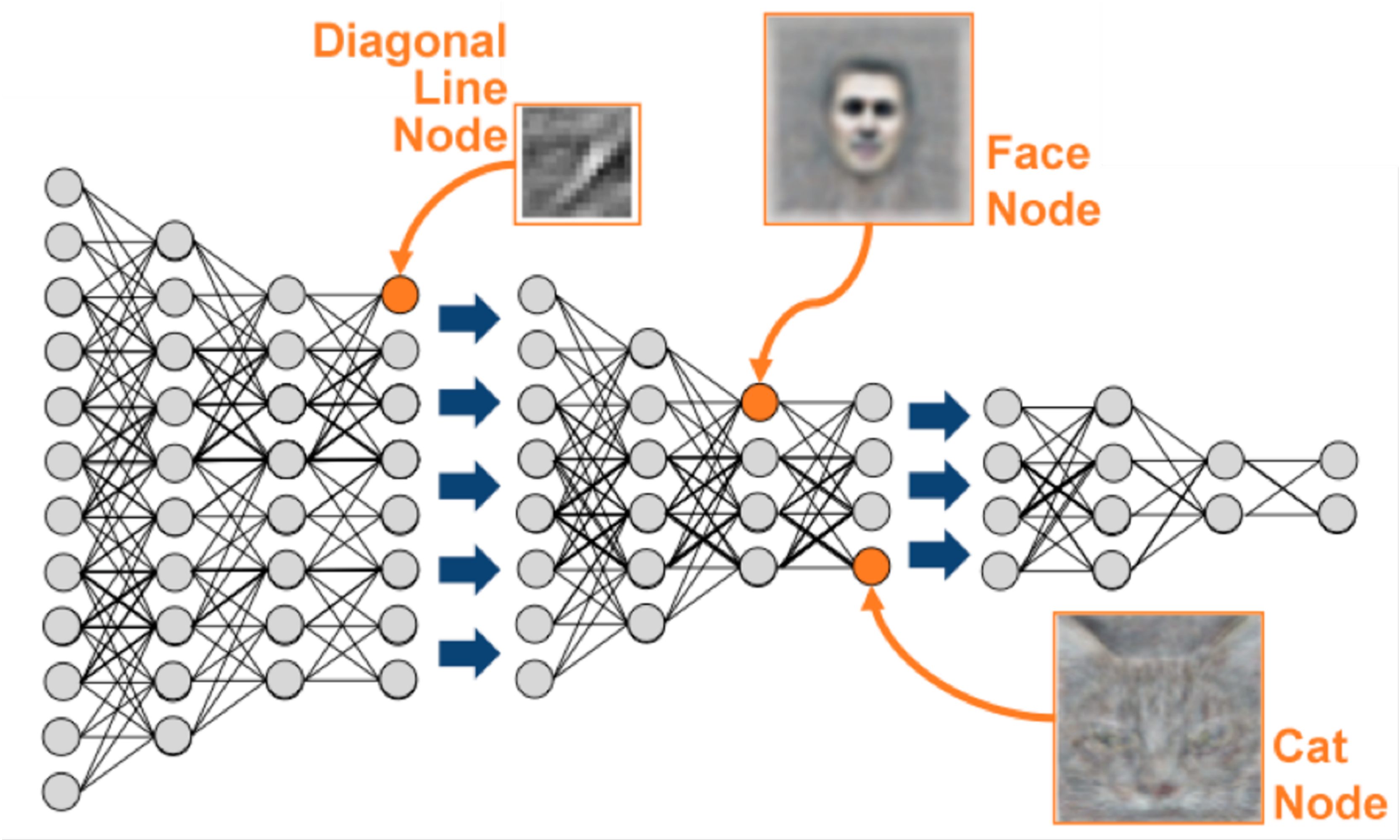
$$\Delta w_{ij} = x_i \delta_j f'(h_j) = x_i \underline{(t_j - y_j) f'(h_j)}$$



x
$\frac{\partial L}{\partial x} = \frac{\partial L}{\partial z}\frac{\partial z}{\partial x}$
"local gradient"
$\frac{\partial z}{\partial x}$
f
$\frac{\partial z}{\partial y}$
z
$\frac{\partial L}{\partial z}$
y
$\frac{\partial L}{\partial y} = \frac{\partial L}{\partial z}\frac{\partial z}{\partial y}$
gradients

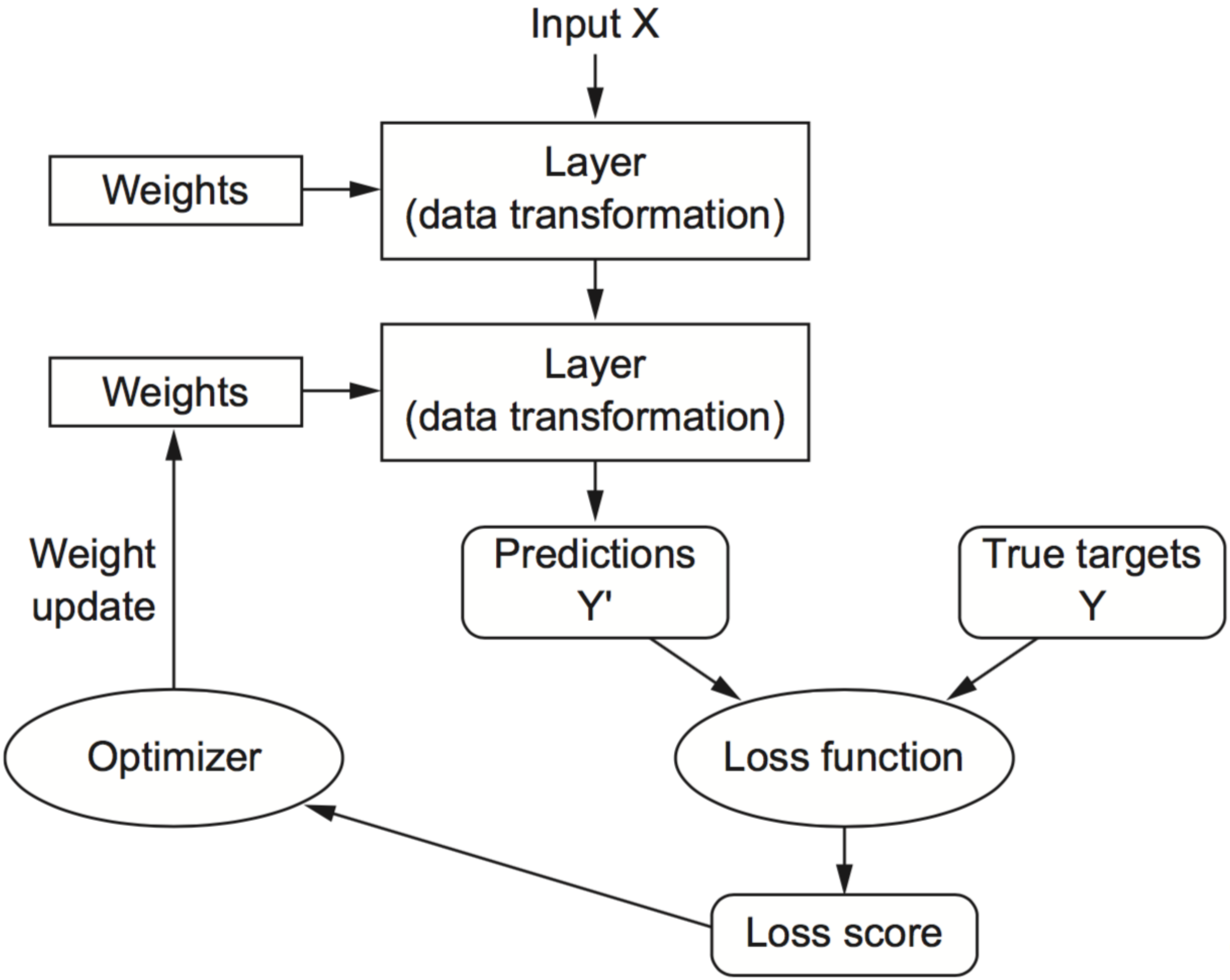
Deep Neural Network

DNN 只是層數比較多的 Neural Net



Deep Supervised Learning

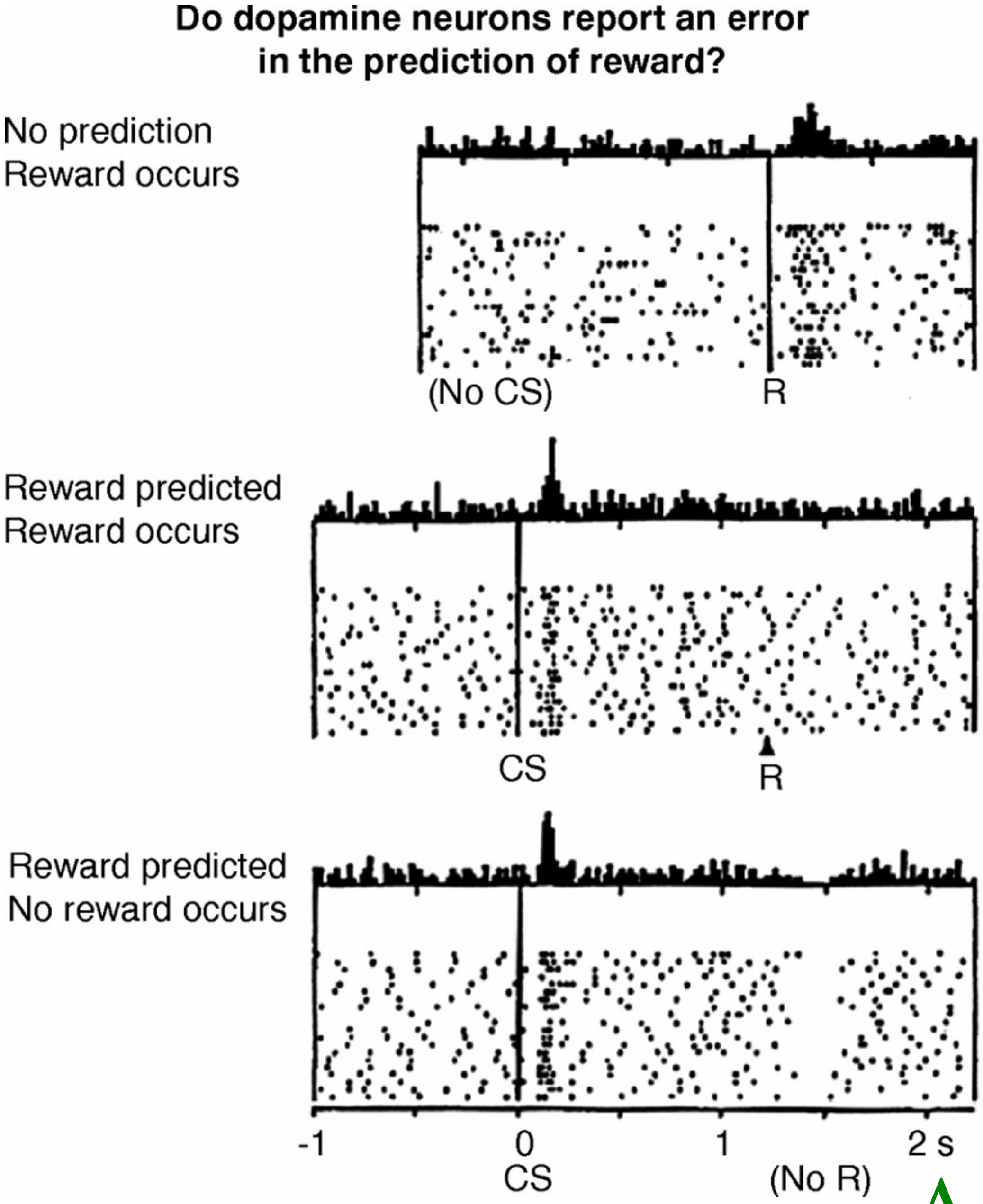
和 Shallow Supervised Learning 程序一樣



Supervised Learning

Reinforcement Learning

獎賞與多巴胺 (Dopamine)



古典制約學習
可看成是學 S-S 連結
或是刺激替代

注意左圖神經元反應的
是獎賞的預測錯誤
而非獎賞本身

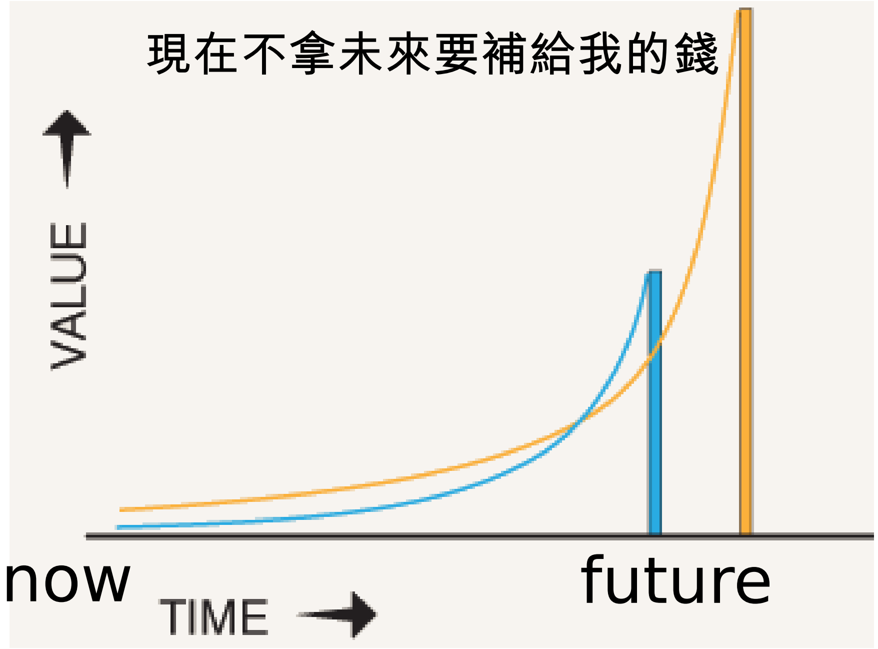
DA dip=R-R̂
而 r 的預測可透過
temporal δ rule 學習

$$\hat{R}^{t+1}=\sum_i W_i^t S_i^t$$

$$\Delta W_i=S_i^t \delta^{t+1}=S_i^t\left(R^{t+1}-\hat{R}^{t+1}\right)$$

更精確的獎賞估計應考慮長遠的未來

沒帶來立即獎賞的 CS 其實預測之後的各種獎賞



獎賞的時間離當下
愈遠價值愈低
(temporal discounting)

$$\hat{V}^{t+1} = \sum_i W_i^t S_i^t$$

$$\Delta W_i = S_i^t \left(V^{t+1} - \hat{V}^{t+1}\right) = S_i^t \delta V^{t+1}$$

但 V 要考慮所有時間的 r:

$$V^{t+1} = R^{t+1} + \gamma R^{t+2} + \gamma^2 R^{t+3} \dots$$

若 γ=0 回到上頁公式
且上式可改寫成遞迴形式：

$$V^{t+1} = R^{t+1} + \gamma \left(R^{t+2} + \gamma R^{t+3} + \dots\right) = R^{t+1} + \gamma V^{t+2}$$

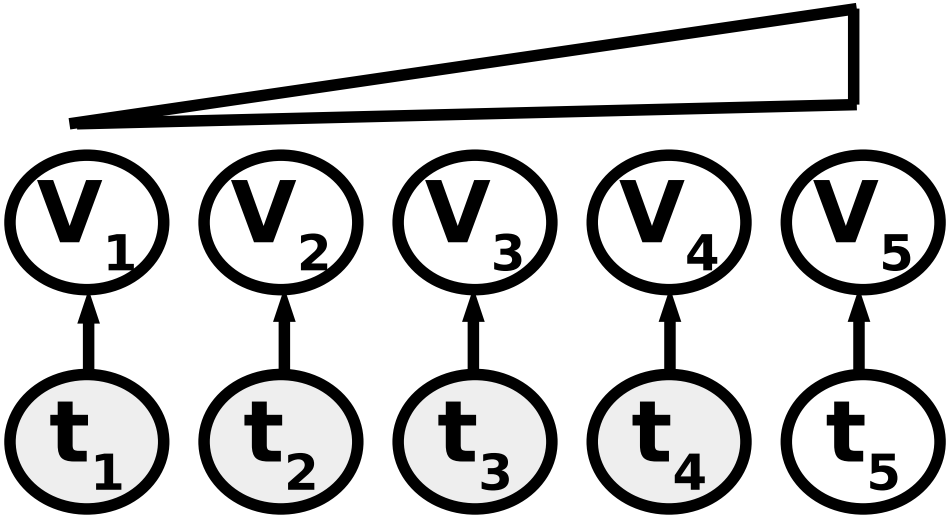
預測 V 的學習目標：等式成立

學習過程：讓 $\delta V \equiv R^{t+1} + \gamma \hat{V}^{t+2}\left(S^{t+1}\right) - \hat{V}^{t+1}\left(S^{t}\right) \rightarrow 0$

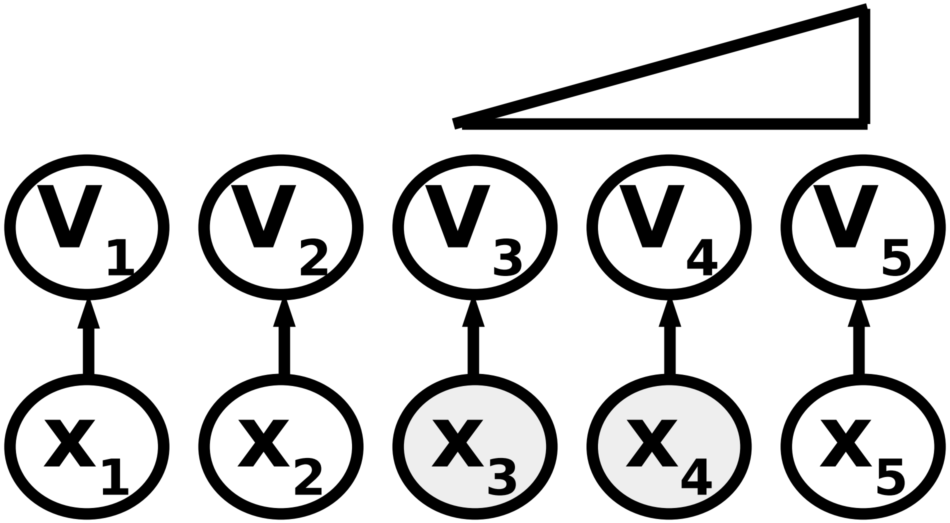
古典制約的 TD Learning

用現在的 R_t 來調上一個時間點的 $V(S_{t-1})$

TD Learning $$V(S_t) \leftarrow \underbrace{V(S_t)}_{\text{Previous estimate}}+\alpha[\underbrace{\underbrace{R_{t+1}}_{\text{Reward t+1}}+\underbrace{\gamma V(S_{t+1})}_{\text{Discounted value on the next step}}}_{\text{TD Target}}-\underline{V(S_t)}]$$ S_t



若 S=t 則 V(t) 變為
無邊界的時間折價函數
（如無刺激出現但
固定 5 秒後出現酬賞）



若 S=x(t) 則 V(x) 變為
有邊界的時間折價函數
（如有刺激出現，且之後
固定 2 秒後出現酬賞）

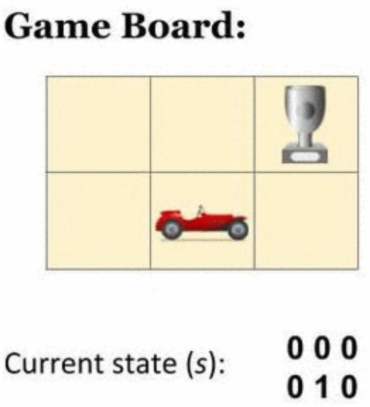
操作制約的 Q-Learning (1/2)

Q(s,a) 是狀態 s 與行動 a 的函數

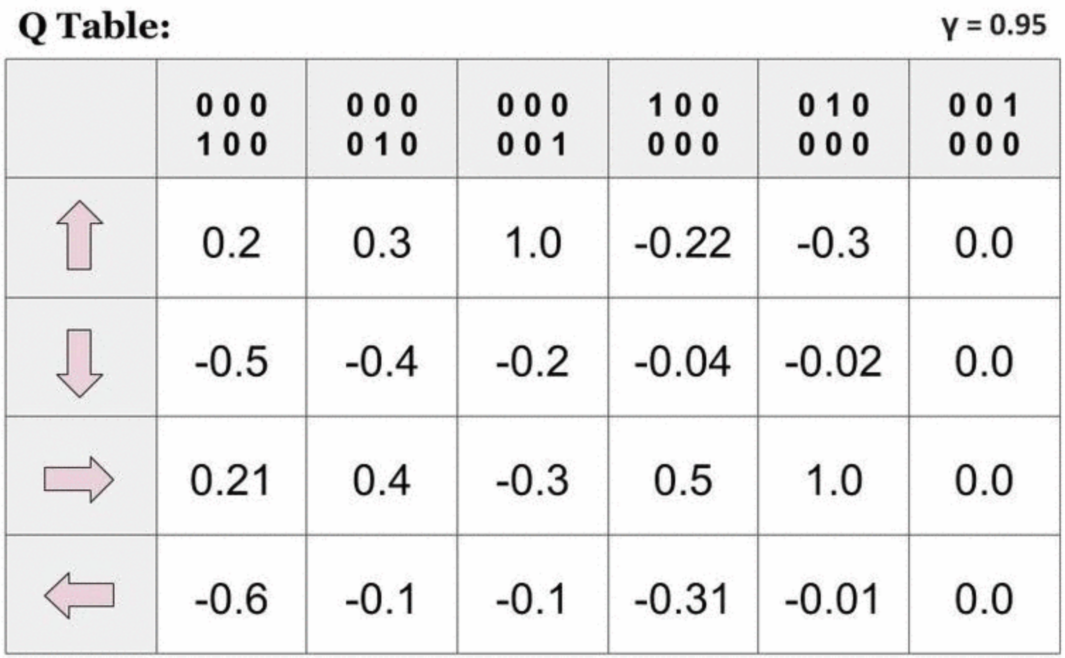
$$Q_{t+1}(s_t,a_t) = \underbrace{Q_t(s_t,a_t)}_{\text{old value}} + \underbrace{\alpha_t(s_t,a_t)}_{\text{learning rate}} \cdot \left(\overbrace{\underbrace{R_{t+1}}_{\text{reward}} + \underbrace{\gamma}_{\text{discount factor}} \underbrace{\max_a Q_t(s_{t+1},a)}_{\text{estimate of optimal future value}}}^{\text{learned value}} - \underbrace{Q_t(s_t,a_t)}_{\text{old value}} \right)$$

在 t 時要選什麼動作變成下一個狀態要根據既定 policy

計算上常離散化 s 與 a 來做成一個 Q(s,a) 的查找表


Game Board:

Current state (s): 000
010


Q Table:

γ = 0.95

	000 100	000 010	000 001	100 000	010 000	001 000
↑	0.2	0.3	1.0	-0.22	-0.3	0.0
↓	-0.5	-0.4	-0.2	-0.04	-0.02	0.0
→	0.21	0.4	-0.3	0.5	1.0	0.0
←	-0.6	-0.1	-0.1	-0.31	-0.01	0.0

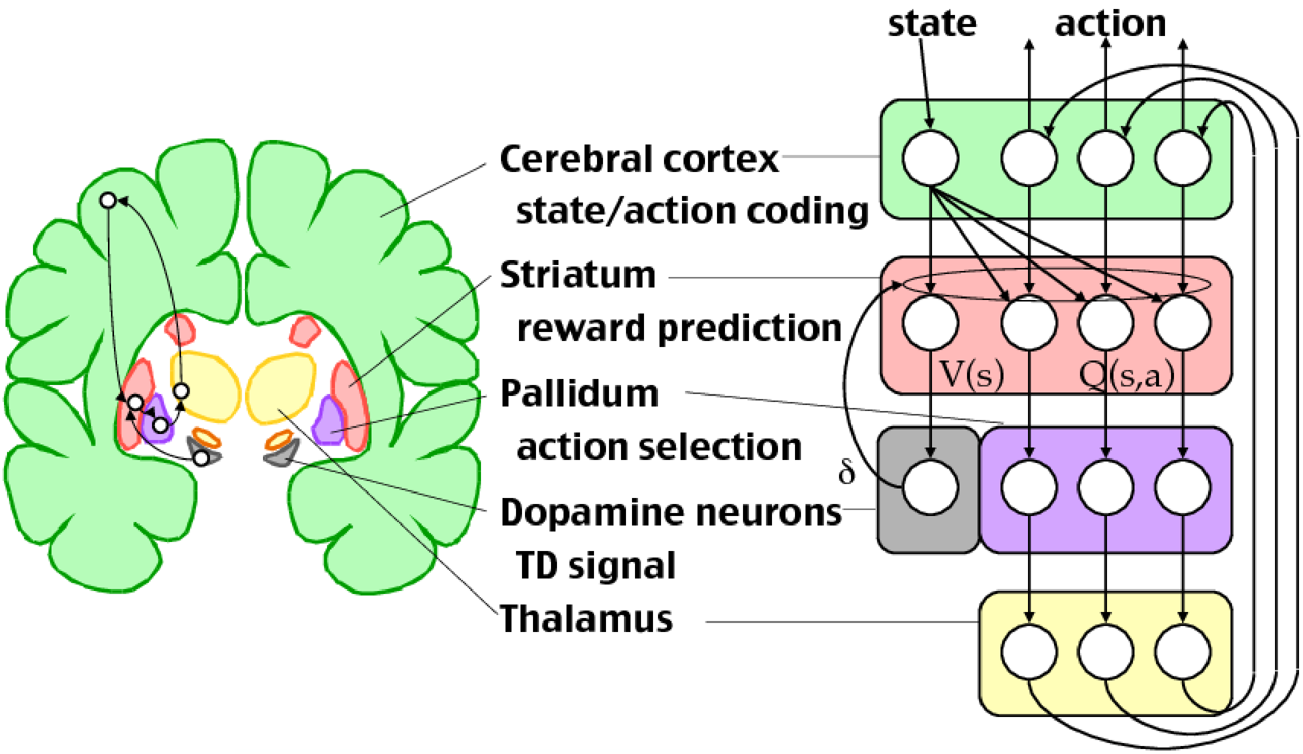
操作制約的 Q-Learning (2/2)

Q(s,a) 是狀態 s 與行動 a 的函數

$$Q_{t+1}(s_t,a_t) = \underbrace{Q_t(s_t,a_t)}_{\text{old value}} + \underbrace{\alpha_t(s_t,a_t)}_{\text{learning rate}} \cdot \left(\overbrace{\underbrace{R_{t+1}}_{\text{reward}} + \underbrace{\gamma}_{\text{discount factor}} \underbrace{\max_a Q_t(s_{t+1},a)}_{\text{estimate of optimal future value}}}^{\text{learned value}} - \underbrace{Q_t(s_t,a_t)}_{\text{old value}} \right)$$

在 t 時要選什麼動作變成下一個狀態要根據既定 policy

計算上常離散化 s 與 a 來做成一個 Q(s,a) 的查找表



state
action
Cerebral cortex
state/action coding
Striatum
reward prediction
Pallidum
action selection
Dopamine neurons
TD signal
Thalamus

Game Over