Wide & Deep Model

Agenda

1. Linear Model

Memorization

Back Propagation

遇到問題

2. Deep Model

Embedding Based

Generalization

Gradient Vanishing

遇到問題

3. Wide & Deep

Joint Training

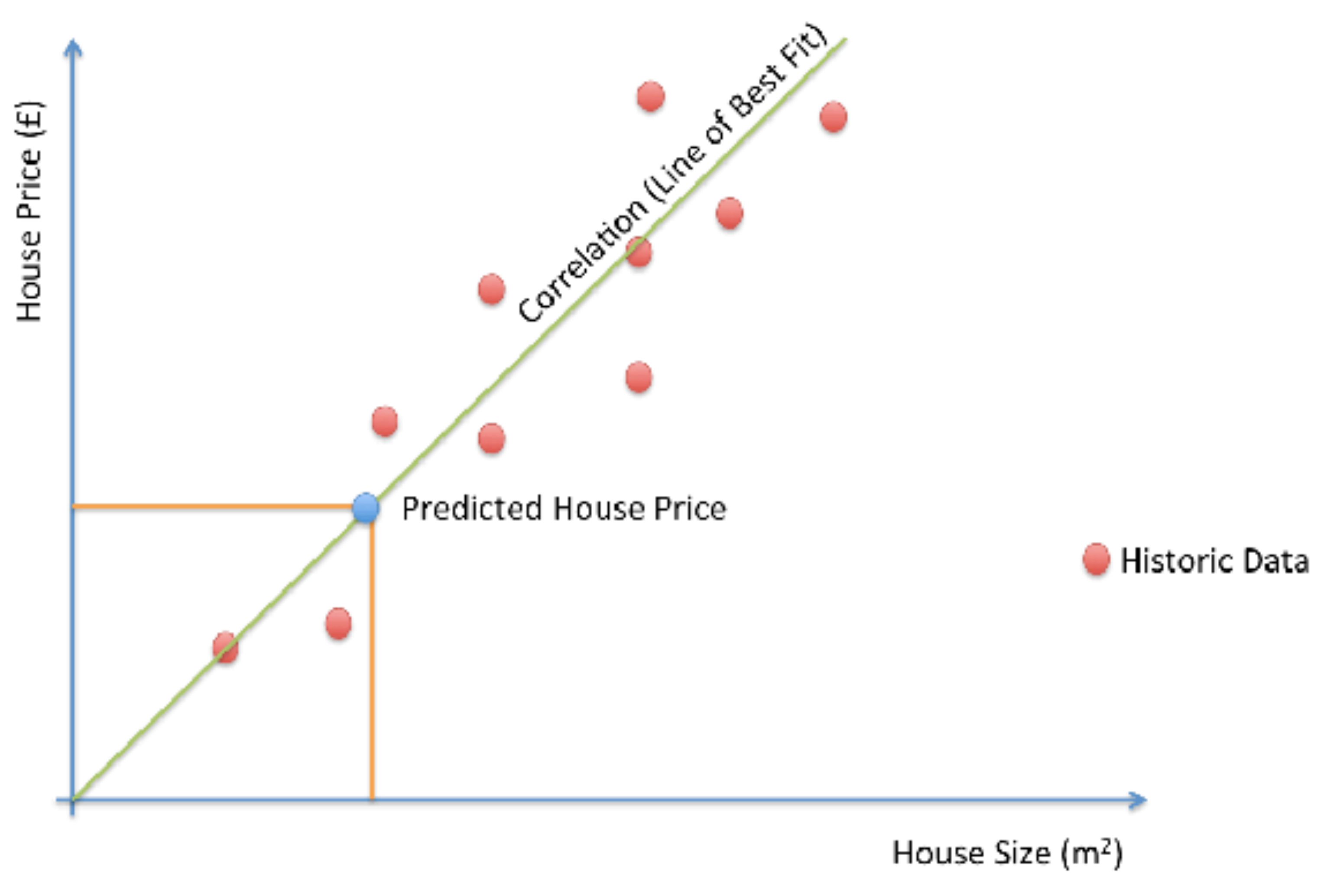
未來方向

4. Appendix

參數比較

Linear Model

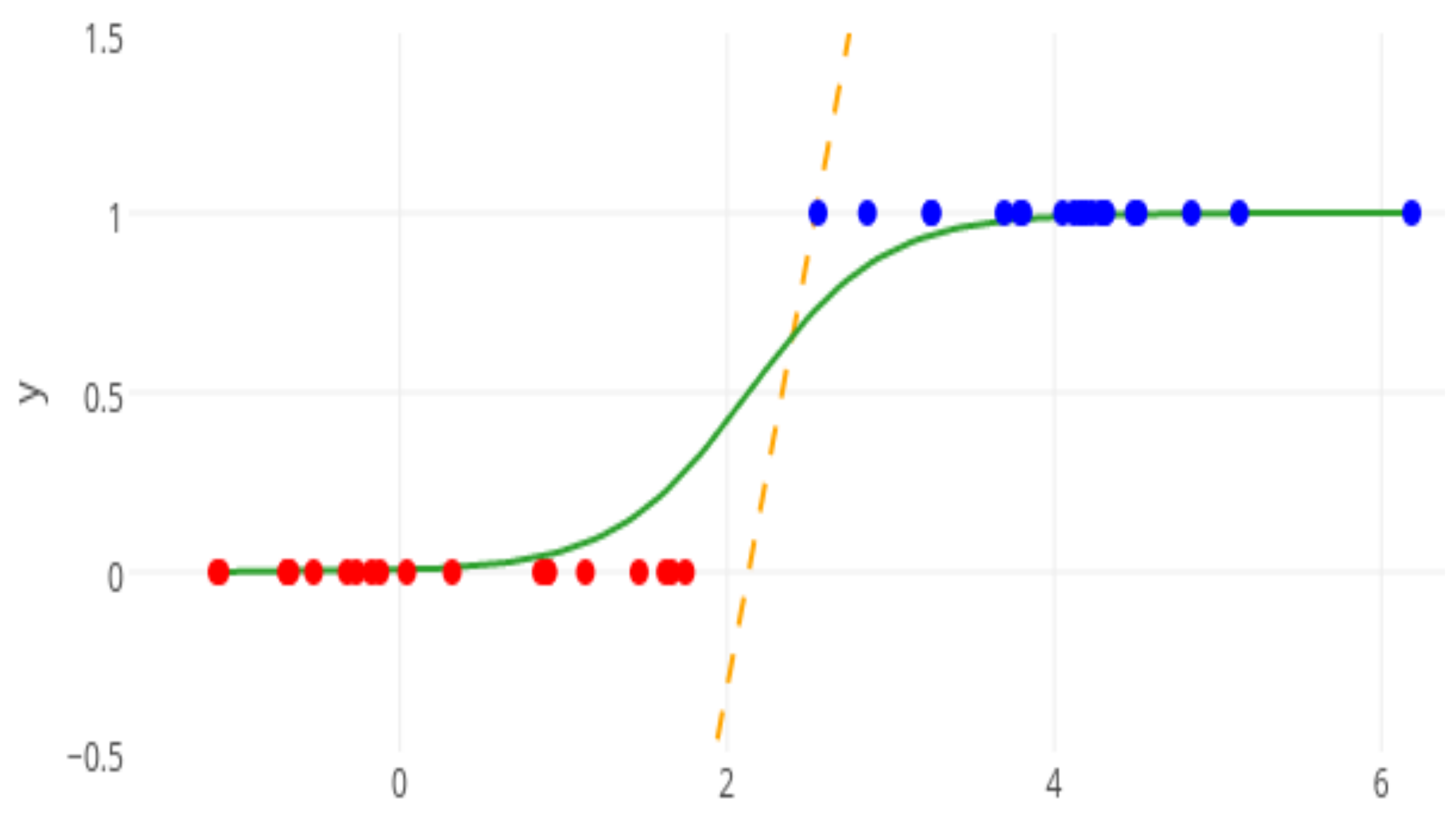
Linear Regression



$$y = b_0 + b_1 x$$

預測數值

Logistic Regression



$$p = \frac{1}{1 + e^{-(b_0 + b_1 x)}}$$

預測機率(二分類)

Logistic Regression

回憶：預測Betty會不會生氣(若為正義魔人)

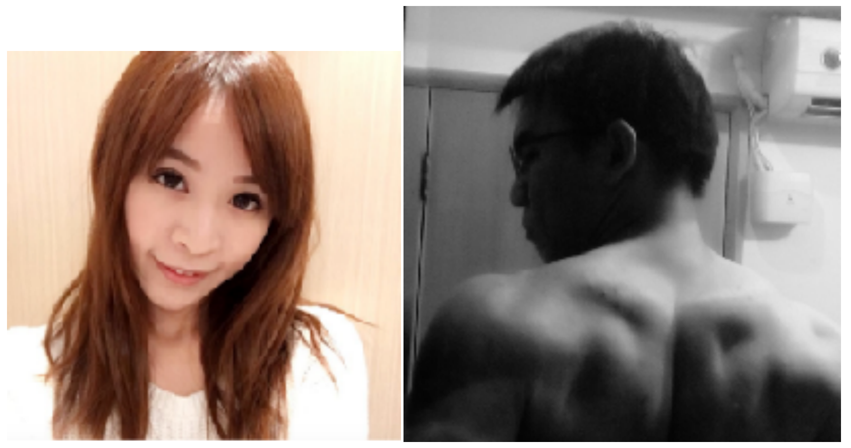
特徵因子可能有單身、跟女生牽手...

如果想結合因子呢 ！？

Multivariate Mutual Information

無關

Y X1



Betty會生氣 Johnson不是單身

無關

Y X2

Betty會生氣Johnson跟女生牽手

有關

Y X1 X2

Betty會生氣 Johnson不是單身 Johnson跟女生牽手

↓

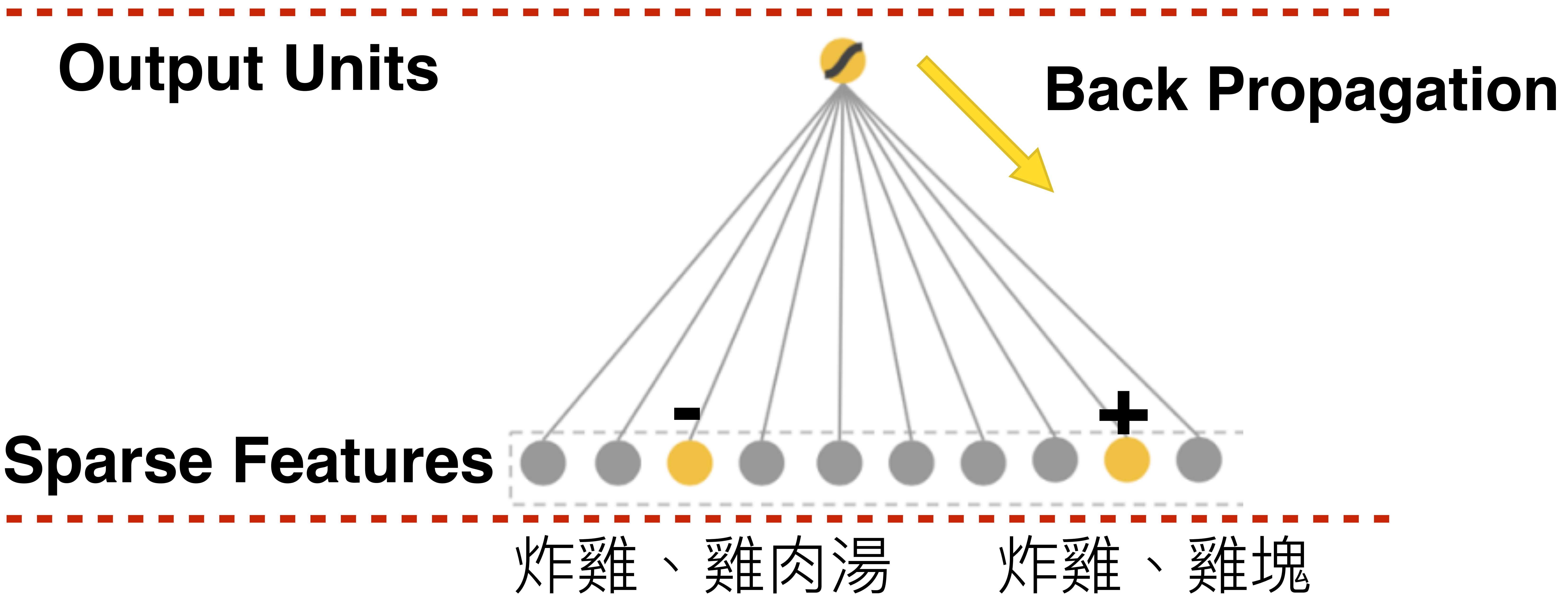
Johnson不是單身、Johnson跟女生牽手產生綜效

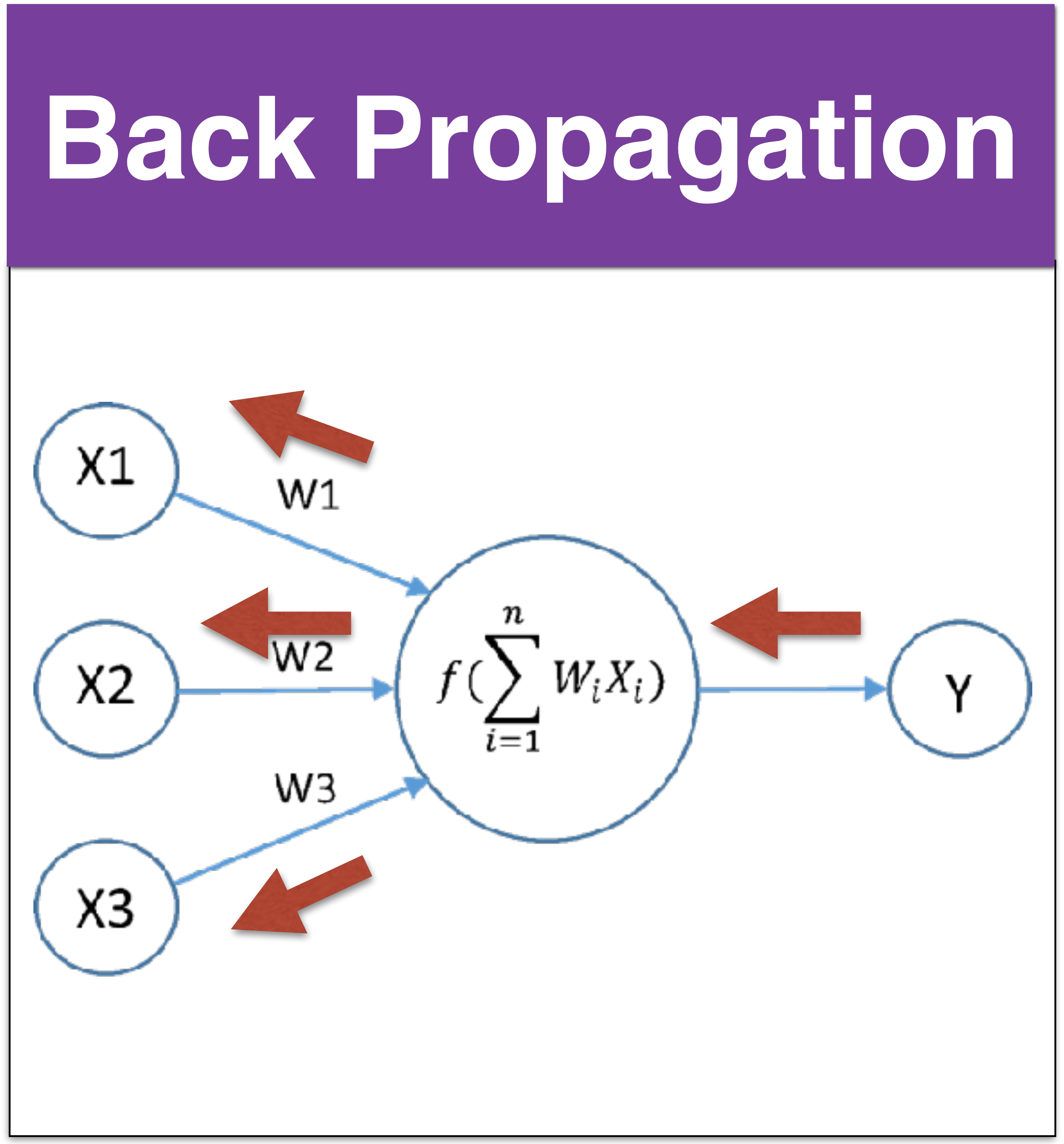
Memorization

表面關係

透過Crossed Column

預測：吃炸雞的還會吃什麼

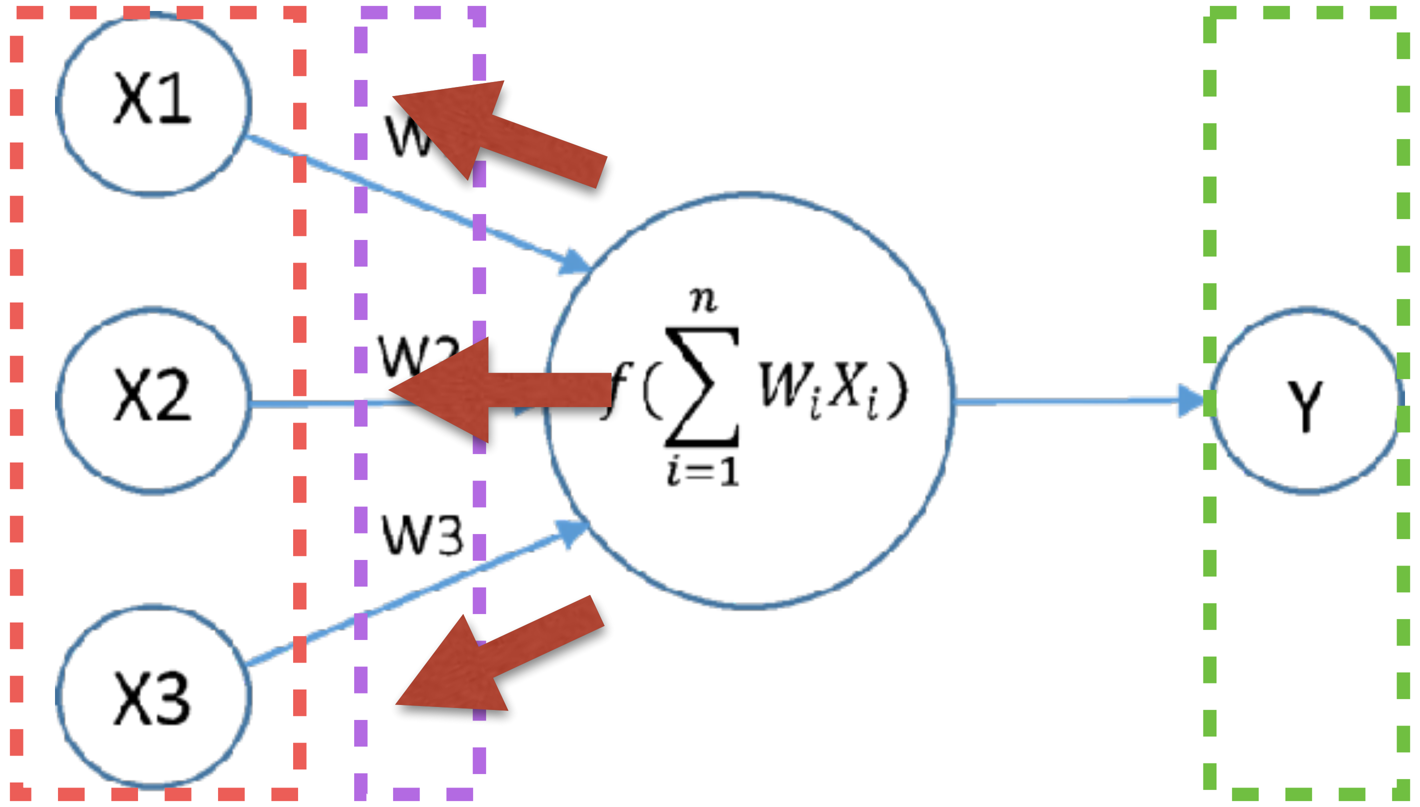




Back Propagation

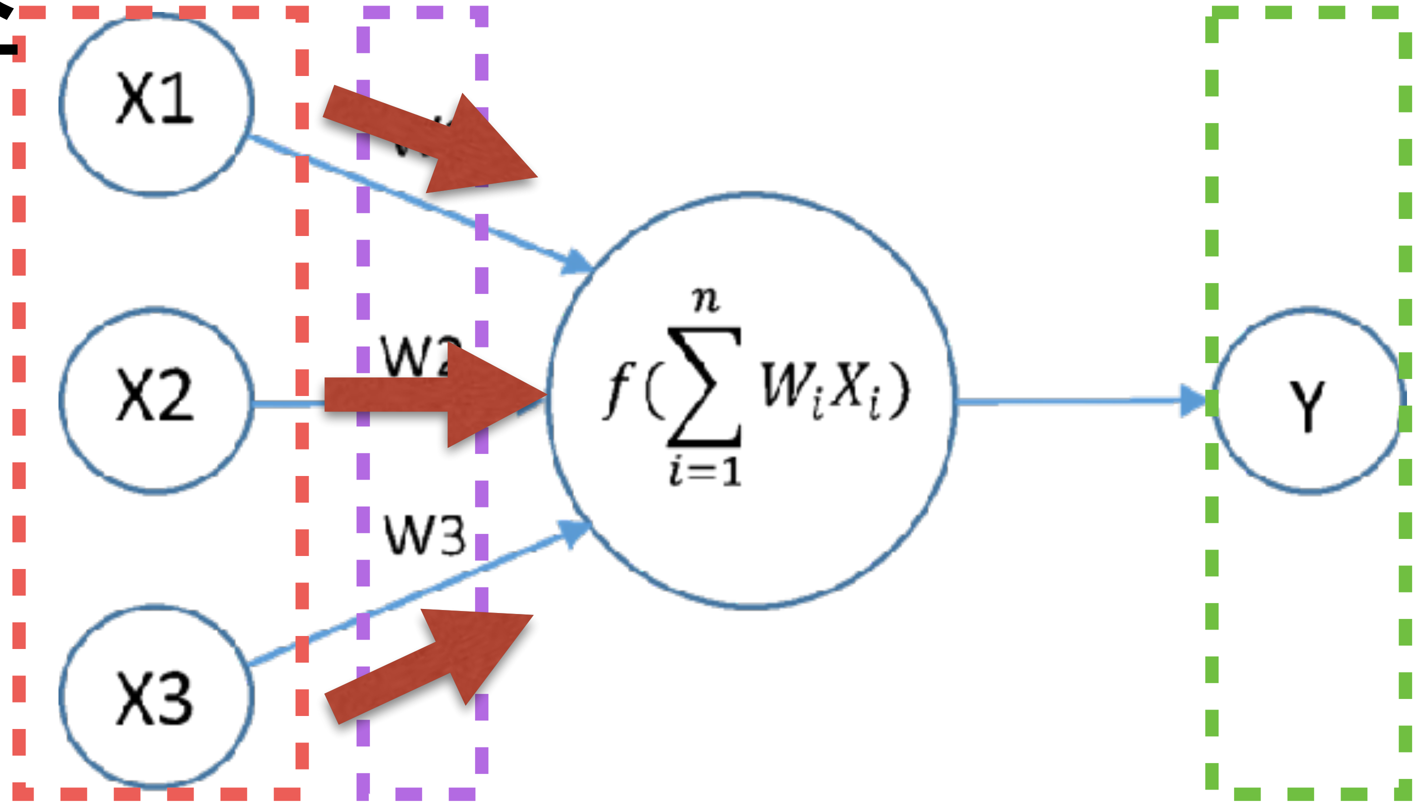
<u>Gradient</u>=Error·Sigmoid′(x)·x

學習階段



已知　未知　已知

回想階段



已知　已知　未知

大美女采襄

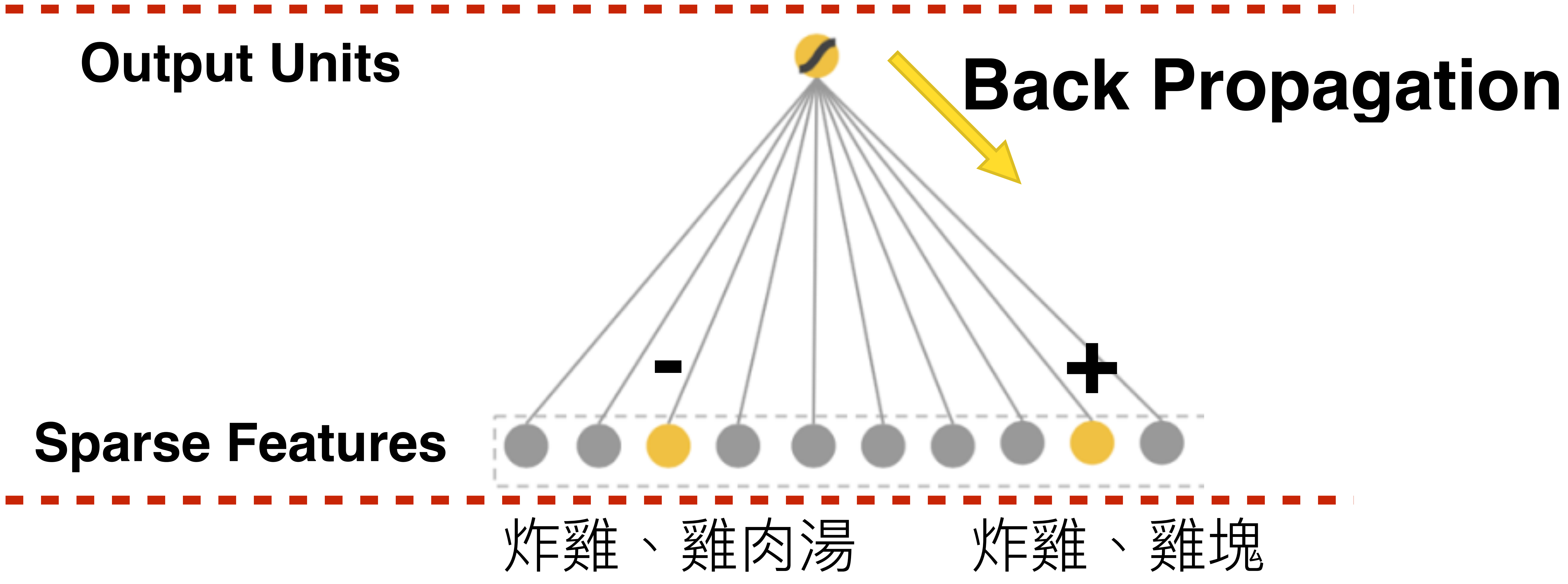
小結

1. BP為監督式學習，需有輸入特徵、目標結果
2. 輸入層神經元＝輸入特徵
   輸出曾神經元＝結果
3. 分為 學習階段、回想階段

<u>Linear Model</u>

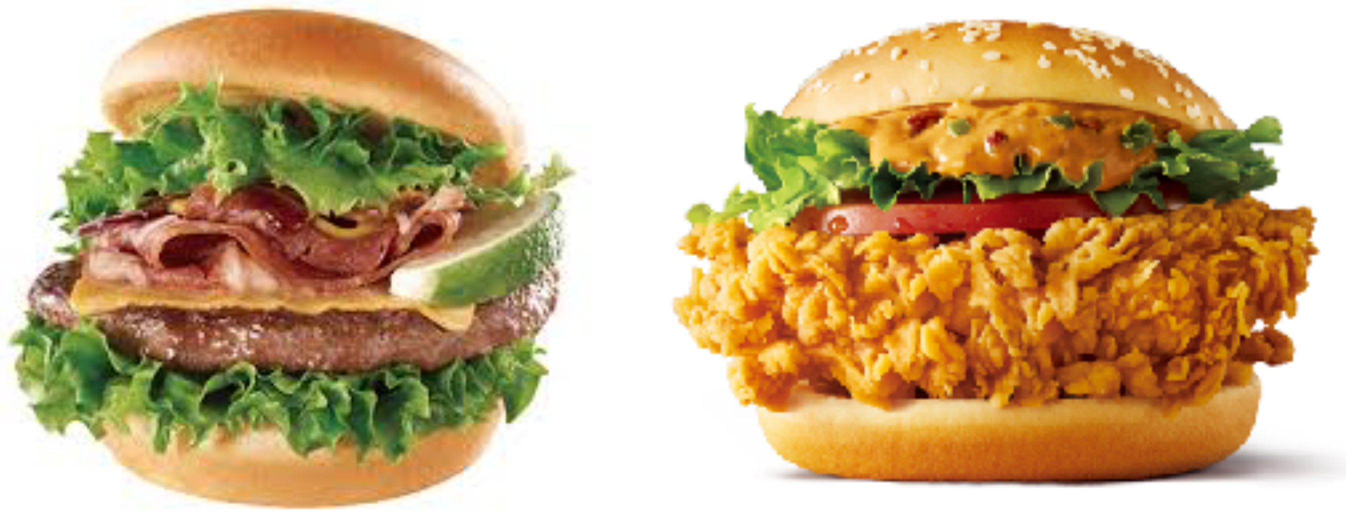
<u>遇到問題</u>

真相往往不是表面所看到的！



無法Generalization

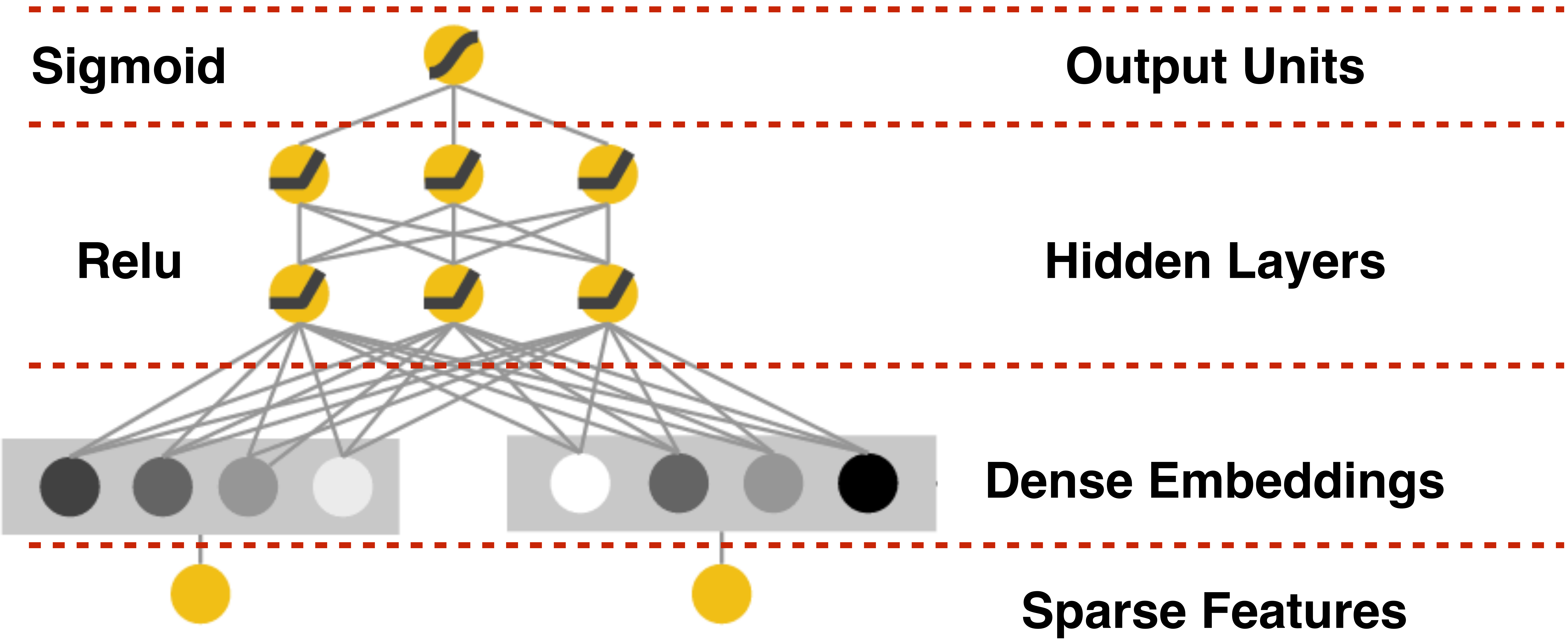
IF：訓練資料沒有



→ 無法習得技能

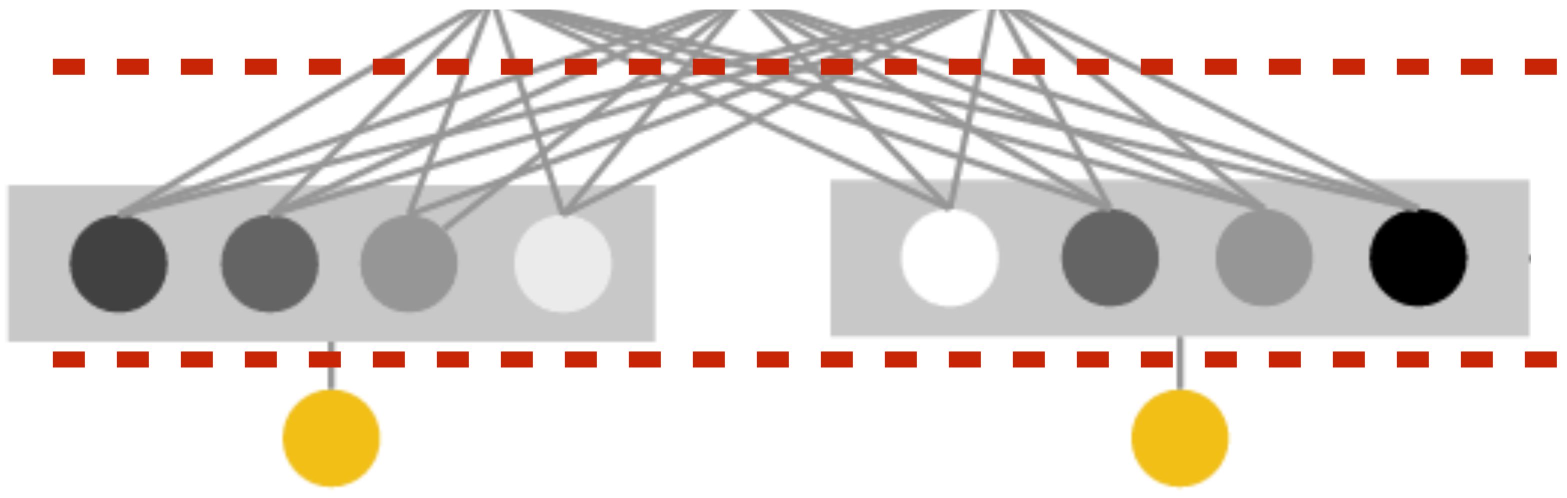
<u>解法 ：Deep Model</u>

Deep Model


Sigmoid
Output Units
Relu
Hidden Layers
Dense Embeddings
Sparse Features

回憶：Word Embedding

冠穎



Dense Embeddings

Sparse Features

word-context 矩陣 **co-occurrence** **context: n=1**

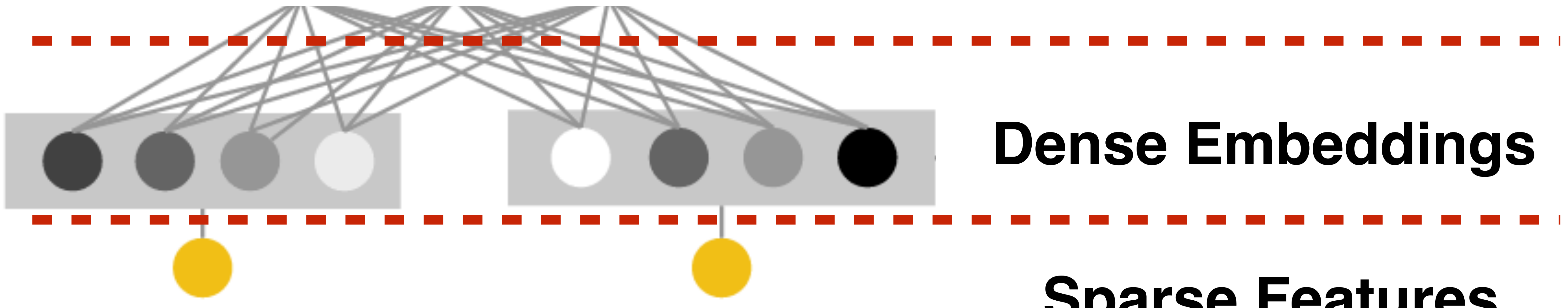
The dog run.
The cat run.
The dog sleep.
The cat sleep.
The dog bark.
The cat meows.
The bird fly.
The bird sleep.

	The	dog	cat	bird	run	sleep	bark	meows	fly
The	0	3	3	2	0	0	0	0	0
dog	3	0	0	0	1	1	1	0	0
cat	3	0	0	0	1	1	0	1	0
bird	2	0	0	0	0	1	0	0	1
run	0	1	1	0	0	0	0	0	0

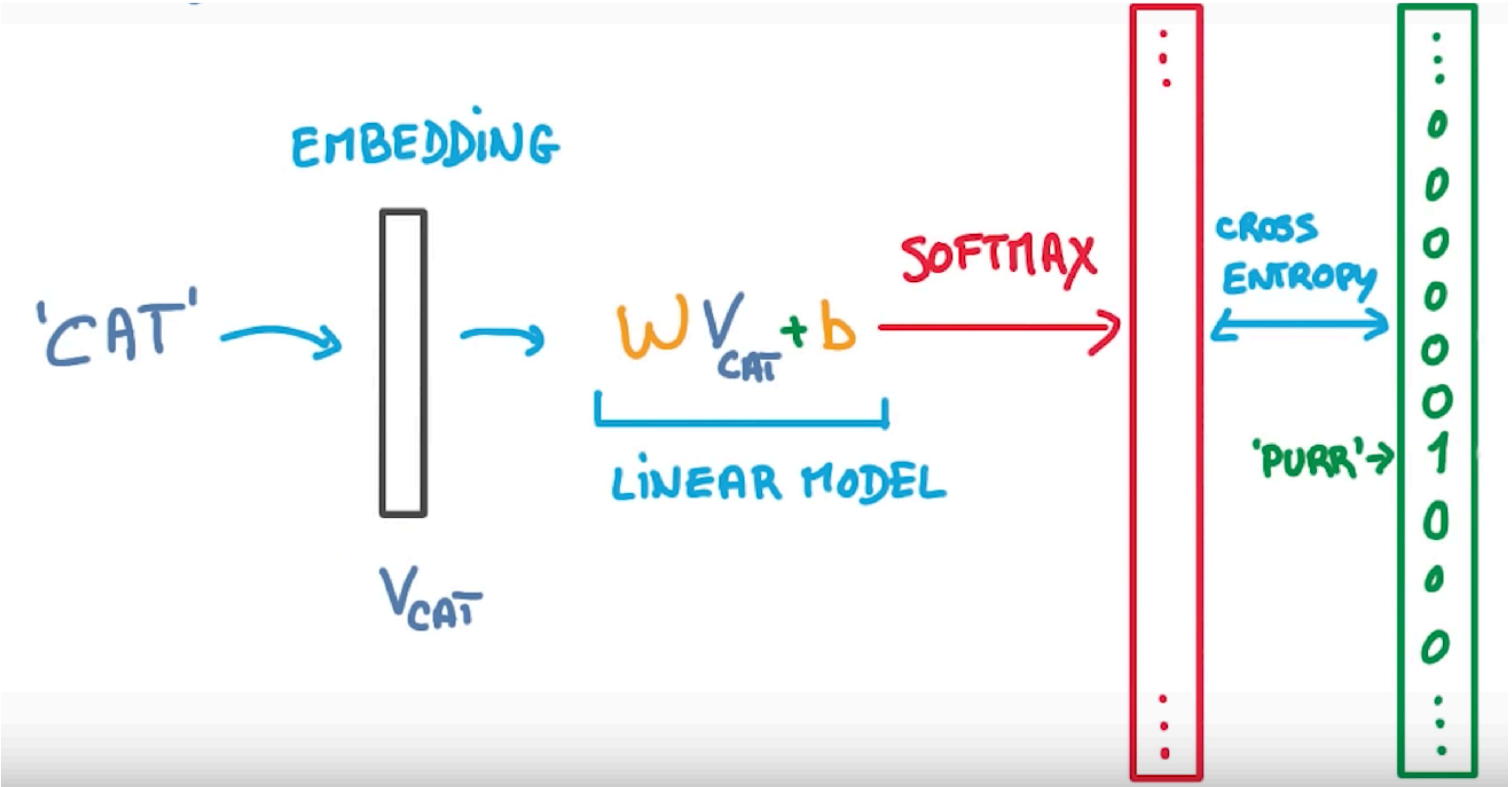
由於 dog 和 cat 這兩個字出現在類似的上下文情境中，因此可以判斷出 dog 和 cat 語意相近。

→ **cosine similarity**

回憶：Embedding



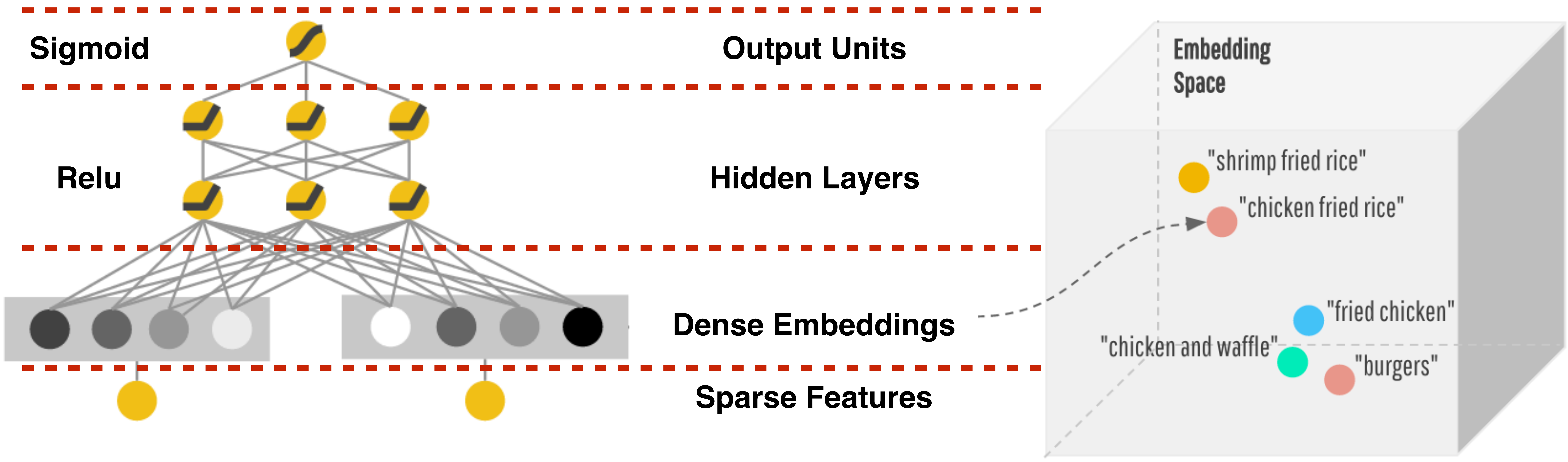
IF：預測CAT和某詞彙的關聯



Generalization

潛在關係

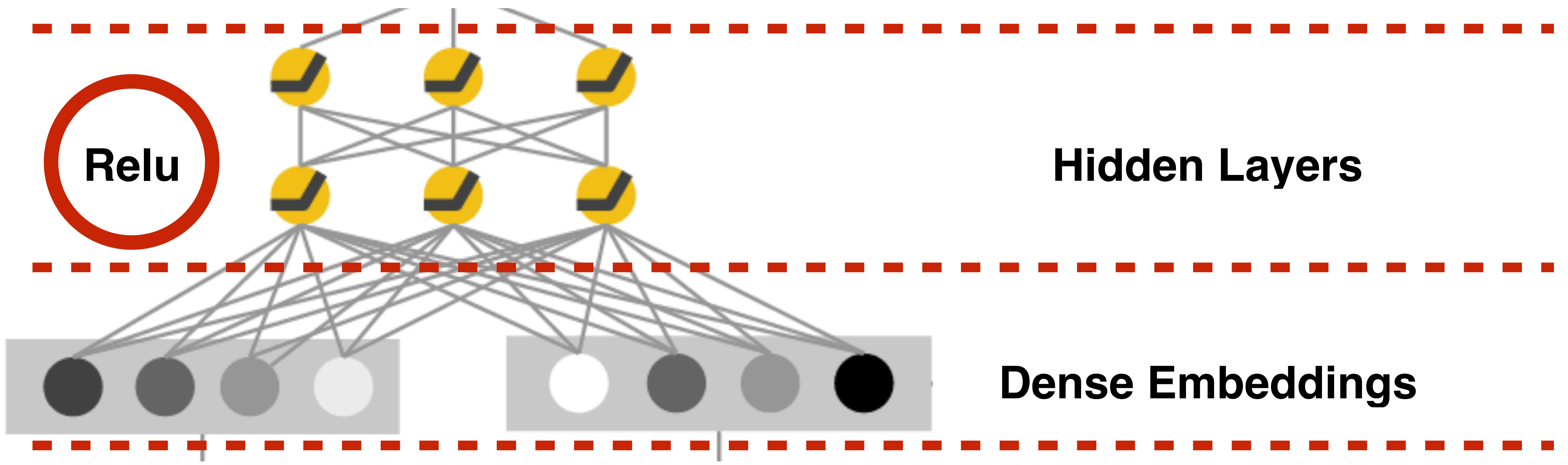
透過Dense Embeddings



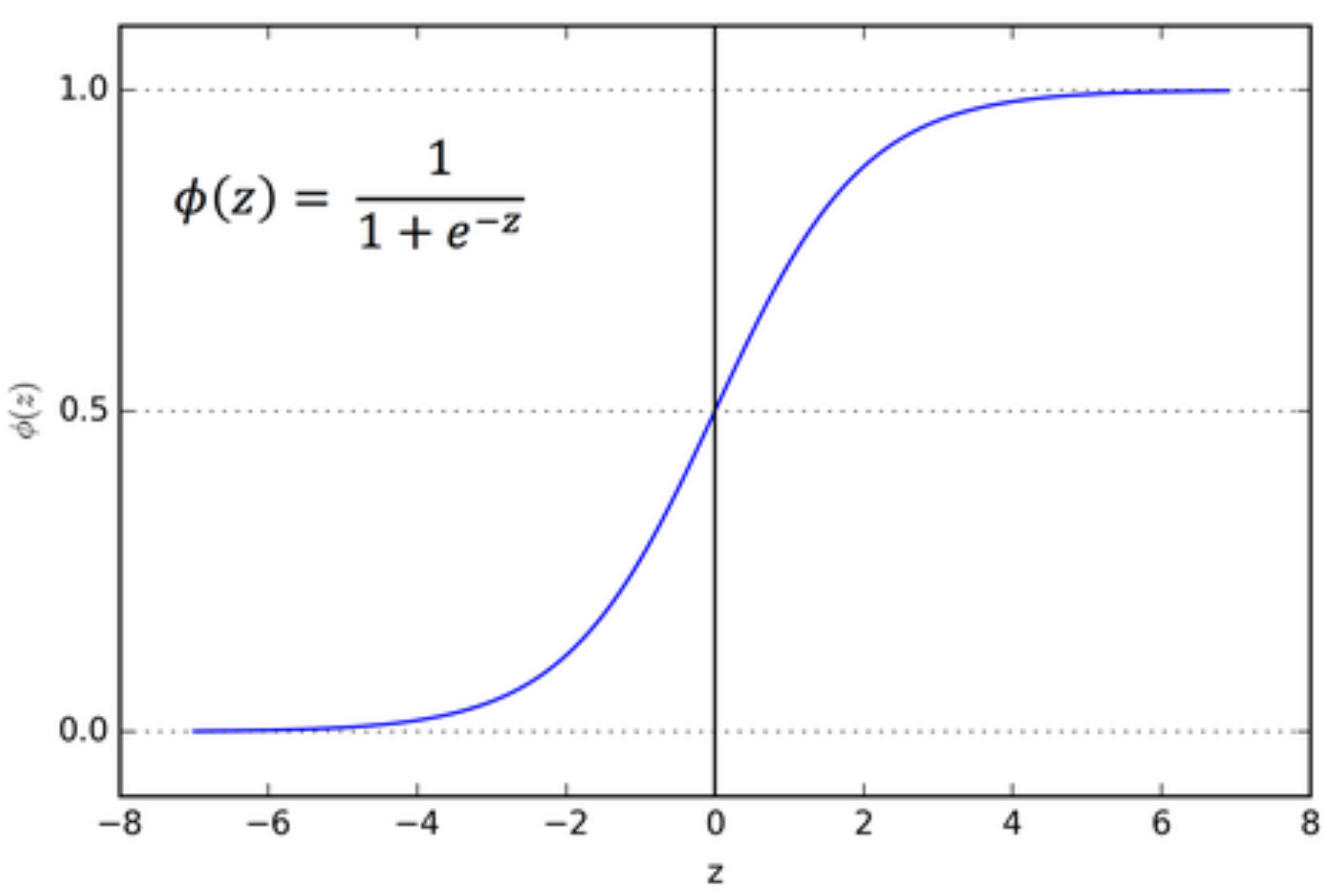
吃炸雞的人可能也想吃漢堡 ！！

Sigmoid vs Relu

Gradient = Error · Sigmoid′(x) · x

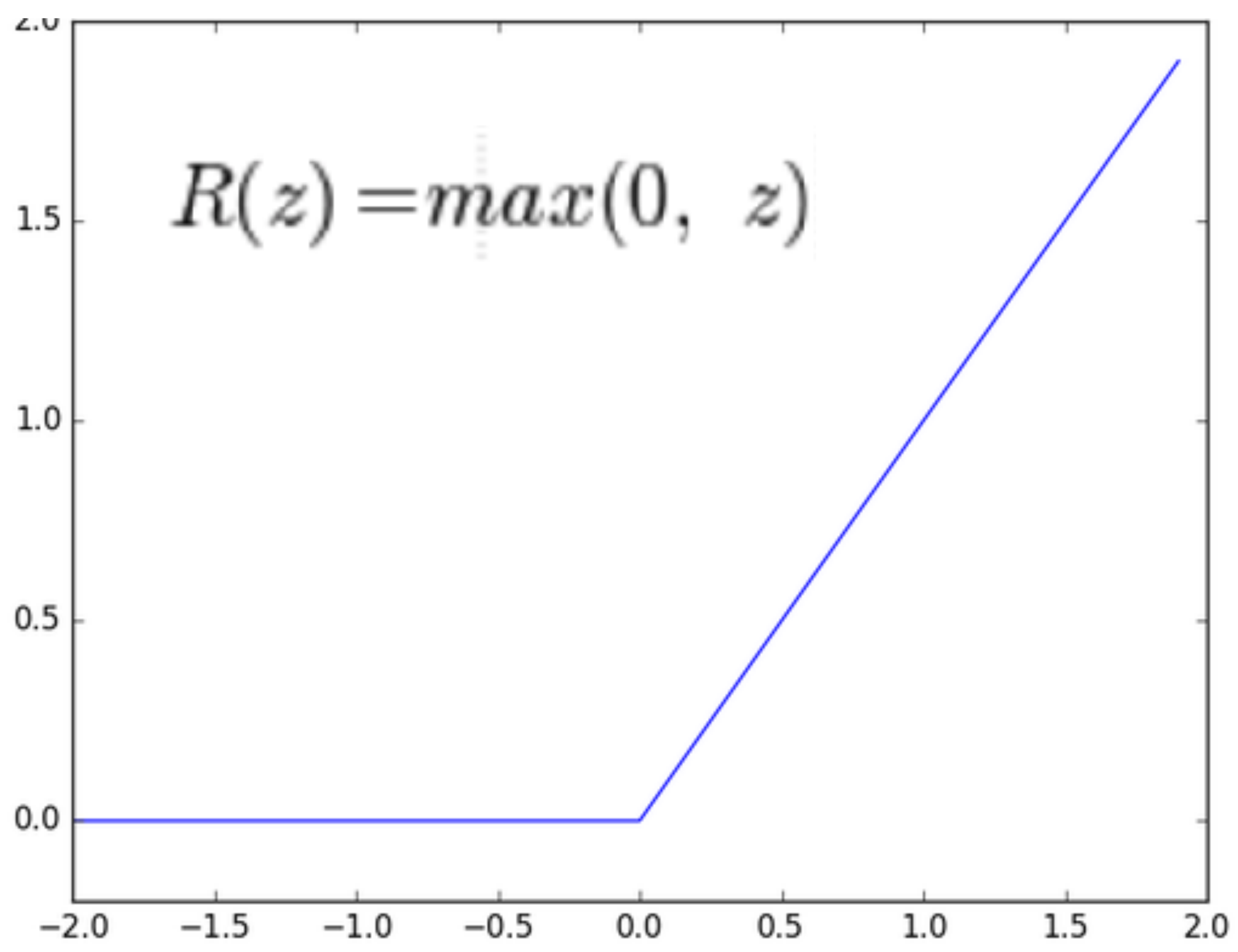


Sigmoid



容易Gradient Vanishing

Relu



部分0 -> 解决overfitting

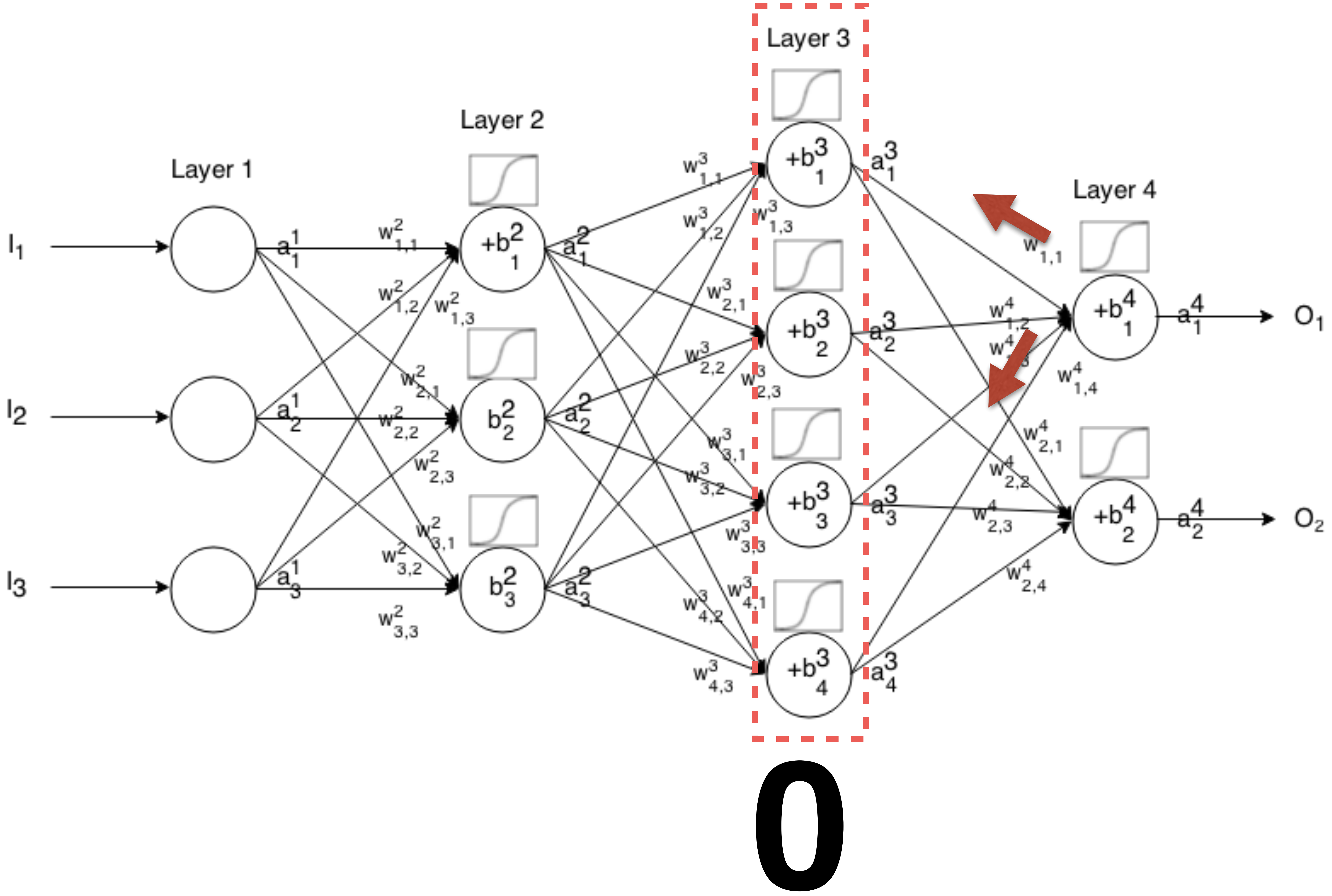
「然後他就死掉了。」

—— 羅瑩雪

Gradient Vanishing

然後他就死掉了

Gradient=Error·Sigmoid′(x)·x



反向

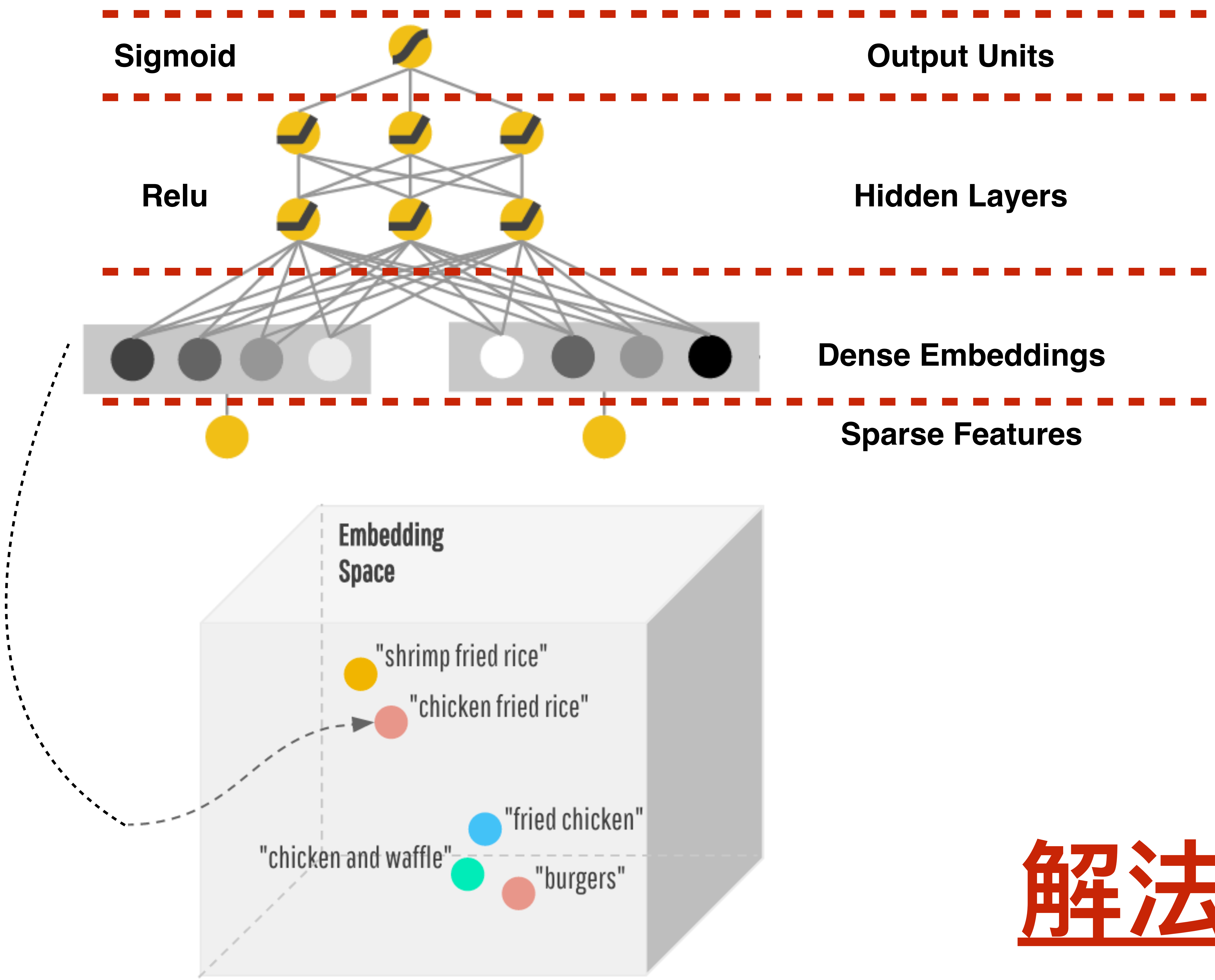
IF 誤差在第3層為0
第1、2層形同虛設

正向

第1、2層亂七八糟
怎麼訓練都很差

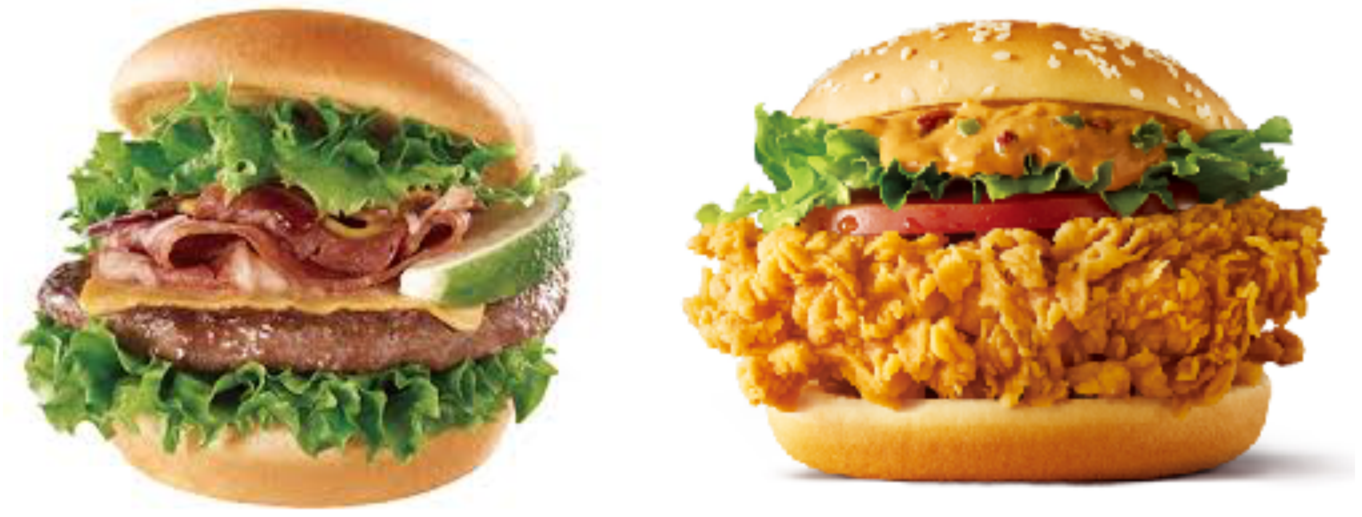
Deep Model 遇到問題

Too good to be true



過度Generalization

Dense到極致



→ 無關變有關

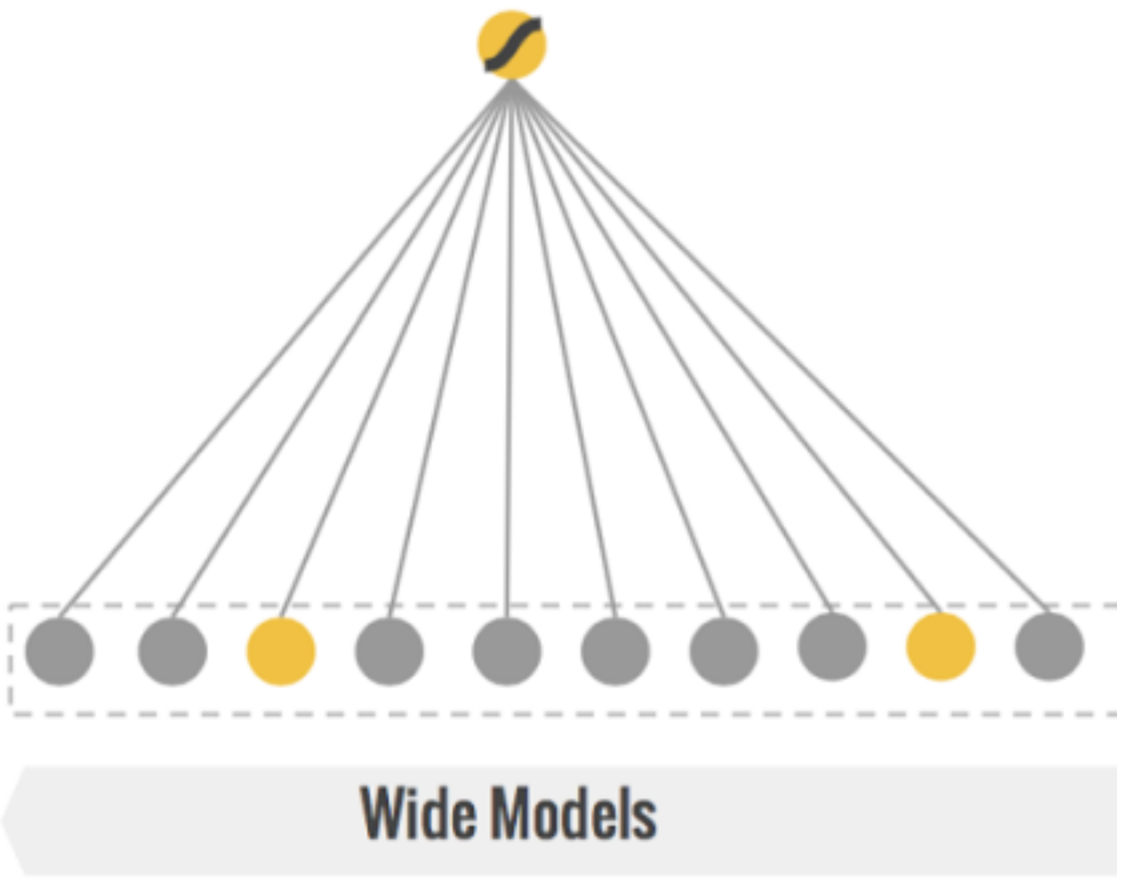
解法 ：Linear Model

How About

Wide

優點

Memorization

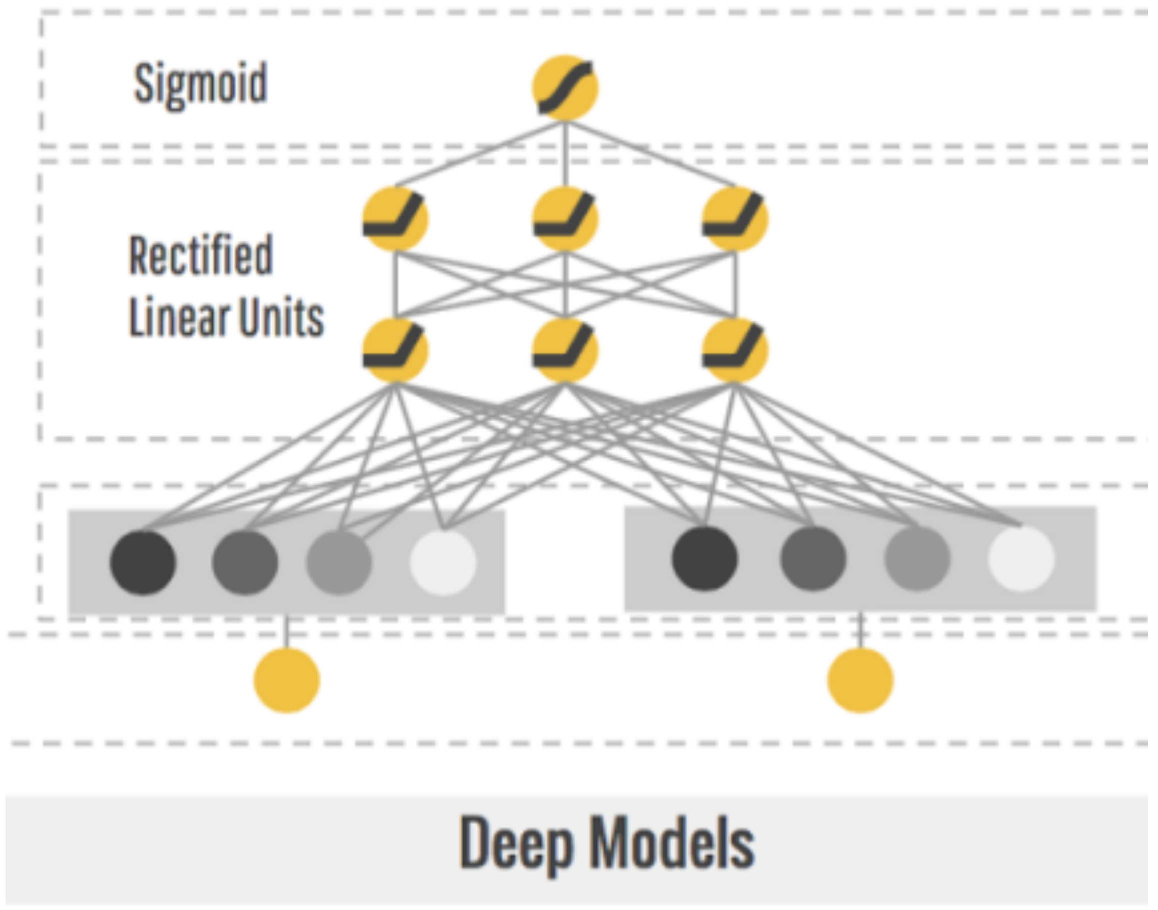


+

Deep

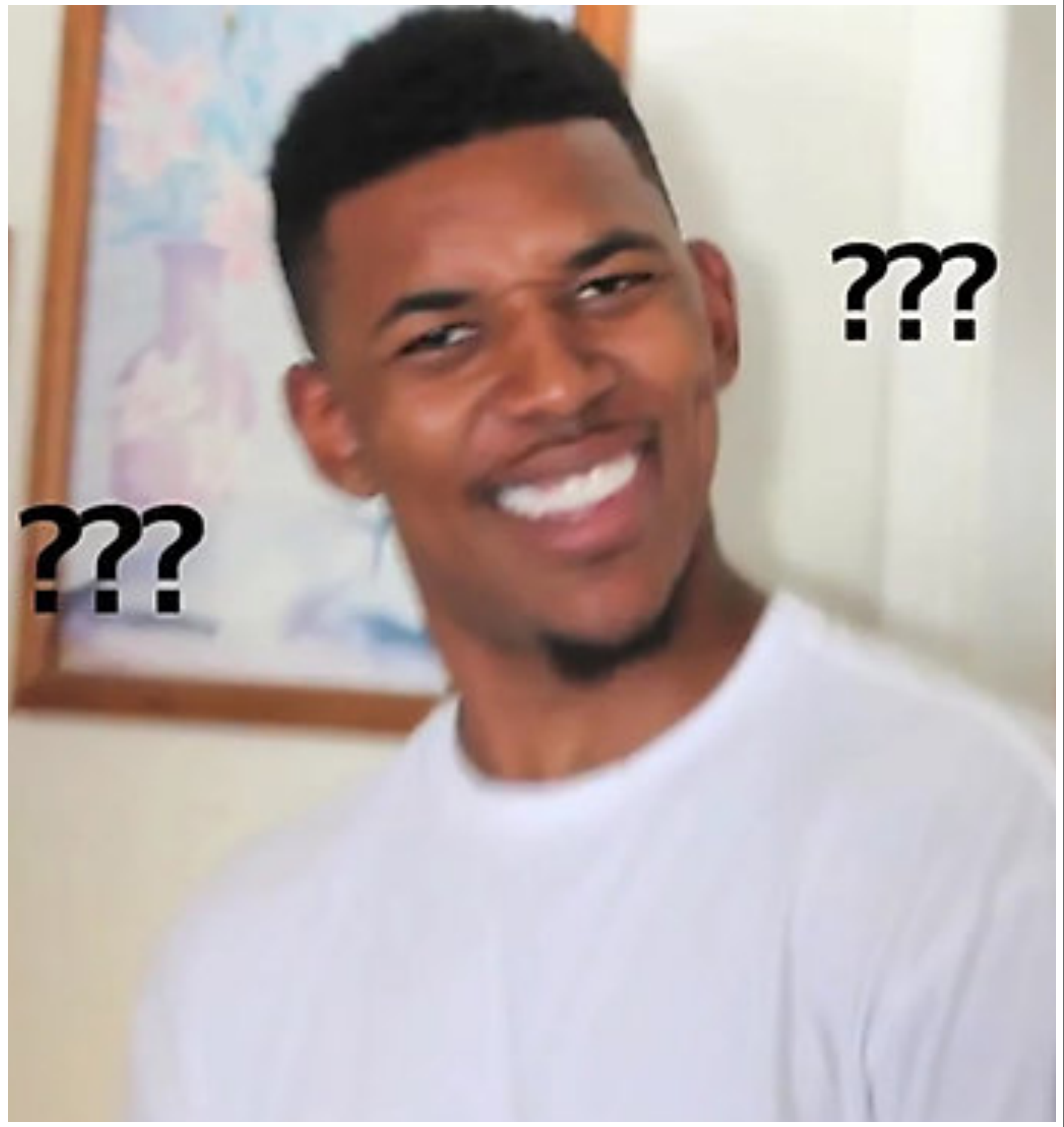
優點

Generalization

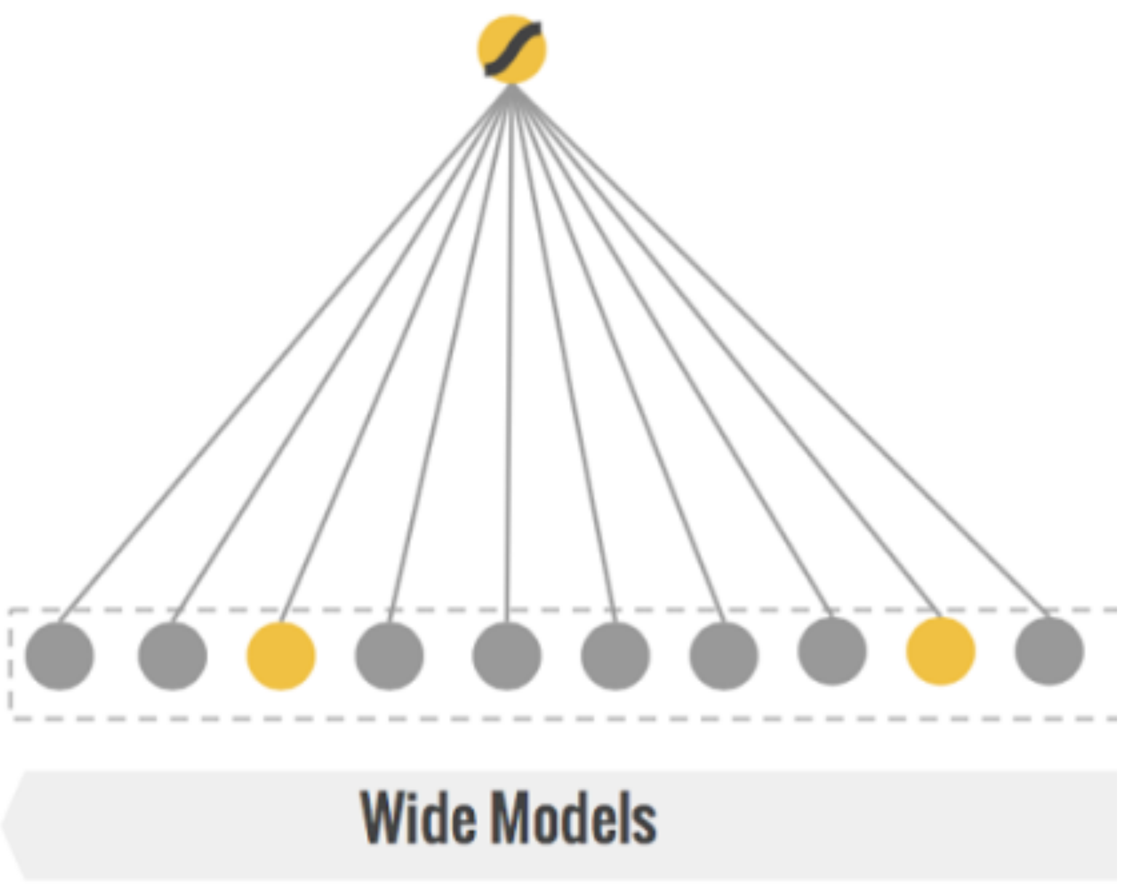


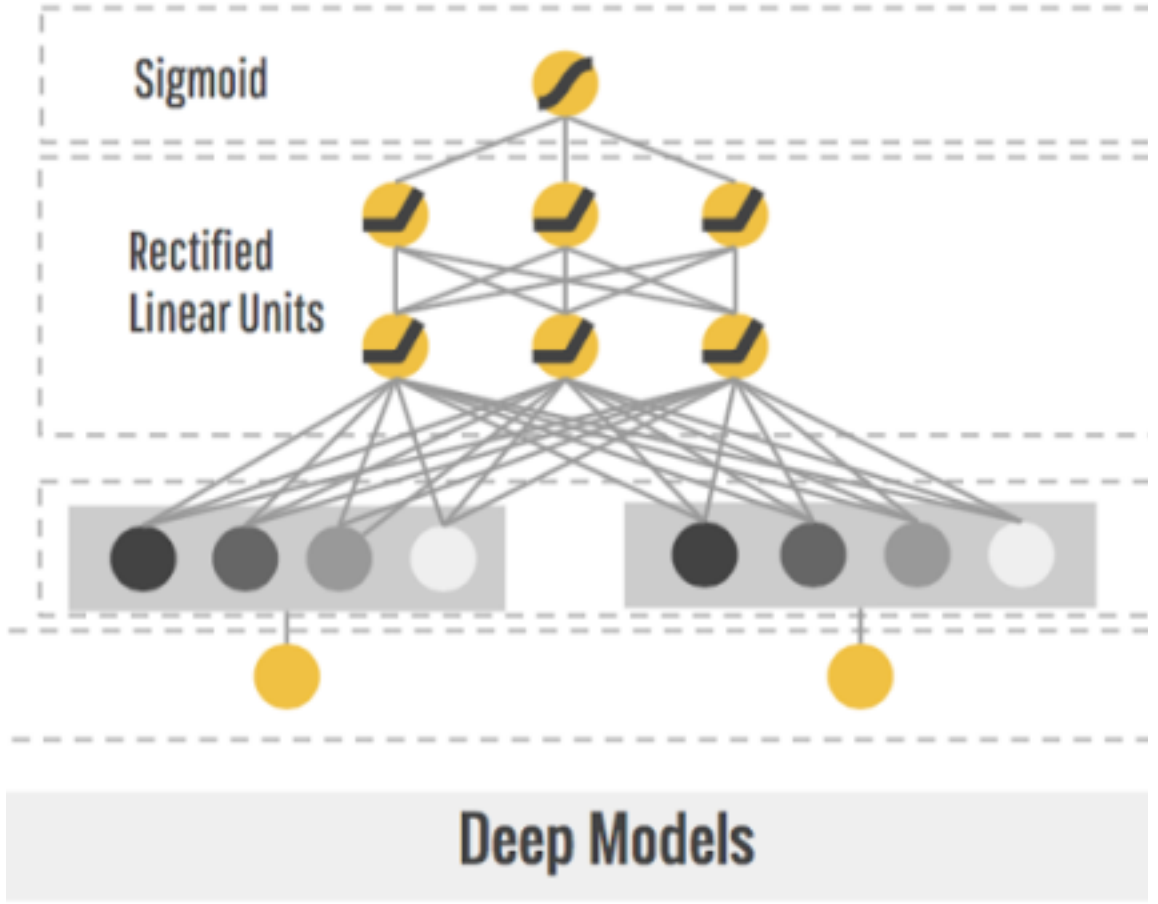
→

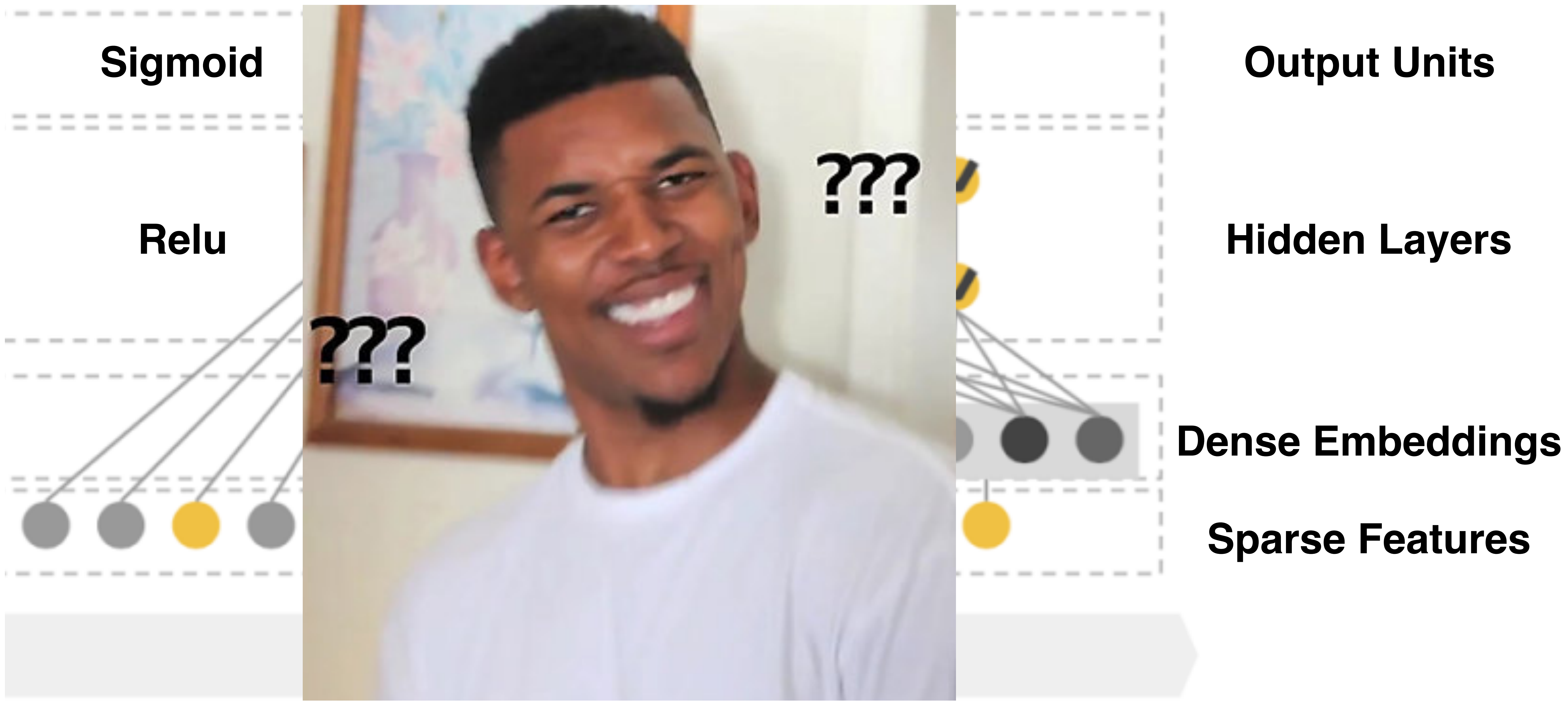
Wide & Deep



Wide & Deep


Wide Models


Sigmoid
Rectified
Linear Units
Deep Models


Sigmoid
Relu
???
???
Output Units
Hidden Layers
Dense Embeddings
Sparse Features

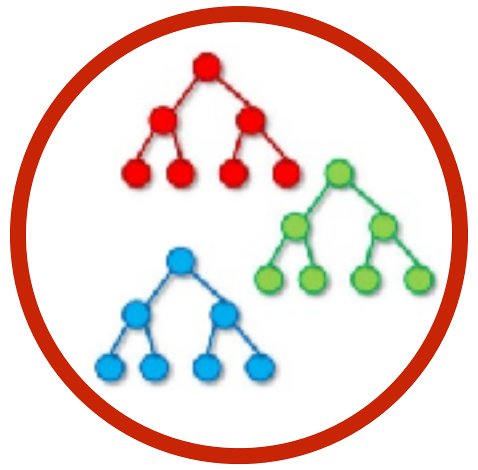
Ensemble vs Joint Training

Ensemble

單獨訓練，投票

↓

特徵數很多 -> 模型很大



	5分	4分	3分	2分	1分

Als

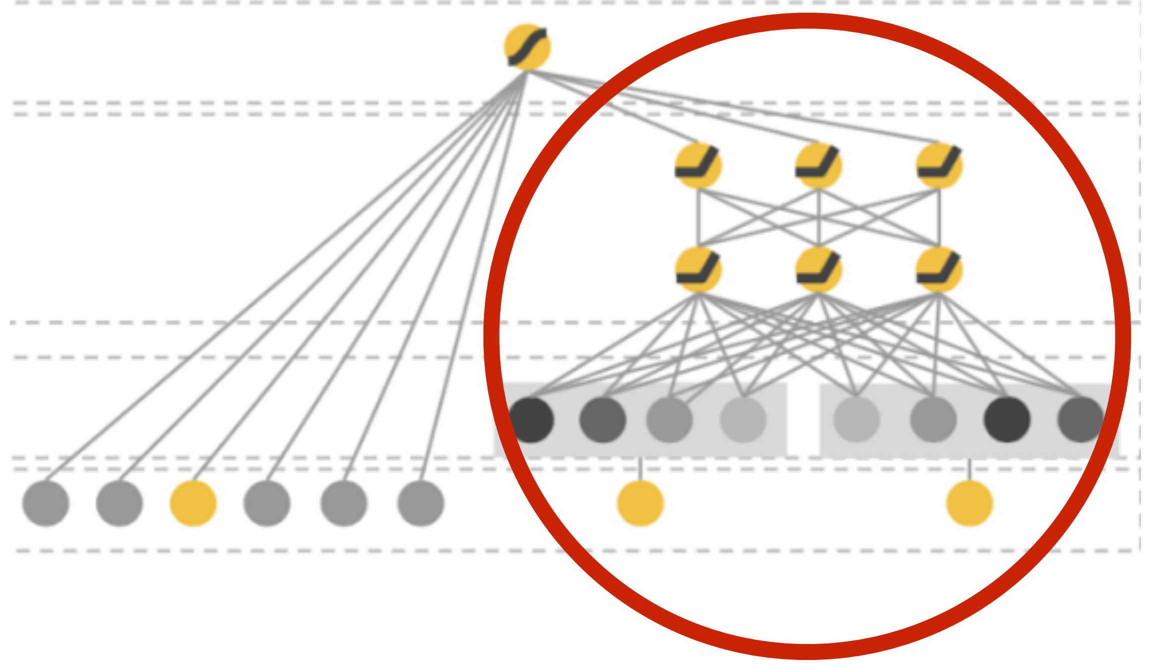
Xgboost

Joint Training

同時訓練

↓

Deep 特徵數較小



$$P(Y=1|x) = \text{sigmoid}(\text{wide} + \text{deep})$$

Future

Logistic Loss
Relu(256)
Relu(512)
Relu(1024)
Concatenated Embeddings
Crossed Product
Embeddings
Embeddings
Embeddings
Embeddings
Embeddings
3C
...
Age
Year_salary
utilization_rate
education
Place
consumption_level
Gender
Product Feature
Continuous Features
Categorical Features

小結

1. Memorization — Crossed Column → 找出表面關係
2. Generalization — Dense Embeddings → 找出潛在關係
3. Wide & Deep — Joint Model → **Wide + Deep**

Reference

1. Wide & Deep → https://arxiv.org/pdf/1606.07792.pdf

2. Gradient Descent Comparison

→ http://sebastianruder.com/optimizing-gradient-descent/index.html#gradientdescentvariants

→ http://vividfree.github.io/机器学习/2015/12/05/understanding-FTRL-algorithm

3. Online Learning → http://dataunion.org/5236.html

4. Deep Neural Networks for YouTube Recommendations

→ https://static.googleusercontent.com/media/research.google.com/zh-TW//pubs/archive/45530.pdf


Thank you

Appendix

BGD vs SGD vs FTRL

Logistic Regression

Online Learning

BGD

1. 更新梯度：整個數據同時訓練
2. 速度緩慢且無法使用Online Learning
3. 可透過regularization產生稀疏性

SGD

1. 更新梯度：隨機挑選一筆

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \eta_t \mathbf{g}_t$$

2. 速度快且震盪大
3. 隨機挑選，很難透過regularization產生稀疏性

FTRL

1. 解決準確度、稀疏問題

$$\mathbf{w}_{t+1} = \arg\min_{\mathbf{w}} \left(\sum_{s=1}^{t} \mathbf{g}_s \cdot \mathbf{w} + \frac{1}{2} \sum_{s=1}^{t} \sigma_s \|\mathbf{w} - \mathbf{w}_s\|_2^2 + \lambda_1 \|\mathbf{w}\|_1 \right)$$

2. 廣泛應用在CTR

延伸：BGD vs SGD vs FOBOS vs RDA vs FTRL

Algorithms

- FOBOS

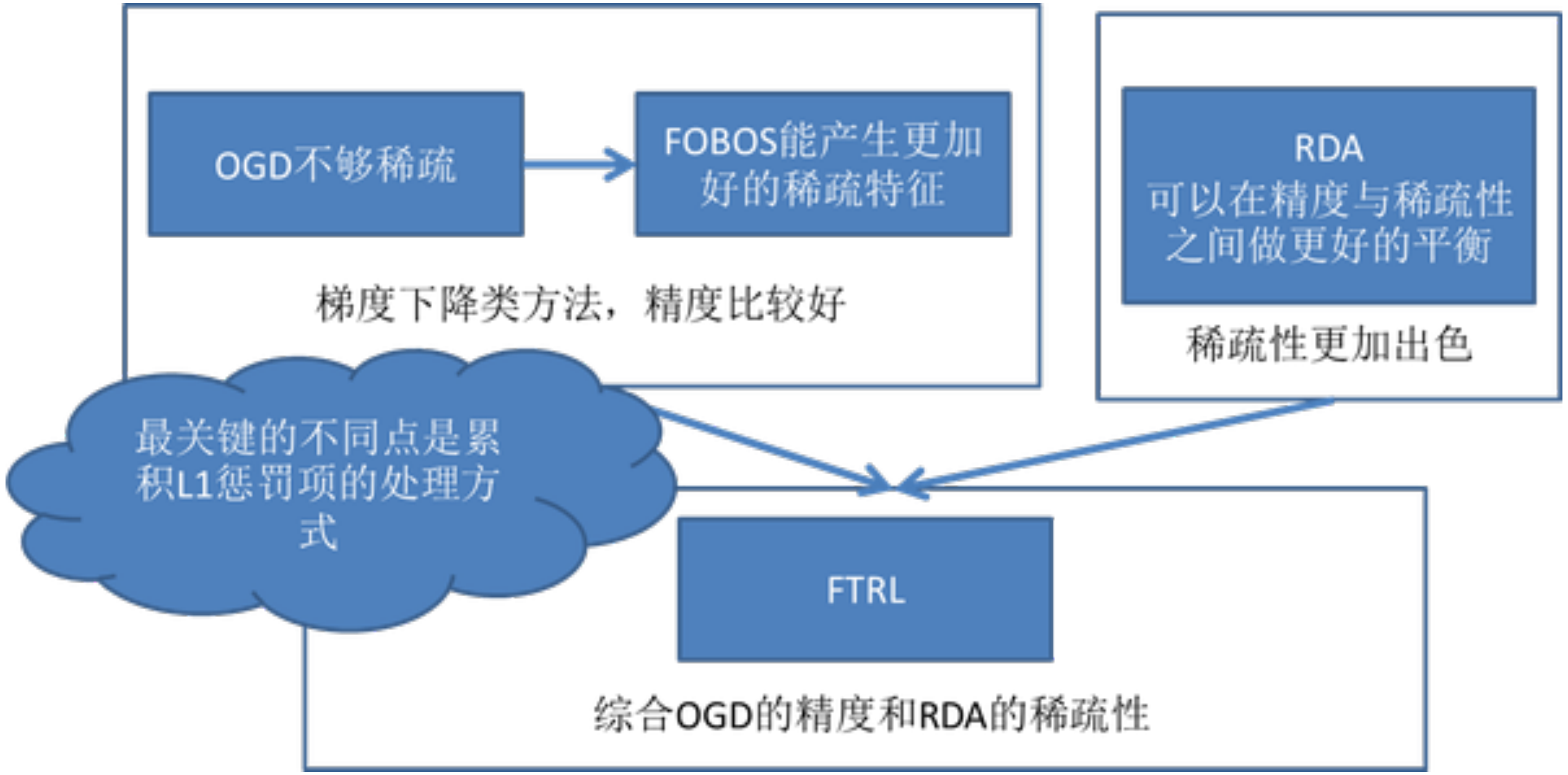
$$x_{t+1} = \arg\min_{x} g_t \cdot x + \lambda\|x\|_1 + \frac{1}{2}\|Q_{1:t}^{\frac{1}{2}}(x - x_t)\|_2^2.$$

- RDA

$$x_{t+1} = \arg\min_{x} g_{1:t} \cdot x + t\lambda\|x\|_1 + \frac{1}{2}\sum_{s=1}^{t}\|Q_s^{\frac{1}{2}}(x - 0)\|_2^2.$$

- FTRL-Proximal

$$x_{t+1} = \arg\min_{x} g_{1:t} \cdot x + t\lambda\|x\|_1 + \frac{1}{2}\sum_{s=1}^{t}\|Q_s^{\frac{1}{2}}(x - x_s)\|_2^2.$$



Follow the Regularized Leader

- FTRL-Proximal

$$w_i^{(t+1)} = \begin{cases} 0 & if \left|z_i^{(t)}\right| < \lambda_1 \\ -\left(\lambda_2 + \sum_{t}^{s=1}\sigma^{(s)}\right)^{-1}\left(z_i^{(t)} - \lambda_1 sgn\left(z_i^{(t)}\right)\right) & otherwise \end{cases}$$

- Per-Coordinate Learning Rates

$$\eta_i^{(t)} = \frac{\alpha}{\beta + \sqrt{\sum_{s=1}^{t}\left(g_i^{(s)}\right)^2}}$$

Algorithm 6. FTRL-Proximal with L1 & L2 Regularization

Input $\alpha,\ \beta,\ \lambda_1,\ \lambda_2$
Initialize $W \in \mathbb{R}^N,\ Z = 0 \in \mathbb{R}^N,\ Q = 0 \in \mathbb{R}^N$
for t =1,2,3… do
  $G = \nabla_W \ell\left(W, X^{(t)}, y^{(t)}\right)$ *# gradient of loss function*
  for i in 1,2,…,N do *# for each coordinate*
    $\sigma_i = \frac{1}{\alpha}\sqrt{q_i + g_i^2} - \sqrt{q_i}$ & $q_i = q_i + g_i^2$ *# equals* $\frac{1}{\eta^{(t)}} - \frac{1}{\eta^{(t-1)}}$
    $z_i = z_i + g_i - \sigma_i w_i$
    $w_i = \begin{cases} 0 & \text{if } \left|z_i^{(t)}\right| < \lambda_1 \\ -\left(\lambda_2 + \frac{\beta + \sqrt{q_i}}{\alpha}\right)^{-1}(z_i - \lambda_1 sgn(z_i)) & otherwise \end{cases}$
  end
end
return W

SGD vs ADAGRAD vs ADAM

Adaptive learning rate

SGD

1. 所有參數使用
相同學習率

$$\theta_{t+1,i} = \theta_{t,i} \boxed{- \eta \cdot g_{t,i}.}$$

自行設置學習率

ADAGRAD

1. 根據詞語出現頻率
調整學習率

$$\theta_{t+1,i} = \theta_{t,i} \boxed{- \frac{\eta}{\sqrt{G_{t,ii} + \epsilon}} \cdot g_{t,i}.}$$

分母大，學習率下降

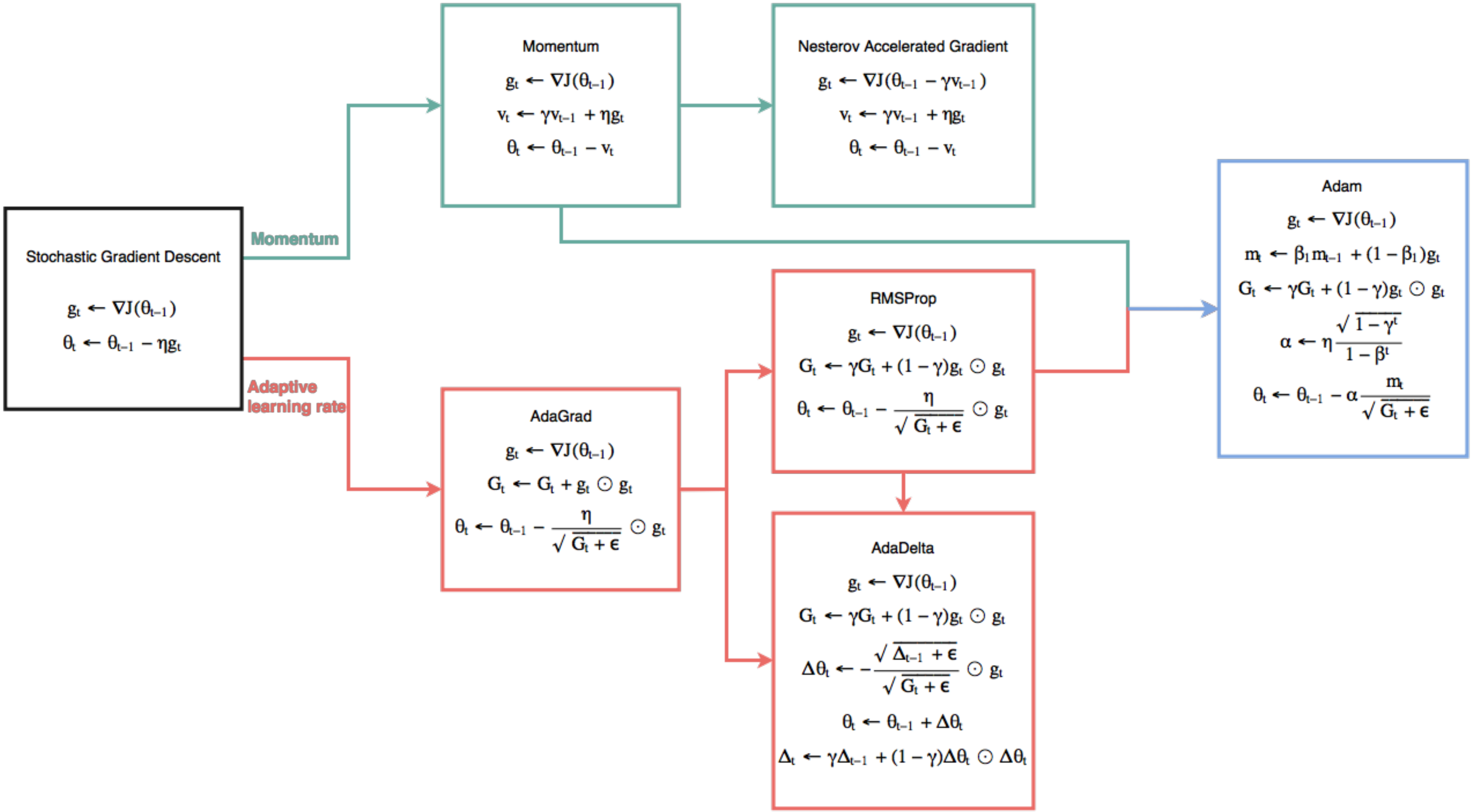
解法 ：ADADELTA、RMSPROP

ADAM

1. ADADELTA + Momentum
針對Momentum進行修正

2. 廣泛應用在參數優化

延伸 : SGD vs ADAGRAD vs ADADELTA vs RMSPROP vs ADAM


Stochastic Gradient Descent
$g_t \leftarrow \nabla J(\theta_{t-1})$
$\theta_t \leftarrow \theta_{t-1} - \eta g_t$
Momentum
Adaptive learning rate
Momentum
$g_t \leftarrow \nabla J(\theta_{t-1})$
$v_t \leftarrow \gamma v_{t-1} + \eta g_t$
$\theta_t \leftarrow \theta_{t-1} - v_t$
Nesterov Accelerated Gradient
$g_t \leftarrow \nabla J(\theta_{t-1} - \gamma v_{t-1})$
$v_t \leftarrow \gamma v_{t-1} + \eta g_t$
$\theta_t \leftarrow \theta_{t-1} - v_t$
AdaGrad
$g_t \leftarrow \nabla J(\theta_{t-1})$
$G_t \leftarrow G_t + g_t \odot g_t$
$\theta_t \leftarrow \theta_{t-1} - \frac{\eta}{\sqrt{G_t + \epsilon}} \odot g_t$
RMSProp
$g_t \leftarrow \nabla J(\theta_{t-1})$
$G_t \leftarrow \gamma G_t + (1-\gamma) g_t \odot g_t$
$\theta_t \leftarrow \theta_{t-1} - \frac{\eta}{\sqrt{G_t + \epsilon}} \odot g_t$
AdaDelta
$g_t \leftarrow \nabla J(\theta_{t-1})$
$G_t \leftarrow \gamma G_t + (1-\gamma) g_t \odot g_t$
$\Delta\theta_t \leftarrow -\frac{\sqrt{\Delta_{t-1} + \epsilon}}{\sqrt{G_t + \epsilon}} \odot g_t$
$\theta_t \leftarrow \theta_{t-1} + \Delta\theta_t$
$\Delta_t \leftarrow \gamma \Delta_{t-1} + (1-\gamma) \Delta\theta_t \odot \Delta\theta_t$
Adam
$g_t \leftarrow \nabla J(\theta_{t-1})$
$m_t \leftarrow \beta_1 m_{t-1} + (1-\beta_1) g_t$
$G_t \leftarrow \gamma G_t + (1-\gamma) g_t \odot g_t$
$\alpha \leftarrow \eta \frac{\sqrt{1-\gamma^t}}{1-\beta^t}$
$\theta_t \leftarrow \theta_{t-1} - \alpha \frac{m_t}{\sqrt{G_t + \epsilon}}$