



# 深度學習概論

Introduction to Deep Learning

蔡瑞煌 老師



國立政治大學金融科技研究中心  
智能理財與深度學習暑期訓練營



1



# AlphaGo

2

## 為什麼要機器學習?

Why we want Machine can be Learning?

3

人工智慧

機器學習



深度學習

4



一連串條件判斷式也是一種人工智慧

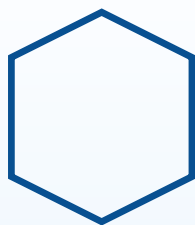


5

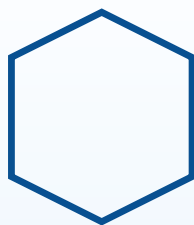


條件式人工智慧

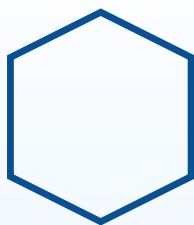
Rule-Based AI's Problem



條件建立困難  
Rule Build Difficult



笨重  
Cumbersome



昂貴  
Expensive

7



條件判斷人工智慧更像是工人智慧



8



為什麼我們不讓機器自己找到規則關係呢？



9



函數是一種關係  
Function is a relation



Function, Wikipedia

10

函數是個魔法盒子

Funtion is an Magic Box

輸入



函數



輸出



魔法盒子

11



讓你想到國中數學了嗎？



12



沒關係我們舉個例子



13

函數是個魔法盒子

Funtion is an Magic Box

過去二十天  
ETF資料



函數



明天  
ETF價格

魔法盒子

14



恭喜你！  
你完成了一次數學建模了！



15



而且如果模型預測的很好  
槓桿開最大催下去，你就發了



16

## 問題是

Question is

要怎麼讓魔法盒子預測？

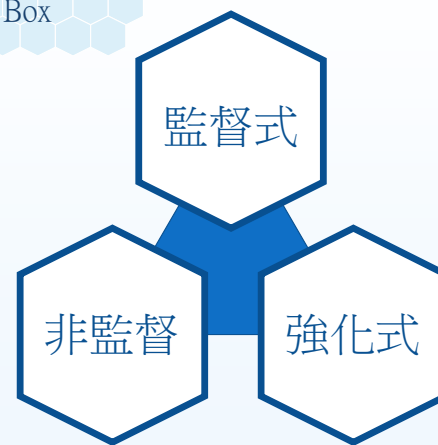
要怎麼改善魔法盒子的預測？

我們要如何評估魔法盒子準不準？

17

## 怎麼預測

Funtion is an Magic Box



18

## 監督式學習

Supervised Learning



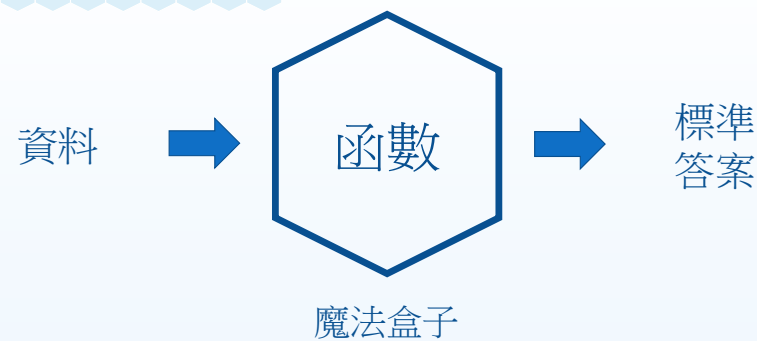
有正確答案的學習



19

## 監督式學習

Supervised Learning



20

## 監督式學習

Supervised Learning

二十天的  
ETF資料



函數



第二十一  
天ETF價格

魔法盒子

21

## 非監督式學習

Unsupervised Learning



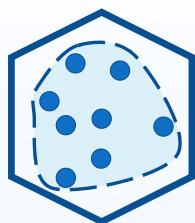
沒有正確答案的學習



22

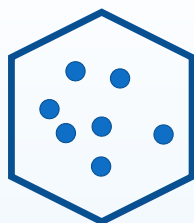
## 非監督式學習

Unsupervised Learning



生成

Generative



特徵抽取

Feature Extraction



分群

Clustering

23

## 強化式學習

Reinforcement Learning



靠棒子跟蘿蔔的學習



24

楊立昆

Yann LeCun

深度學習三巨頭之一  
卷積神經網路的發明者  
Facebook AI 研究院院長



25



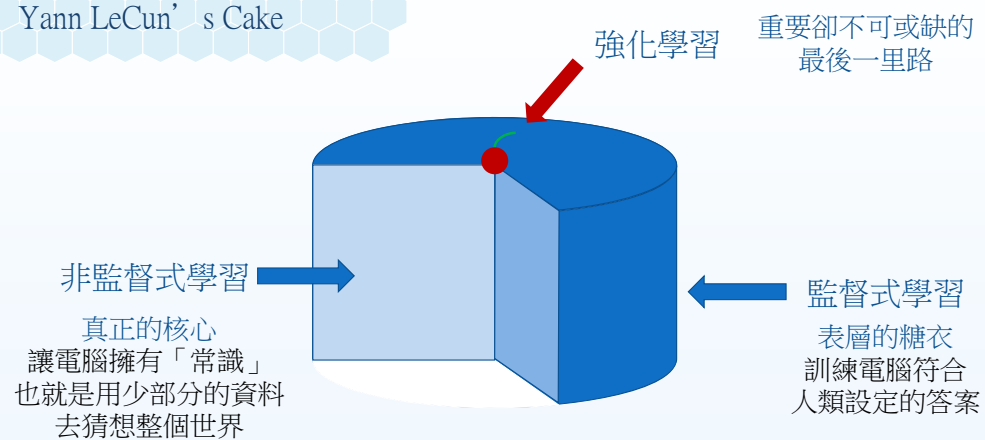
楊立昆對人工智慧提出一個比喻



26

楊立昆的蛋糕比喻

Yann LeCun's Cake



27

怎麼預測

How to Predict



丟資料進模型想辦法訓練他



28

## 怎麼預測

How to Predict



要怎的對資料下手呢？



29

資料

30

訓練資料

測試資料

31

訓練資料

檢驗資料

測試資料

32



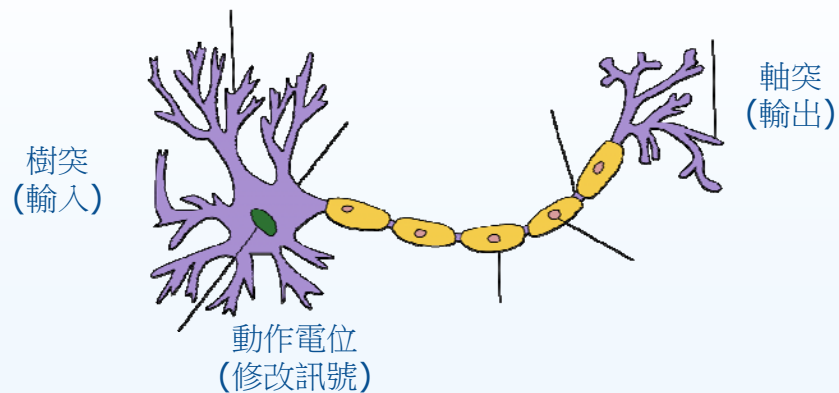
# 什麼是深度學習?

What is Deep Learning?

33

## 向大自然學習

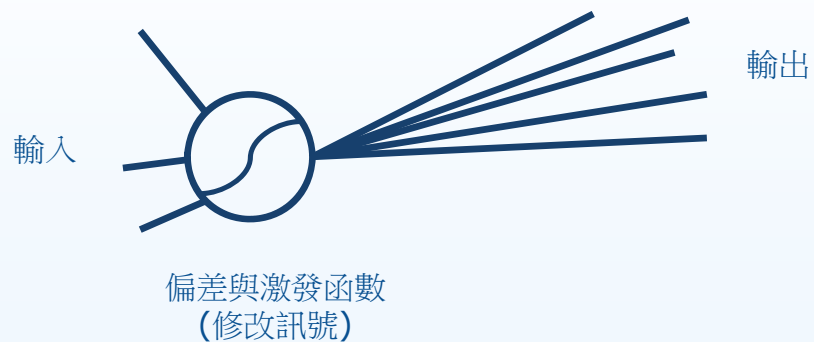
Learn from Nature



34

## 抽象化神經元

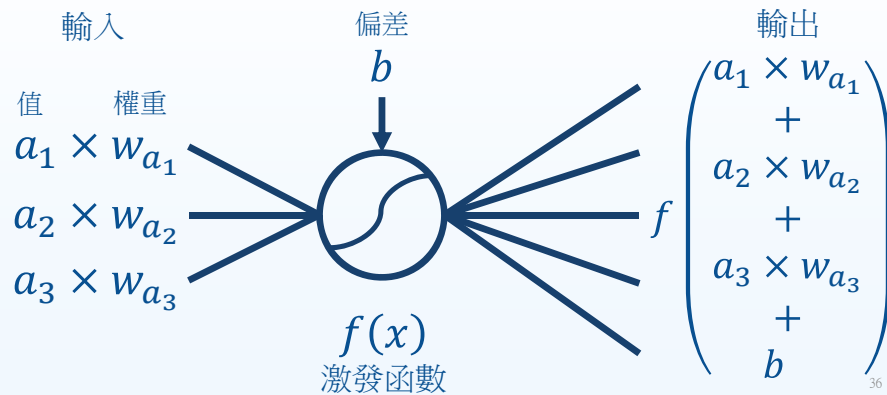
Abstract Neuron



35

## 抽象化神經元

Abstract Neuron



36

## 全連接神經網路

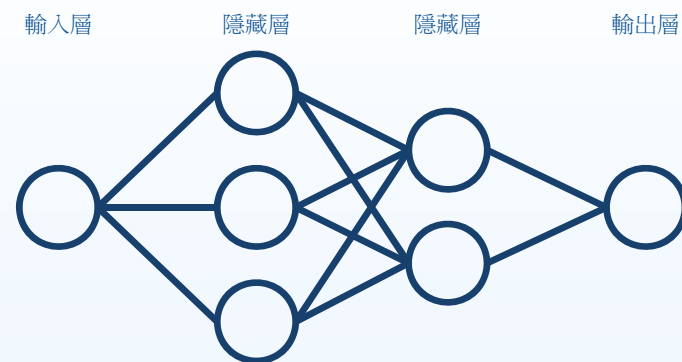
Fully-Connected Neural Network



37

## 全連接神經網路

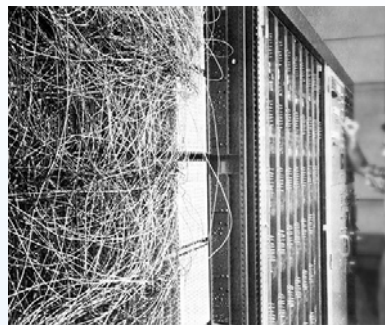
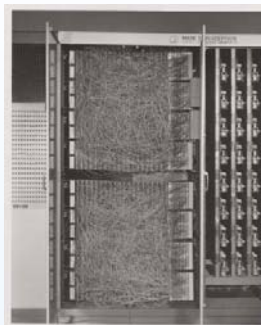
Fully-Connected Neural Network



38

1957就有人做了

Cornell University, 1957



感知器

Perceptron

Frank Rosenblatt

39

等等1959就有人做了  
那現在為什麼那麼紅？



40

## 為什麼紅？

Why Hot?



電腦變快

Faster Computer



大數據

Big Data



演算法改善

Improved Algorithm

41



我們其實可以近似任意的函數



42

## 通用近似定理

Universal approximation theorem



只要一層足夠多神經元的隱藏層就能逼近  
任何在歐氏空間上緊緻子集的連續函數



43



神經網路是用很多很多小函數來  
逼近一個我們希望得到的大函數



44

## 激發函數

Activate Function



那我們要用哪種小函數呢？

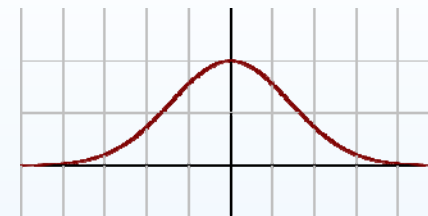


45

## 高斯分佈函數

Activate Function

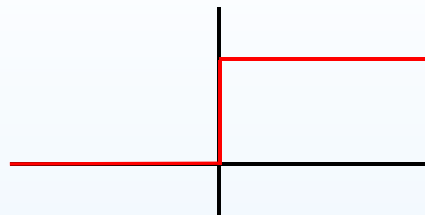
$$f(x) = e^{-x^2}$$



46

## Binary step

$$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$$

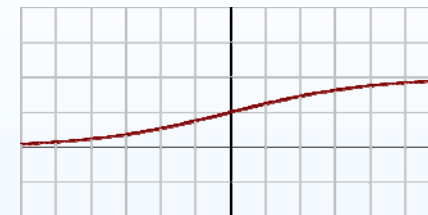


47

## S型函數

Sigmoid

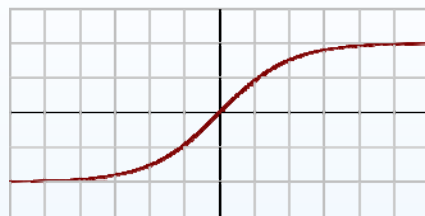
$$f(x) = \frac{1}{1 + e^{-x}}$$



48

## Hyperbolic tangent

$$f(x) = \tanh(x) = \frac{1 - e^{-2x}}{1 + e^{-2x}}$$



49

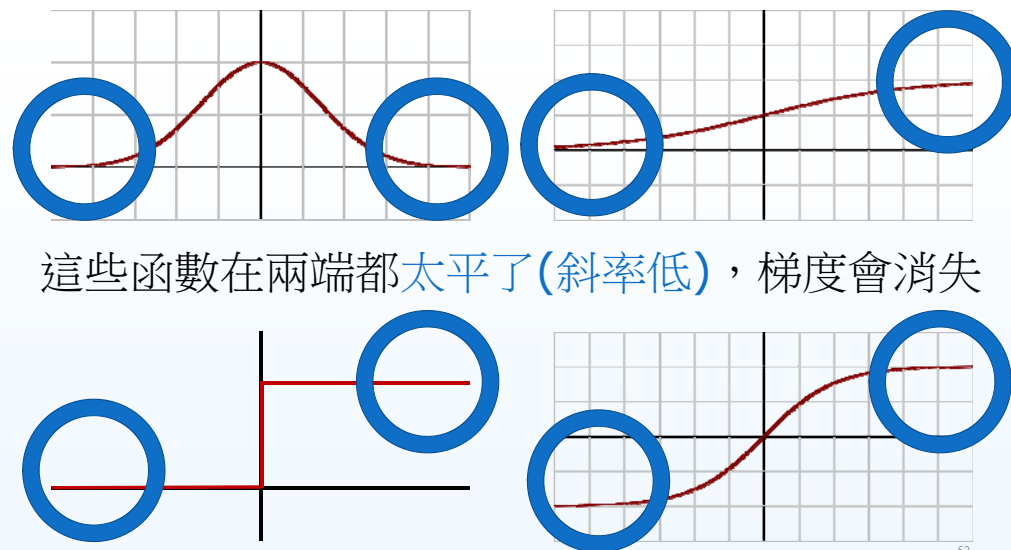
但這些都有個問題

But, here is a problem

50

我們在訓練模型的時候是利用微分取梯度的  
(後面會詳述)

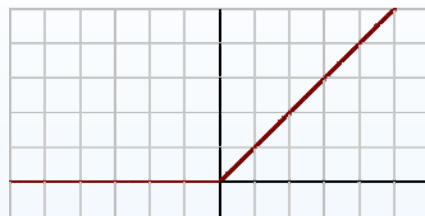
51



52

## ReLu

$$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$$



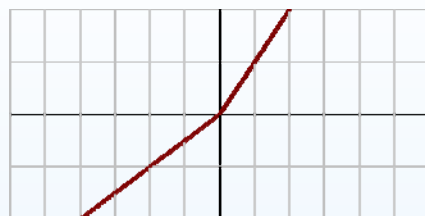
53

但是這樣會 $x < 0$ 導致神經元無法重啓  
解決方法是加上一個 $\alpha > 0$ 的常數

54

## Leaky ReLu

$$f(x) = \begin{cases} \alpha x & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$$

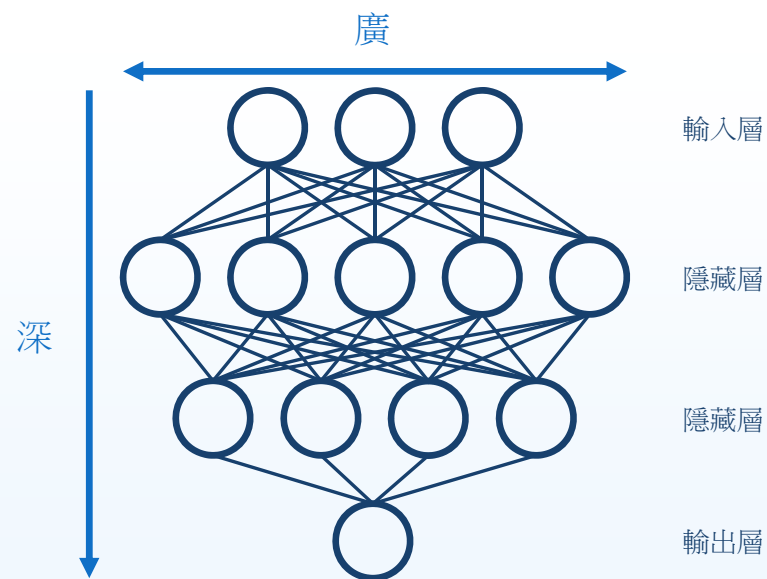


55

深還是廣？

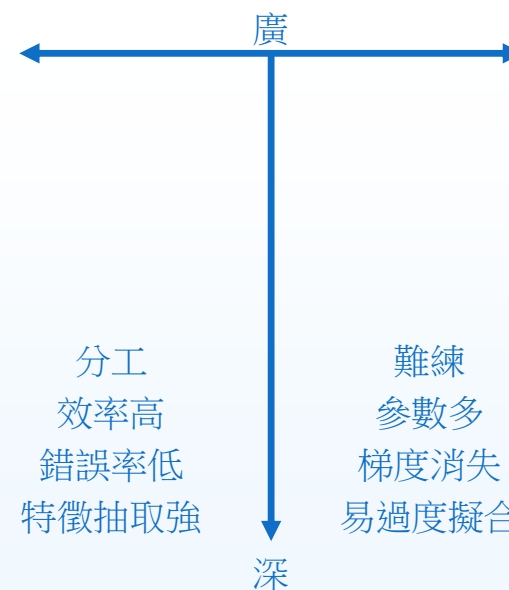
Be Deep or Be Wide

56



57

優



58

怎麼評估准不準?

How to Score it

59



我們其實只有下面兩樣東西



60

正確答案

模型答案

61

損失函數

Loss Function

能夠表達  
正確答案和模型答案  
差異的函數

62

平均絕對差

Mean Absolute Error

$$\sum \frac{|y_{data} - y_{predict}|}{n}$$

$y_{data}$  = 正確答案     $y_{predict}$  = 模型答案

63

平均平方差

Mean Square Error

$$\sum \frac{(y_{data} - y_{predict})^2}{n}$$

$y_{data}$  = 正確答案     $y_{predict}$  = 模型答案

64



## 更多的損失函數

More Loss Function

MAPE	Categorical Crossentropy
MALE	Binary Crossentropy
Hinge	Total Variation
LogCosh	KL Divergence

65

## 怎麼訓練我的神經網路?

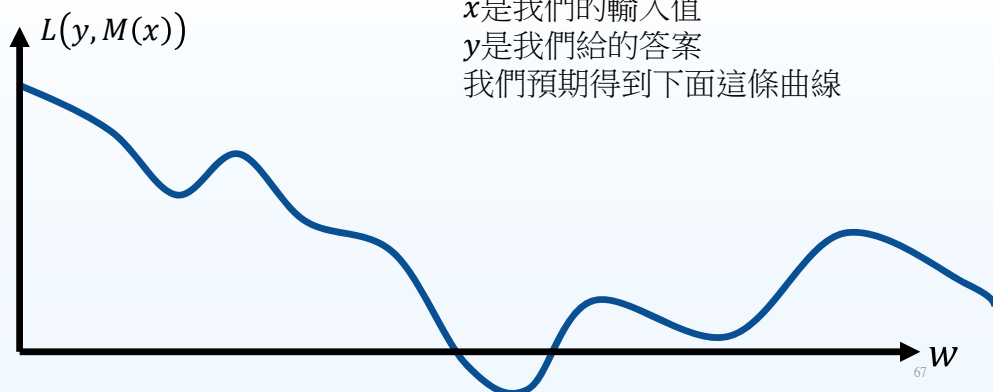
How to Train our Neural Networks

66

## 梯度下降法

Gradient Descent

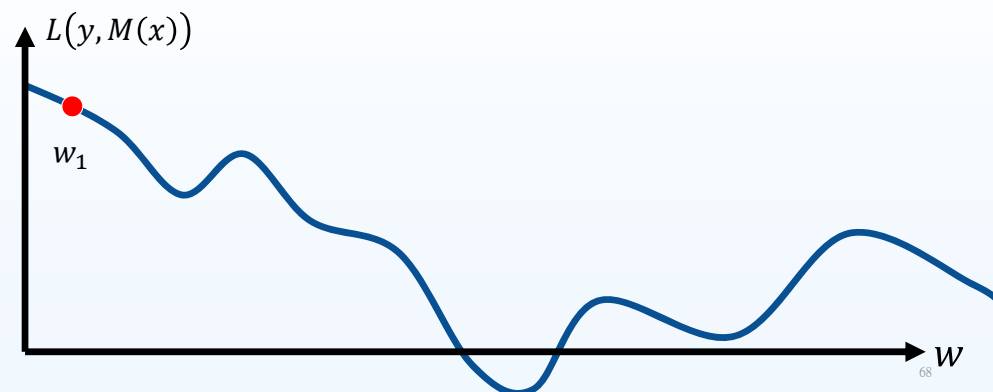
假設我們有個  
模型： $M(x)$   
模型權重： $w$   
損失函數： $L(y, M(x))$   
 $x$ 是我們的輸入值  
 $y$ 是我們給的答案  
我們預期得到下面這條曲線



## 梯度下降法

Gradient Descent

在 $w_1$ 這點對 $L(y, M(x))$ 微分



## 梯度下降法

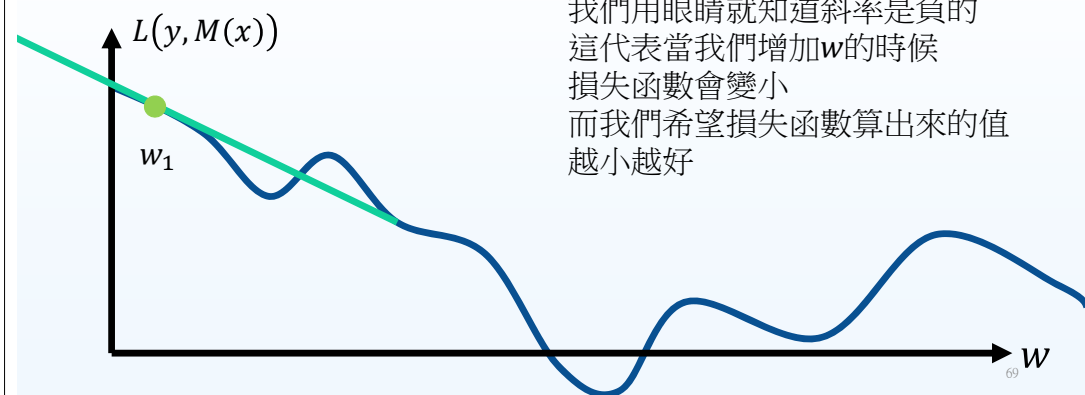
Gradient Descent

我們可以得到一個斜率

$$\frac{dL}{dw_1}$$

根據國中數學

我們用眼睛就知道斜率是負的  
這代表當我們增加 $w$ 的時候  
損失函數會變小  
而我們希望損失函數算出來的值  
越小越好



## 梯度下降法

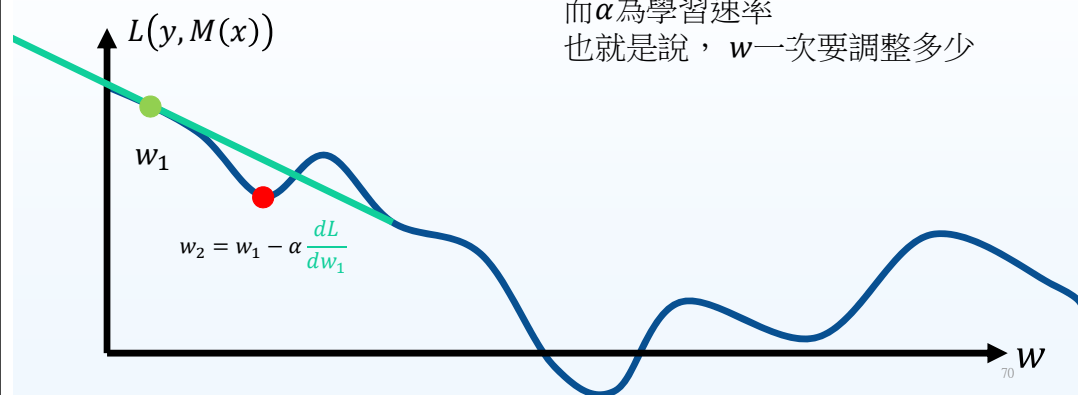
Gradient Descent

所以我們新的 $w$ ，  
也就是 $w_2$ 就會等於

$$w_2 = w_1 - \alpha \frac{dL}{dw_1}$$

而 $\alpha$ 為學習速率

也就是說， $w$ 一次要調整多少



調整每個權重都是這樣

只是把其他權重當作常數微分 (偏微分) 後  
得到梯度 (可以視為多維度下的斜率)

實務上會隨機抽一小單位資料改善模型的權重來加速  
也就是

隨機梯度下降

Stochastic Gradient Descent

SGD



## Batch



訓練完整個訓練資料中的一小單位資料



## Epoch



訓練全部的訓練資料



73



就這樣一步一步總有一天會到最佳解的



嗎？

74

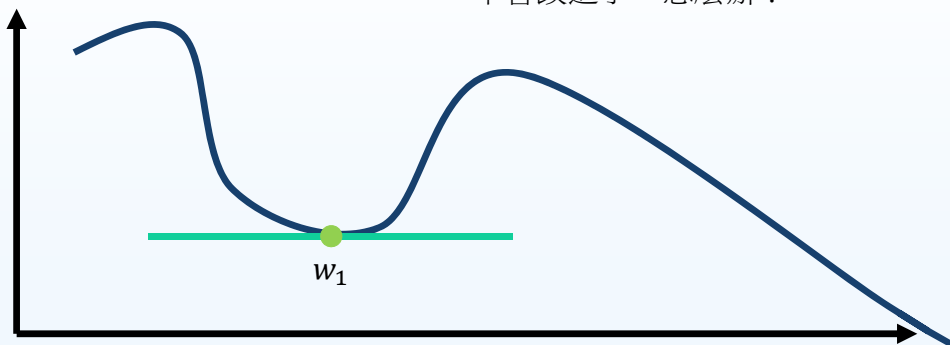
## 動量

Momentum

如果我們今天很不幸的落在低谷  
微分後斜率又剛好等於零

$$w_2 = w_1 - \alpha \frac{dL}{dw_1} = w_1 - \alpha \times 0 = w_1$$

不會改進了，怎麼辦？



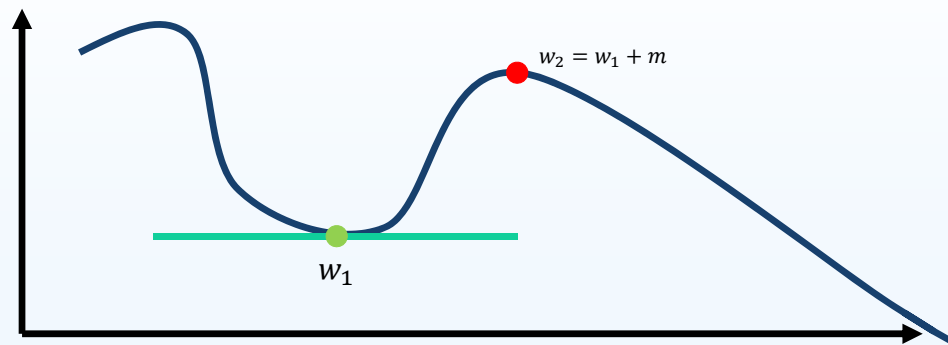
75

## 動量

Momentum

那就加入一個動量常數 $m$   
讓 $w_2$ 變成

$$w_2 = w_1 + m - \alpha \frac{dL}{dw_1} = w_1 + m$$



76

## 更多的優化法

More Optimizers

Adam

Adagrad

Adadelta

Nadam

Adamax

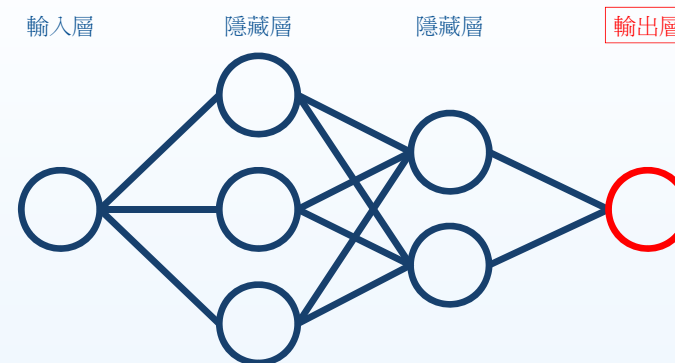
RMSprop

77

## 反向傳播算法

Back Propagation

從輸出層向輸入層微分

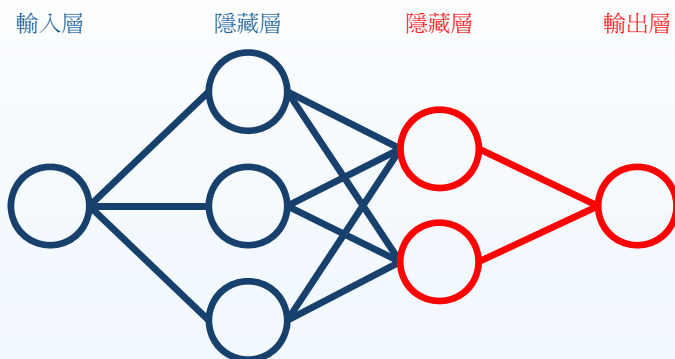


78

## 反向傳播算法

Back Propagation

從輸出層向輸入層微分

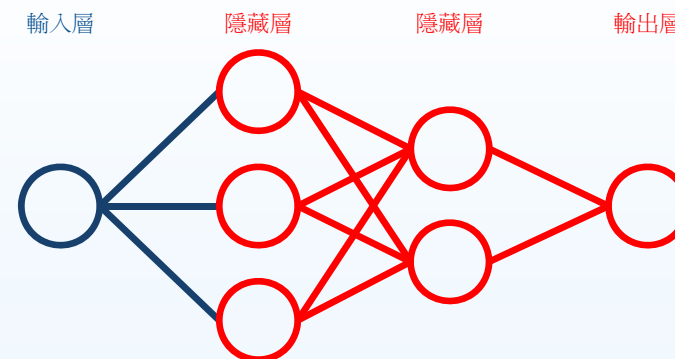


79

## 反向傳播算法

Back Propagation

從輸出層向輸入層微分

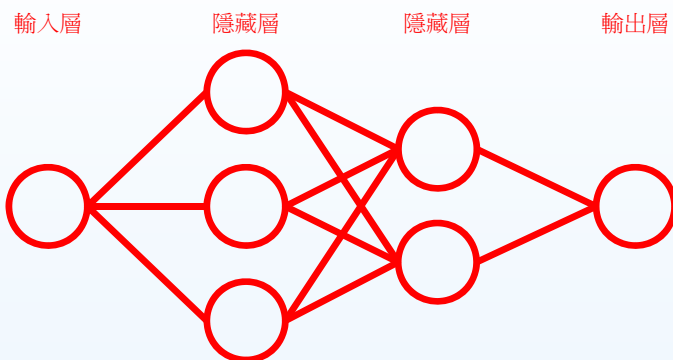


80

## 反向傳播算法

Back Propagation

從輸出層向輸入層微分



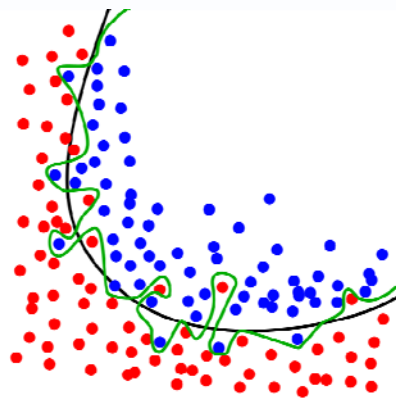
81

但其實我們很害怕一件事

82

## 過度擬合

Over Fitting



83

怎麼辦？

84



既然練過頭，那就不要練啊

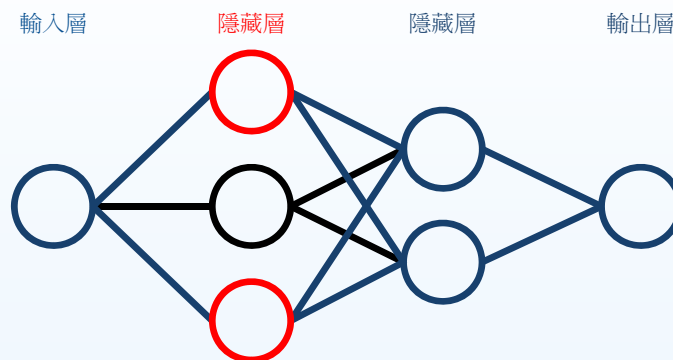


85

不參與訓練

Drop Out

把部份神經元關掉不做訓練



86

Keras 介紹與實做

Introduce Keras

87

如何實現深度學習模型

How to Achieve Deep Learning Model

手算

88

如何實現深度學習模型

How to Achieve Deep Learning Model

# 手算

對理論會更有感覺  
但是很慢  
還會算錯很不方便

89

如何實現深度學習模型

How to Achieve Deep Learning Model

# Numpy

要刻每一層、優化器、損失函數  
調用GPU加速不方便

90

## 為什麼要用GPU？

Why GPU?

91



GPU

核心多  
(上千個)

單核心速度慢

擅長平行運算



CPU

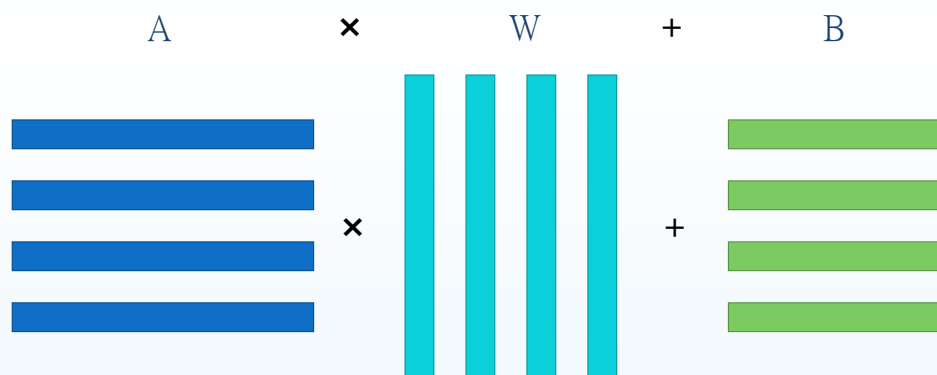
核心少  
(小餘三十個)

單核心速度快

擅長序列運算

建議購買NVIDIA GPU，CUDA支援的框架比較多

92



平行運算在矩陣運算速度上具有優勢

93

## 深度學習運算框架

Deep Learning Framework

為了方便調用GPU  
並且不要重複造輪子  
我們通常會使用一些框架  
現在還在框架大戰



94

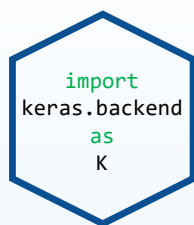
## K Keras



支援多種後端  
Multiple Backend



模組化  
Modular



適度的擴展性  
Enough Elasticity

95



我們今天有兩個資料集

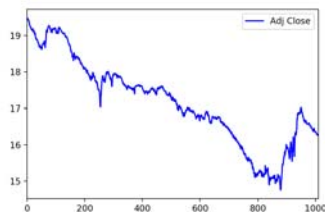


96





MNIST  
手寫數字辨識  
深度學習的HelloWorld



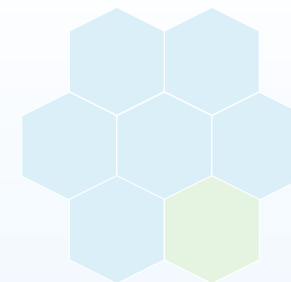
ETF  
全球智能提供  
介紹實務上的應用

97

## MNIST手寫數字辨識

訓練資料： 六萬  
測試資料： 一萬

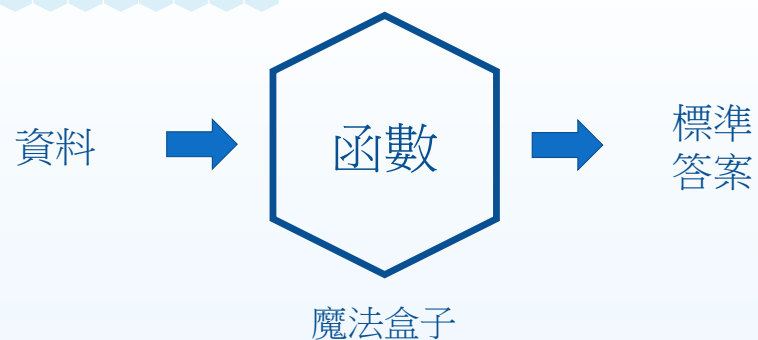
資料分類比夠大  
機器學習界的HelloWorld



LeCun, Yann. "The MNIST database of handwritten digits." <http://yann.lecun.com/exdb/mnist/> (1998).

98

來建模型吧



魔法盒子

99

試試全連接監督學習



28x28



0  
1

100

## One-hot Embedding

0 1 2 3 4 5 6 7 8 9

這些數字彼此是沒有相關性的  
1跟7比較像，但是1卻不一定在7的旁邊

101

## One-hot Embedding

最好的方法是讓每一種分類  
自己獨立成一個向量

0 1 2 3 4 5 6 7 8 9

[0 0 0 0 1 0 0 0 0 0]

102

## 首先先讀入套件跟資料

```
%matplotlib inline #讓jupyter notebook可以出現你畫的圖片
import numpy as np
import matplotlib.pyplot as plt
from keras.datasets import mnist #資料集
from keras.utils import np_utils #做one-hot embedding
from keras.models import Sequential #讀入Sequential模式以建立模型
from keras.layers import Dense, Activation #我們需要的keras層
from keras.optimizers import SGD #這次用的優化器

(x0_train, y0_train), (x0_test, y0_test) = mnist.load_data() #讀入MNIST資料集
```

103

## 資料整理

```
#將輸入資料打平
x_train = x0_train.reshape(60000, 28*28)
x_test = x0_test.reshape(10000, 28*28)

#將輸出資料做One-hot Embedding
y_train = np_utils.to_categorical(y0_train,10)
y_test = np_utils.to_categorical(y0_test,10)
```

104

## 構建模型

### 序列式模型

```
model = Sequential()
```

105

## 構建模型

### 序列式模型

500個神經元的全連接層

激發函數用S型函數

```
model = Sequential()  
model.add(Dense(units=500, input_dim=784))  
model.add(Activation('sigmoid'))
```

106

## 構建模型

### 序列式模型

500個神經元的全連接層

激發函數用S型函數

500個神經元的全連接層

激發函數用S型函數

```
model = Sequential()  
model.add(Dense(units=500, input_dim=784))  
model.add(Activation('sigmoid'))  
model.add(Dense(units=500))  
model.add(Activation('sigmoid'))
```

107

## 構建模型

### 序列式模型

500個神經元的全連接層

500個神經元的全連接層

10個神經元的全連接層

激發函數用Softmax

```
model = Sequential()  
model.add(Dense(units=500, input_dim=784))  
model.add(Activation('sigmoid'))  
model.add(Dense(units=500))  
model.add(Activation('sigmoid'))  
model.add(Dense(units=10))  
model.add(Activation('softmax'))  
model.summary() #檢查模型形狀
```

108

## 編譯模型

```
model.compile(loss='mse', optimizer=SGD(lr=0.1), metrics=['accuracy'])
```

決定損失函數和優化器  
並且新增準確度作為測量記錄

109

## 訓練模型

```
model.fit(x_train, y_train, batch_size=100, epochs=20)
```

決定批次大小以及訓練次數

110

## 試用結果

```
result = model.predict_classes(x_test)
```

111

## 儲存模型

```
model_json = model.to_json()  
open('handwriting_model_architecture.json', 'w').write(model_json)  
model.save_weights('handwriting_model_weights.h5')
```

儲存模型形狀及權重

112



對Keras有基本概念了吧  
我們開始試試看預測ETF吧



113

ETF每天都有下面這些資料

開盤價 收盤價 最高價 最低價 交易量 調整收盤價

114

來建模型吧

資料



函數



標準  
答案

魔法盒子

115

來建模型吧

前二十天的  
開盤價



函數



第二十一天的  
開盤價

魔法盒子

116

來建模型吧

前二十天的  
開盤價



全連接  
神經網路



第二十一天的  
開盤價

117

可能更好的模型

前二十天的  
開盤價  
收盤價  
最高價  
最低價  
交易量  
調整收盤價



全連接  
神經網路



第二十一天的  
開盤價  
漲幅

118

可能更好的模型

前二十天的  
開盤價  
收盤價  
最高價  
最低價  
交易量  
調整收盤價



全連接  
神經網路



第二十一天的  
開盤價  
漲幅

很多份ETF

119