



深度學習 類神經網路介紹

王家祥

2020-11-06

CONTENTS

1. 類神經網路介紹
2. 常用框架
3. Keras簡單類神經網路實作Demo
4. 補充(10月搞炸的BERT Model)



A large red hexagon with a white slash and the number 01 inside.

/01

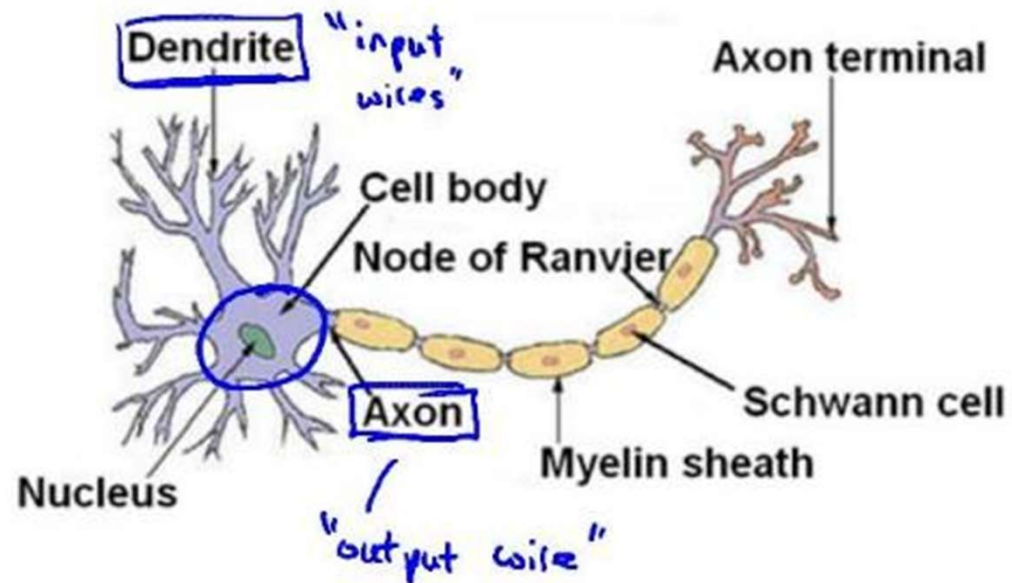
類神經網路介紹

1.1 神經元架構

Dendrite/樹突(input)

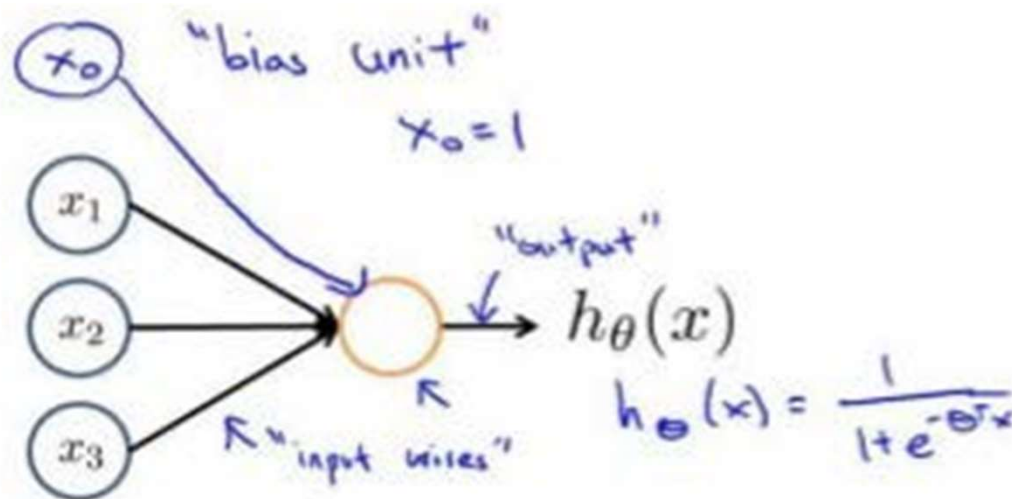
Axon/軸突(output)

Nucleus/神經元(activate, compute)



1.2定義

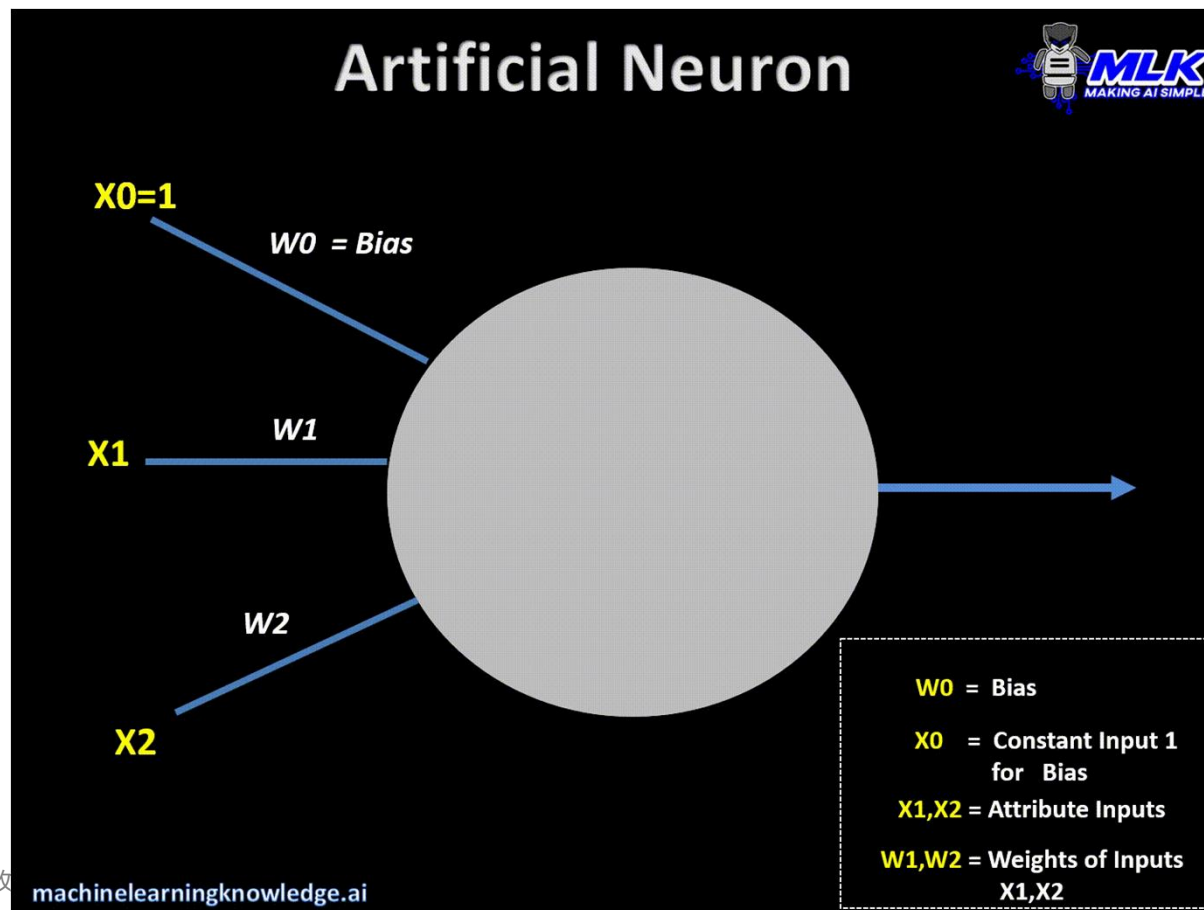
- 類神經網路是一種模仿生物神經系統的數學模型。
- 神经网络模型建立在很多神经元之上，每一个神经元又是一个个学习模型。这些神经元（也叫激活单元，activation unit）采纳一些特征作为输入，并且根据本身的模型提供一个输出。



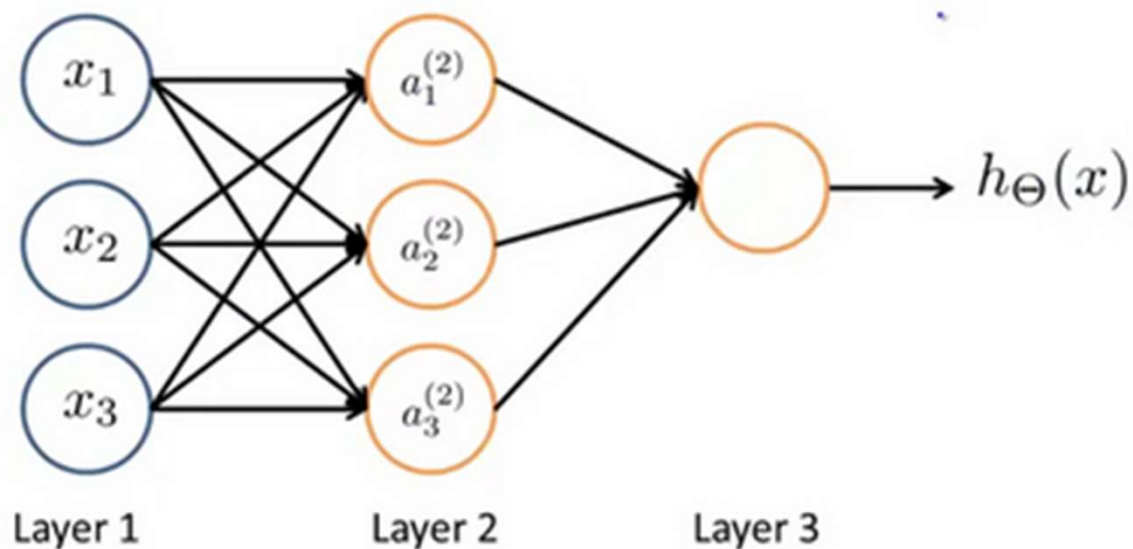
Sigmoid (logistic) activation function.

1.2定義

- 類神經網路是一種模仿生物神經系統的數學模型。



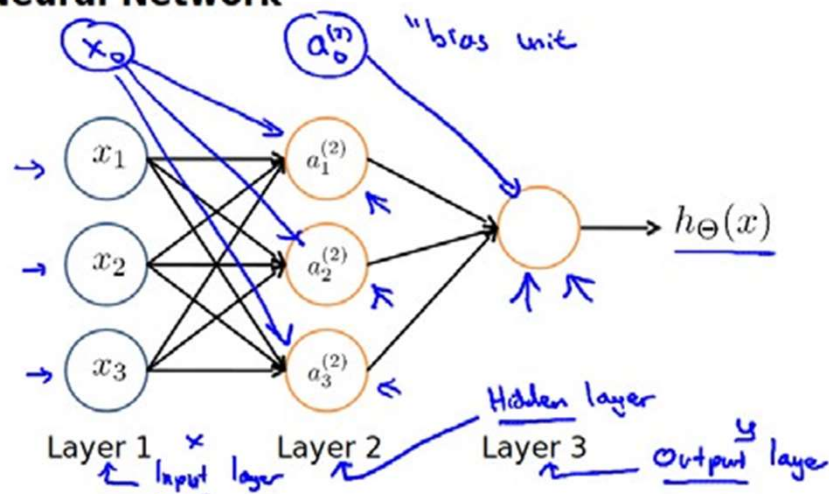
1.3 簡單的神經網路架構



其中, x_1, x_2, x_3 是输入单元 (input units) ,
我们将原始数据输入给它们。 a_1, a_2, a_3 是中间单元,
它们负责将数据进行处理, 然后呈递到下一层。
最后是输出单元, 它负责计算 $h(x)$ 。

1.3 簡單的神經網路架構

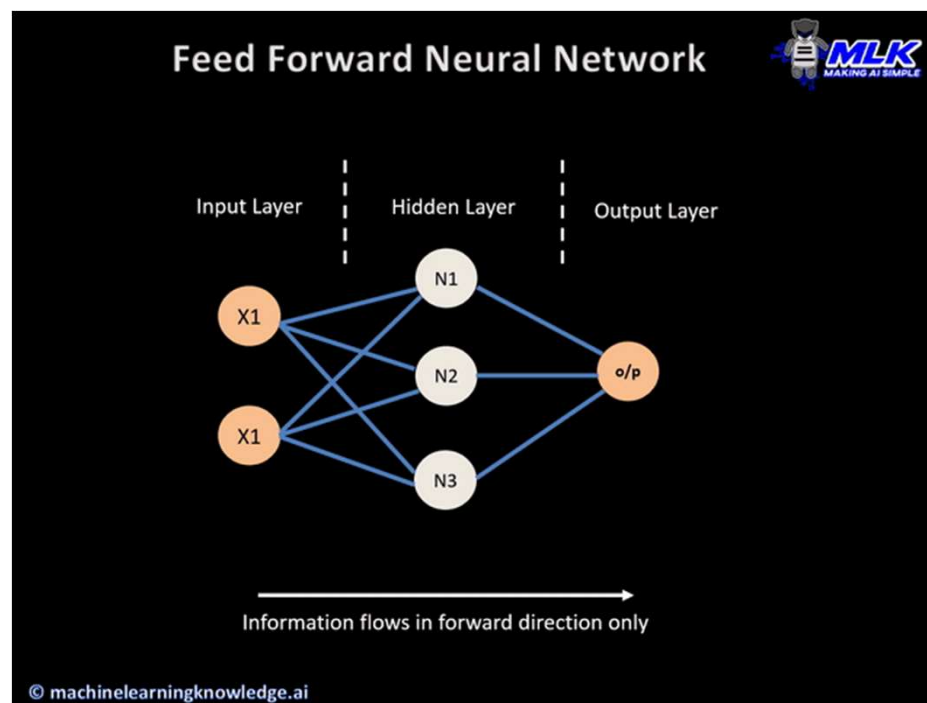
Neural Network



$$\begin{aligned}
 a_1^{(2)} &= g(\Theta_{10}^{(1)} x_0 + \Theta_{11}^{(1)} x_1 + \Theta_{12}^{(1)} x_2 + \Theta_{13}^{(1)} x_3) & a_2^{(2)} &= g(\Theta_{20}^{(1)} x_0 + \Theta_{21}^{(1)} x_1 + \Theta_{22}^{(1)} x_2 + \Theta_{23}^{(1)} x_3) \\
 a_3^{(2)} &= g(\Theta_{30}^{(1)} x_0 + \Theta_{31}^{(1)} x_1 + \Theta_{32}^{(1)} x_2 + \Theta_{33}^{(1)} x_3) & h_{\Theta}(x) &= g(\Theta_{10}^{(2)} a_0^{(2)} + \Theta_{11}^{(2)} a_1^{(2)} + \Theta_{12}^{(2)} a_2^{(2)} + \Theta_{13}^{(2)} a_3^{(2)})
 \end{aligned}$$

1.4 前向傳播

- 上面进行的讨论中只是将特征矩阵中的一行（一个训练实例）喂给了神经网络，我们需要将整个训练集都喂给我们的神经网络算法来学习模型。
- 我们可以知道：每一个a都是由上一层所有的x和每一个x所对应的w决定的。
- 我们把这样从左到右的算法称为前向传播算法(FORWARD PROPAGATION)



1.4 前向傳播

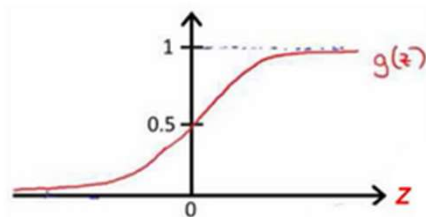
- 寫成向量格式如下:

$$g \left(\begin{bmatrix} \theta_{10}^{(1)} & \theta_{11}^{(1)} & \theta_{12}^{(1)} & \theta_{13}^{(1)} \\ \theta_{20}^{(1)} & \theta_{21}^{(1)} & \theta_{22}^{(1)} & \theta_{23}^{(1)} \\ \theta_{30}^{(1)} & \theta_{31}^{(1)} & \theta_{32}^{(1)} & \theta_{33}^{(1)} \end{bmatrix} \times \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix} \right) = g \left(\begin{bmatrix} \theta_{10}^{(1)} x_0 + \theta_{11}^{(1)} x_1 + \theta_{12}^{(1)} x_2 + \theta_{13}^{(1)} x_3 \\ \theta_{20}^{(1)} x_0 + \theta_{21}^{(1)} x_1 + \theta_{22}^{(1)} x_2 + \theta_{23}^{(1)} x_3 \\ \theta_{30}^{(1)} x_0 + \theta_{31}^{(1)} x_1 + \theta_{32}^{(1)} x_2 + \theta_{33}^{(1)} x_3 \end{bmatrix} \right) = \begin{bmatrix} a_1^{(2)} \\ a_2^{(2)} \\ a_3^{(2)} \end{bmatrix}$$

- 輸出的值則為:

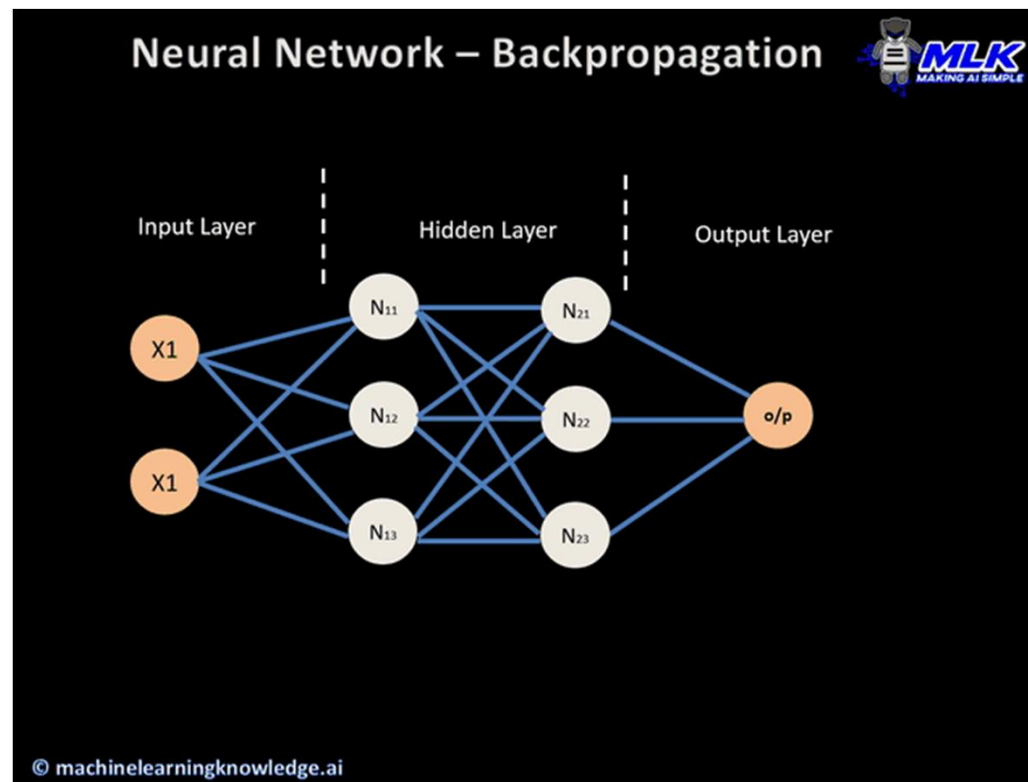
$$g \left(\begin{bmatrix} \theta_{10}^{(2)} & \theta_{11}^{(2)} & \theta_{12}^{(2)} & \theta_{13}^{(2)} \end{bmatrix} \times \begin{bmatrix} a_0^{(2)} \\ a_1^{(2)} \\ a_2^{(2)} \\ a_3^{(2)} \end{bmatrix} \right) = g \left(\theta_{10}^{(2)} a_0^{(2)} + \theta_{11}^{(2)} a_1^{(2)} + \theta_{12}^{(2)} a_2^{(2)} + \theta_{13}^{(2)} a_3^{(2)} \right) = h_{\theta}(x)$$

- $g(x)$ 圖像: sigmoid



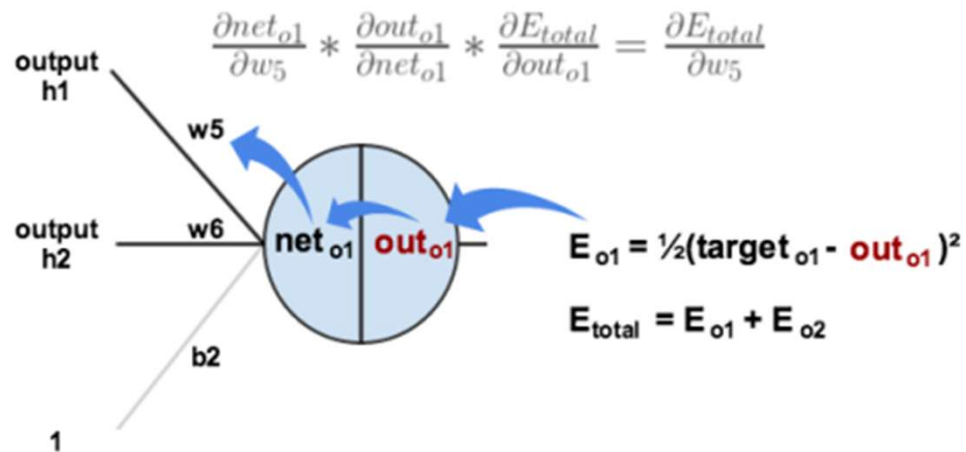
1.5 反向傳播

- 定義: 損失函數求導更新之前的係數 (ex: $\frac{1}{2}(y' - y)^2$)



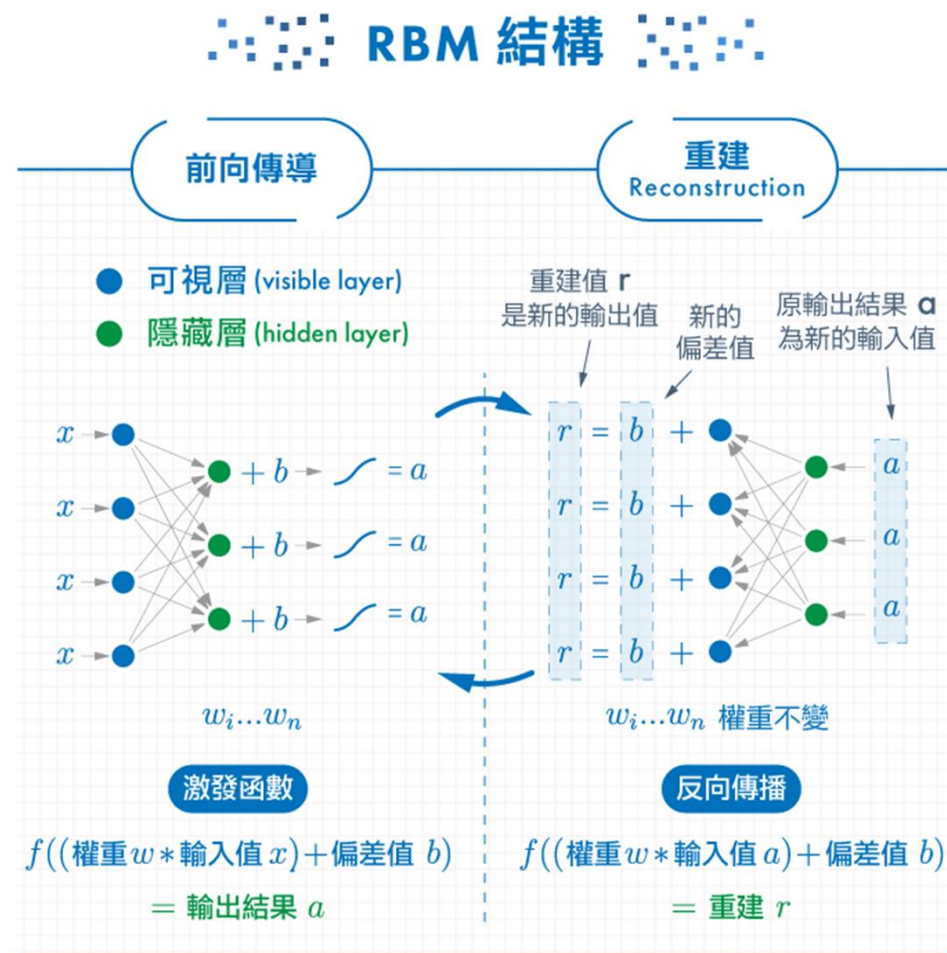
1.5 反向傳播

- 反向传播是根据链式求导法则对参数w,b更新)



1.5 反向傳播

- 整體架構:



A large red hexagon with a white slash and the number 02 inside.

/02

Python常用框架

2.神經網路框架

- 頂級深度學習框架四大陣營：
- 1.TensorFlow, 前端框架Keras, 背後巨頭Google
- 2.PyTorch, 前端框架FastAI, 背後巨頭Facebook
- 3.MXNet, 前端框架Gluon, 背後巨頭Amazon
- 4.CognitiveToolkit(CNTK), 前端框架Keras或Gluon, 背後巨頭Microsoft。

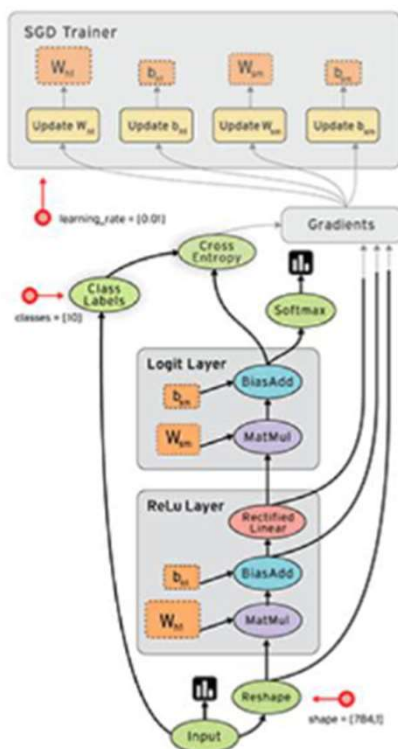
2.神經網路框架



框 架	机 构	支持语言	Stars	Forks	Contributors
TensorFlow	Google	Python/C++/Go/...	41628	19339	568
Caffe	BVLC	C++/Python	14956	9282	221
Keras	fchollet	Python	10727	3575	322
CNTK	Microsoft	C++	9063	2144	100
MXNet	DMLC	Python/C++/R/...	7393	2745	241
Torch7	Facebook	Lua	6111	1784	113
Theano	U. Montreal	Python	5352	1868	271
Deeplearning4J	DeepLearning4J	Java/Scala	5053	1927	101
Leaf	AutumnAI	Rust	4562	216	14
Lasagne	Lasagne	Python	2749	761	55
Neon	NervanaSystems	Python	2633	573	52

2.神經網路框架

- TensorflowTensorFlow是一個採用數據流圖（dataflowgraphs），用於數值計算的開源軟件庫。
- 節點（Nodes）在圖中表示數學操作，圖中的線（edges）則表示在節點間相互聯系的多維數據數組，即張量（tensor）。
- 它靈活的架構可以在多種平台上展開計算，例如台式計算機中的一個或多個CPU（或GPU），服務器，移動設備等等



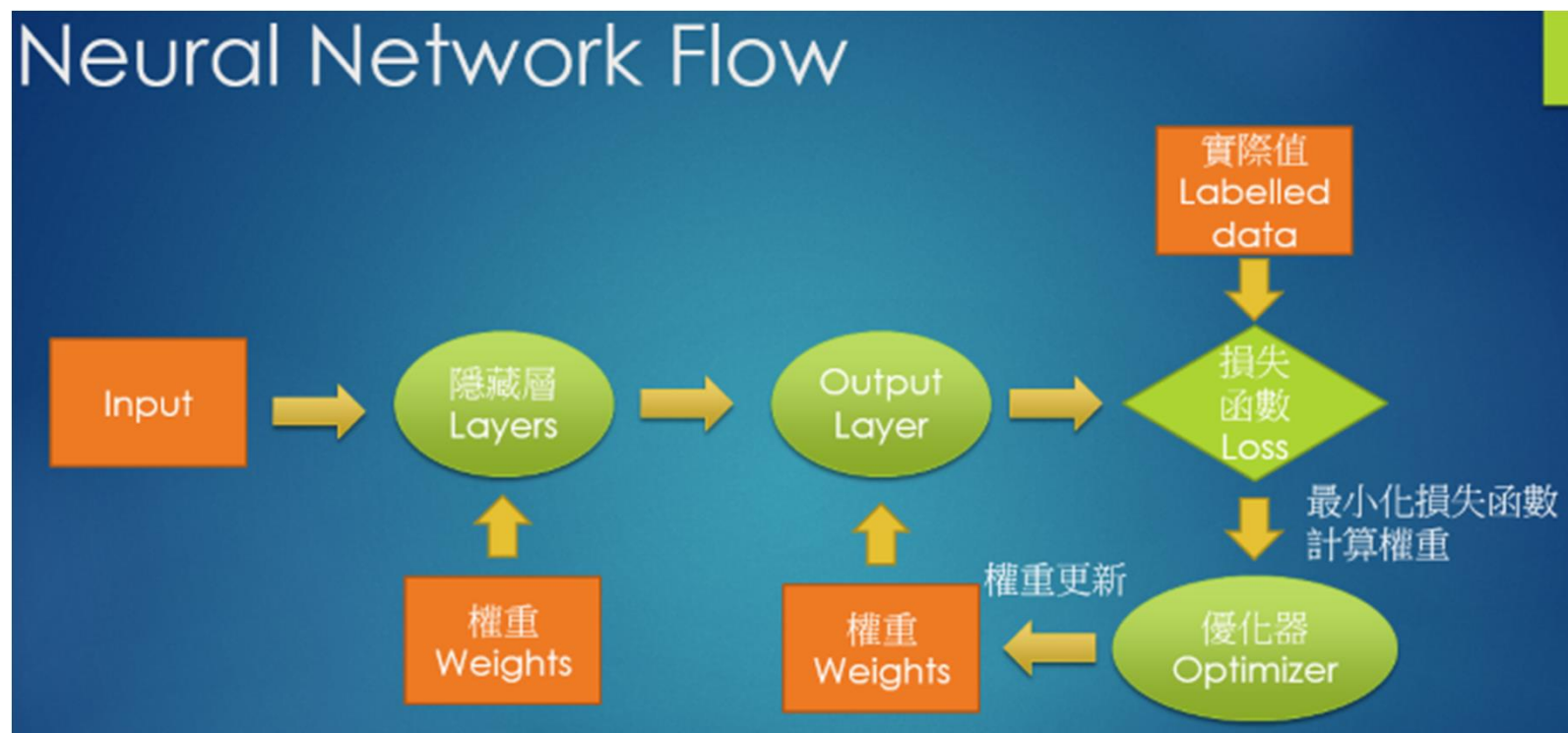
2.神經網路框架

- tensorflow
- 1.x , 2.x 版本 (新版跟舊版的接口差異大, 相容性問題多)
- keras 框架
- 高階的 api 框架, 撰寫容易
- 建立簡單的線性執行的模型

```
# 建立簡單的線性執行的模型
model = Sequential()
# Add Input layer, 隱藏層(hidden layer) 有 256個輸出變數
model.add(Dense(units=256, input_dim=784, kernel_initializer='normal', activation='relu'))
# Add output layer
model.add(Dense(units=10, kernel_initializer='normal', activation='softmax'))
```

python

2.神經網路框架



A large red hexagon with a white slash and the number 03 inside.

/03

簡單類神經網路實作 Demo

簡單類神經網路實作 Demo

- http://www.gunniliang.com/notebooks/Delete/git_r/two_month_report/202011_2021_1/11_2_to_11_6_first/code/minst_ex1.ipynb

Model: "sequential"

Layer (type)	Output Shape	Param #
=====		
flatten (Flatten)	(None, 784)	0
=====		
dense (Dense)	(None, 128)	100480
=====		
dense_1 (Dense)	(None, 10)	1290
=====		
Total params: 101,770		
Trainable params: 101,770		
Non-trainable params: 0		
=====		



/04

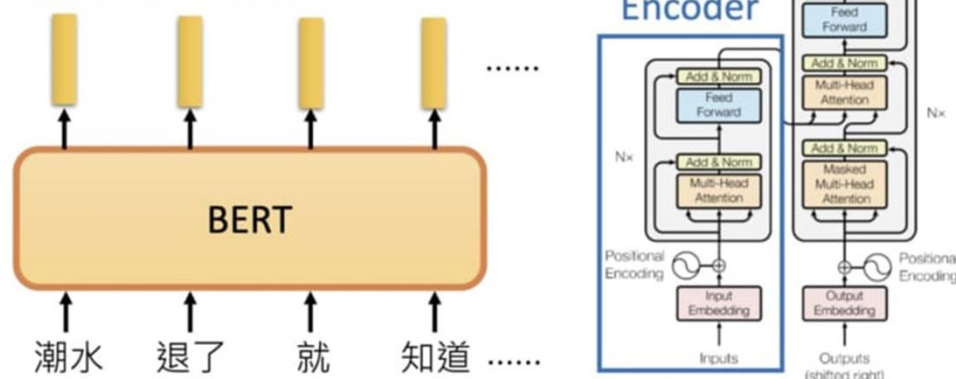
補充(10月搞炸的BERT)

BERT Bidirectional Encoder Representations from Transformers

- Ref https://leemeng.tw/attack_on_bert_transfer_learning_in_nlp.html

Bidirectional Encoder Representations from Transformers (BERT)

- BERT = Encoder of Transformer
Learned from a large amount of text without annotation





Thanks

**Next will be CNN
Network**

Sean and NN introduce