機器學習期中考整理

Created by Weber ,YC Huang 2019-11-03

1. 資料前處理

1.1 資料探勘 :

1.1.1 **KDD** 流程

資料來源 > 資料合 > 資料選取與格式改變 > 資料探勘 > 評估與呈現 >擷取知識

1.1.2 資料探勘處理原則 (SEMMA)

流程	解釋
Sample	資料取樣
Explore	視覺化且簡單說明資料屬性
Modify	選取變數並改變其呈現方法
Model	使用不同統計或機器學習方法
Assess	估準確度與最有用的模型

1.1.3 資料探勘傳遞途徑

資料收集 -> 資料預處理(特徵值提取、資料清理與整合) -> 分析處理 -> 輸出結果

1.2 結構資料處理 :

1.2.1 Dirty data :

- 不完整 : 收集資料時沒有正確收到資料、人為軟硬體錯誤
- 不一致 : 不同的資料來源
- 資料嘈雜 : 資料收集與轉換出問題

1.2.2 資料預處理主要任務

任務	說明
清理 (Cleaning)	填滿遺漏值、處理noisy data、去除離群值、解決資料不一致性
整合 (Integration)	結合不同資料庫或不同資料來源之資料，去除冗贅資料
轉化 (Transformation)	資料正規化(如，標準化資料)
減少 (Reduction)	去除無用之資料變數(如，特徵值選取，主成分分析，抽樣等)
離散化 (discretization)	對數值資料較有用

1.3 非結構資料處理：

1.3.1 斷詞處理

處理	說明
Tokenization	斷詞，將文字序列斷開為最小語意單位
Token	最小語意單位詞彙
Type	各種獨立(unique) 的token
Term	移除停用詞與正規化之後的詞彙
Stop word	停用詞，文件常出現的文字，對於資料分析無幫助，如介係詞，可藉由詞頻排序找出停用詞

中文斷詞工具 : CKIP, Stanford word segmenter, Jieba, Monpa

其他處理技術 :

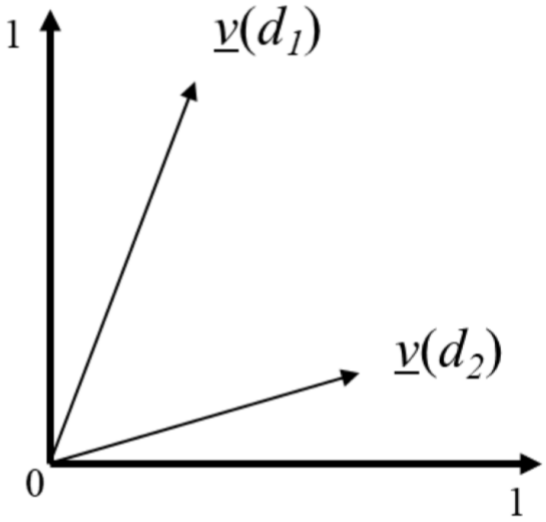
- 正規化token : 如online, on-line; USA, U.S.A。以及處理同義詞
- Stemming and lemmatization : 將詞彙導向其原形，如 cars, car's -> car

1.3.2 語言統計學三大定律 :

定律	說明
Heap's law	估計給定語料token中之term數 $M = kT^b$ ，b約為0.5，k介於30至100，若給定文件數上升字典大小會上升
Zipf's law	term的分布，Rank * Frequency 約為常數，所以通常排名第二的term詞頻約為排名第一之一半。Zipf's law 衍生意涵 : (1) 人類常用的詞彙非常少 (2)不常用的、頻率低詞彙很多
Benford law	-

2. 資料表示

2.1 Vector space model



2.1.1 餘弦相似性 (Cosine Similarity)： 通過計算兩個向量的夾角餘弦值來評估他們的相似度。 在文本分析上，文檔餘弦相似性範圍為 0-1。分母為單位向量距離相乘，分子為向量內積。餘弦越趨近1代表夾角越接近0，表示兩個向量越相似。

2.1.2 預處理與非結構資料表式方法

主要任務：

- 收集文本
- 文本斷詞
- 語言預處理
- 索引出每個詞出現的文本

2.2 詞袋模型 BOW model

一個文檔可以用一個裝著這些詞的袋子來表示，這種表示方式不考慮文法以及詞的順序。

例如，Mary is quicker than John == John is quicker than Mary

2.2.1 TF-IDF

概念	說明
Term Frequency (TF)	詞頻，基於文本中詞彙出現頻率之權重
Inverse Document Frequency (IDF)	逆向文檔頻率，詞彙之 IDF 可以由總文件數目除以包含該詞彙之文件的數目，再取以10為底的對數得到，公式：$IDF = log\frac{N}{df}$
TF-IDF	$tfidf = tf \times idf$

2.3 詞彙嵌入 Word embedding

應用於詞彙的映射，將單詞從原先所屬的空間映射到新的多維空間中，也就是把原先詞所在空間嵌入到一個新的空間中去。

2.3.1 CBOW & Skip-gram

CBOW是已知當前詞的上下文，來預測當前詞，而Skip-gram則相反，是在已知當前詞的情況下，預測其上下文。

2.3.2 ELMo & BERT

2.3.3 Word2Vec

Google 於 2013 年由 Tomas Mikolov 等人所提出，透過學習大量文本資料，將字詞用數學向量的方式來代表他們的語意。並將字詞嵌入到一個空間後，讓語意相似的單字可以有較近的距離。 Word2Vec 主要包含 CBOW & Skip-gram 兩種模型。

3. 分類基礎

3.1 Supervised Learning

應用於情緒分析、影評分析、文本主題分類

3.1.1 Labeling : 為訓練資料標註類別之處理流程，多應用於機器分類

3.1.2 訓練流程 : 匯入**訓練資料**，使用**演算法**或**模型**學習一個**分類器**，使用**測試資料**評估分類器優劣。

公式 : $\Gamma(D) = \gamma$ (D 訓練集，γ 分類器)

3.1.3 Binary Classification & Multi-class Classification : 將資料分為兩類別或多類別

3.1.4 One vs all : 需要為每一個類別分別建立一個唯一的二分法分類器，屬於此類的所有樣本均為+1，其餘的全部為-1，直到所以的類別都有當作+1類為止。最後看decision value哪個比較大，資料就判給哪一類。

3.1.5 One vs one : 對於一個K類多分類問題，訓練 $K(K-1)/2$ 個二分類分類器；每一個二分類分類器從初始多分類訓練集中收集其中兩個類別的所有樣本，並學習去區分這兩個類別。 在預測時，會有一個投票：所有 $K(K-1)/2$ 個二分類分類器被應用於一個未知樣本，並且那個得到最多「+1」預測的類別會成為最終的多分類預測結果。

3.2 Performance Evaluation of Classification (必考)

3.2.1 The evaluation metrics :

- Precision & Recall (查準率 與 查全率) :

contingency table	Relevant	Not relevant
Retrieved	# of true positives (tp)	# of false positives (fp)
Not retrieved	# of false negatives (fn)	# of true negatives (tn)

$$\text{Precision} = \frac{\#(\text{relevant item retrieved})}{\#(\text{retrieved items})} = \text{tp} / (\text{tp} + \text{fp})$$

有搜到且相關的比例

$$\text{Recall} = \frac{\#(\text{relevant item retrieved})}{\#(\text{relevant items})} = \text{tp} / (\text{tp} + \text{fn})$$

有相關且有搜到的比例

> Recall 通常只會越查越高，與 Precision 不同 (檢索越多次查準率下降)，Recall 越高 Precision 通常越低

指標	公式
Accracy	$Acc = \frac{tp+tn}{tp+fp+fn+tn}$
F1	$F1 = \frac{2PR}{P+R}$

> 如果資料偏態過大，會造成無相關文件太多，不適用 Accracy；通常檢視結果，會比較F1，越大結果越好

Precision, Recall, F1 值都介於 0-1 之間，可以用百分比表示

訓練資料通常某類別越大，該類別結果會越好。所以我們可以透過降低大類別數量，或是增加小類別資料數來精準化結果；

平均	說明
Macro-Aceraging	所有類別的每一個統計指標值的算數平均值
Micro-Aceraging	數據集中的每一個示例不分類別進行統計建立全局混淆矩陣，然後計算相應的指標。
Weighted-Aceraging	依據數據集數量做加權運算

- Macro-Aceraging
  - $Pmacro = \frac{1}{n}\Sigma Pi$
  - $Rmacro = \frac{1}{n}\Sigma Ri$
- Micro-Aceraging
  - $Pmicro = \frac{\Sigma TPi}{\Sigma TPi+\Sigma FPi}$
  - $Rmicro = \frac{\Sigma TPi}{\Sigma TPi+\Sigma FNi}$
- Weighted-Aceraging
  - E.g. $\frac{1}{3} \times P1 + \frac{2}{3} \times P2 = Pweighted$

> Micro-Aceraging 情形下，大類別常常把持小類別，若要有效看出小類別結果，要應用 Macro-Aceraging

3.3 Training Algorithm Evaluations

3.3.1 切記不要使用相同的資料集，同時當作訓練與測試用資料，會發生 over-trained 問題，以後分類器難以套用於其他資料使用

3.3.2 K-fold cross-validation :

將訓練集分割成*k*個子樣本，一個單獨的子樣本被保留作為驗證模型的數據，其他*k* − 1個樣本用來訓練。交叉驗證重複*k*次，每個子樣本驗證一次，平均*k*次的結果或者使用其它結合方式，最終得到一個單一估測。這個方法的優勢在於，同時重複運用隨機產生的子樣本進行訓練和驗證，每次的結果驗證一次，10次交叉驗證是最常用的。

3.3.3 Leave-one-out cross-validation :

只使用原本樣本中的一項來當做驗證資料，而剩餘的則留下來當做訓練資料。這個步驟一直持續到每個樣本都被當做一次驗證資料。

3.4 Naive bayes classification

Naive bayes classification 是一個機率學習模型， 計算給定樣本的每個類別的機率，然後輸出機率最高的樣本類別。會稱為 Naive bayes 因為模型的獨立性假設非常天真。但由於其有效性、便利性使其目前還是非常受歡迎，是文本分類的基準模型。

公式 : $C_m = arg\,max \frac{P(c)P(d|c)}{P(d)}$ ， P(c) 先驗機率(label) ； P(d|c) 後設機率

- Bernoulli Naive Bayes : 博努力簡單貝葉演算法主要用於處理二元特徵資料**(Binary feature)**，例如 : 0, 1。
- Multinomial Naive Bayes : 多項式簡單貝葉演算法主要用於**離散資料 (Discrete data)**，例如，電影評分。這些離散資料會有特定頻率計數。
- Gaussian Naive Bayes : 高斯簡單貝葉演算法主要用於**連續型資料 (Continuous data)**。

假設 : term 與給定類別之間是獨立的；term 的條件機率與文檔中位置也是獨立的(忽略位置，cf. Bag-of-word)

3.5 k-Nearest Neighbor

k是一個用自定義的常數。一個沒有類別標籤的向量（查詢或測試點）將被歸類為最接近該點的k個樣本點中最頻繁使用的一類。找出和新數據附近的K個鄰居(資料)，這些鄰居是哪一類(標籤)的它就是哪一類

- 優點 : 可以用於非線性分類 、 訓練時間複雜度比SVM的演算法低 、 和NB的演算法比，對資料沒有假設，準確度高
- 缺點 : 樣本不平均的時候，對稀有類別的預測準確率低 、 一般數值很大的時候不用這個，計算量太大。但是單個樣本又不能太少，否則容易發生誤分、 最大的缺點是無法給出數據的內在含義

3.6 Decision Tree

樹狀結構是由自上而下的遞歸分類方式構造，變數屬性需是類別型態。根據所選屬性遞歸劃分，直到所有屬性都被歸類至相同的子節點為止。

- 優點 : 直觀的決策規則、可以處理非線性特徵、考慮了變量之間的相互作用
- 缺點 : 訓練集上的效果高度優於測試集，即過擬合(overfitting, cf. Random Forest)、 沒有將排名分數作為直接結果

https://kknews.cc/tech/gvage8.html

- Select the attribute with the highest information gain
- Let p_i be the probability that an arbitrary tuple in D belongs to class C_i, estimated by $|C_{i,D}|/|D|$
- Expected information (entropy) needed to classify a tuple in D:

$$Info(D) = -\sum_{i=1}^{m} p_i \log_2(p_i)$$

- Information needed (after using A to split D into v partitions) to classify D:

$$Info_A(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} \times I(D_j)$$

- Information gained by branching on attribute A

$$Gain(A) = Info(D) - Info_A(D)$$

- Class P: buys_computer = “yes”
- Class N: buys_computer = “no”

$$Info(D) = I(9,5) = \underbrace{-\frac{9}{14}\log_2(\frac{9}{14})}_{\text{Yes}} \underbrace{-\frac{5}{14}\log_2(\frac{5}{14})}_{\text{No}} = 0.940$$

$$Info_{age}(D) = \frac{5}{14}I(2,3) + \frac{4}{14}I(4,0) + \frac{5}{14}I(3,2) = 0.694$$

age	p_i	n_i	$I(p_i, n_i)$
<=30	2	3	0.971
31…40	4	0	0
>40	3	2	0.971

age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
31…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes
>40	medium	no	excellent	no

$\frac{5}{14}I(2,3)$ means “age <=30” has 5 out of 14 samples, with 2 yes’es and 3 no’s. Hence

$$Gain(age) = Info(D) - Info_{age}(D) = 0.246$$

Similarly,

$$Gain(income) = 0.029$$
$$Gain(student) = 0.151$$
$$Gain(credit\_rating) = 0.048$$

3.7 Logistic Regression

相依變數為類別型態的回歸模型，在此模型中，使用logistic函數對描述單個試驗可能結果的機率進行建模，輸出的結果並不是一個離散值或者確切的類別。相反，得到的是一個與每個觀測樣本相關的機率列表。

- 優點：便利的觀測樣本機率分數
- 缺點：當特徵空間很大時，邏輯回歸的性能不是很好

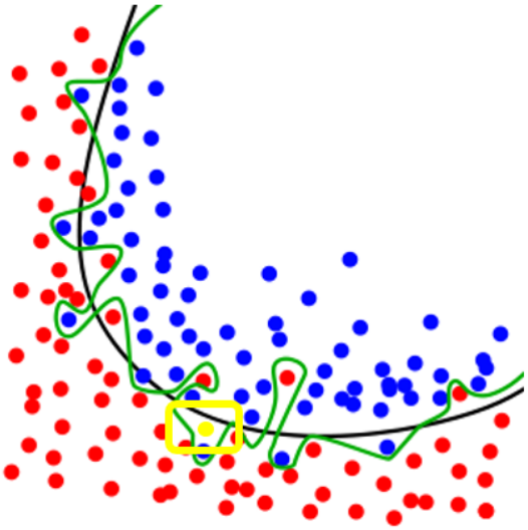
3.8 Support Vector Machine

模型特色，是依靠邊界樣本來建立需要的分離曲線，可以處理非線性決策邊界。

- 優點：能夠處理大型特徵空間(cf. LR)、 能夠處理非線性特徵之間的相互作用
- 缺點：當觀測樣本很多時，效率並不是很高、有時候很難找到一個合適的核函數

3.9 Overfitting

顧名思義就是過度學習訓練資料，變得無法順利去預測或分辨不是在訓練資料內的其他資料。



有個方法可以偵測是否有Overfitting的情況發生， 將所有的Training data坼成二部分，一個是Training Set跟Validate Set，Training Set就是真的把資料拿去訓練的，而Validate Set就是去驗證此Model在訓練資料外的資料是否可行。

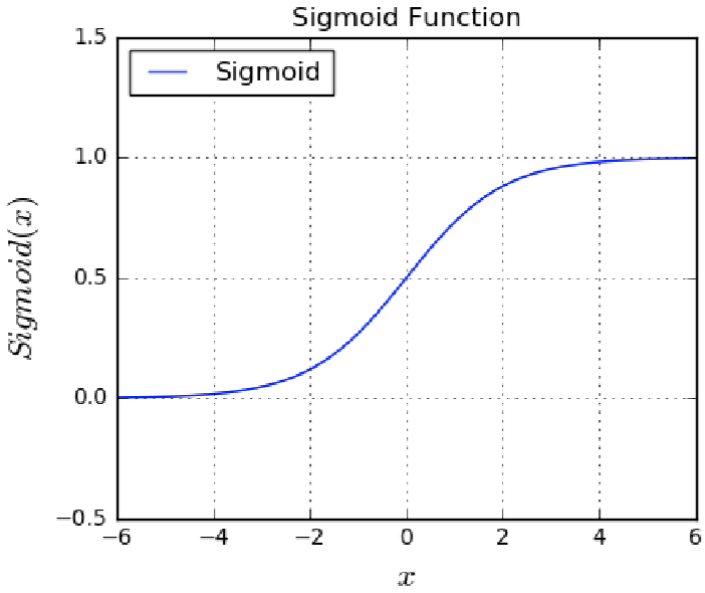
4. 深度學習

4.1 Activation Function

4.1.1 Sigmoid

範圍 0~1

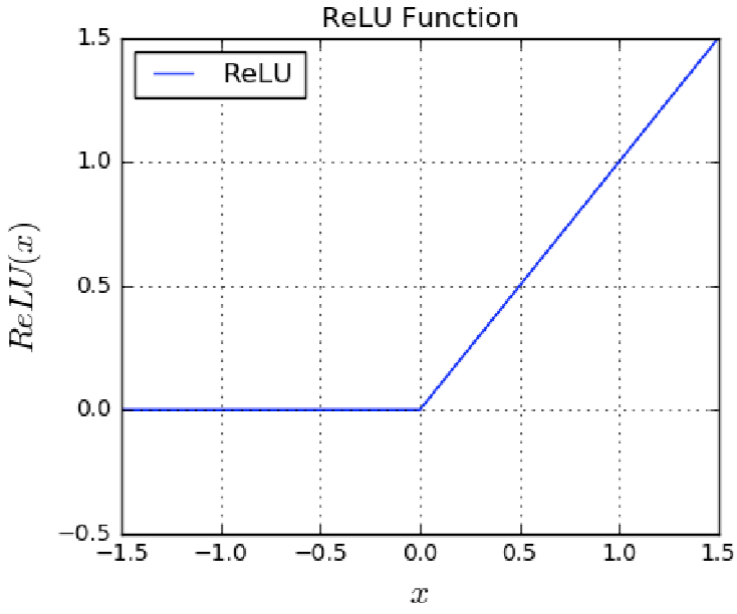
公式 : $f(x) = \frac{1}{1+e^{-}x}$



4.1.2 reLU

範圍 0~x

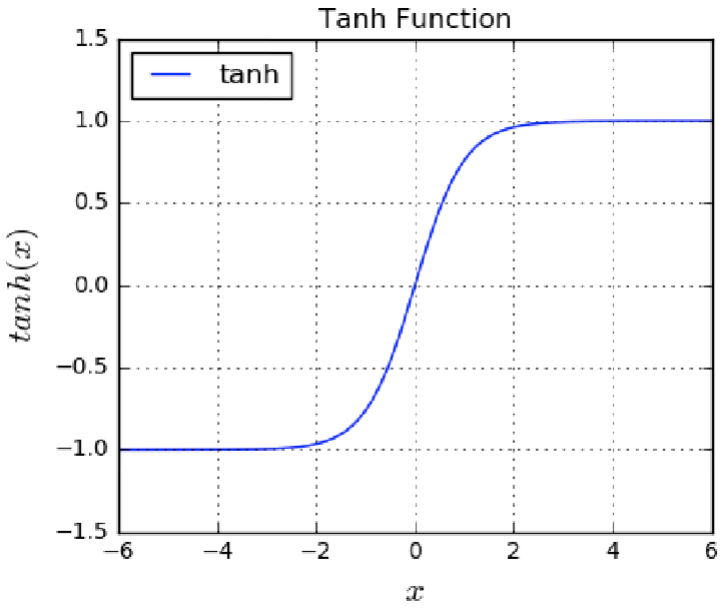
公式 : $f(x) = max(0, x)$



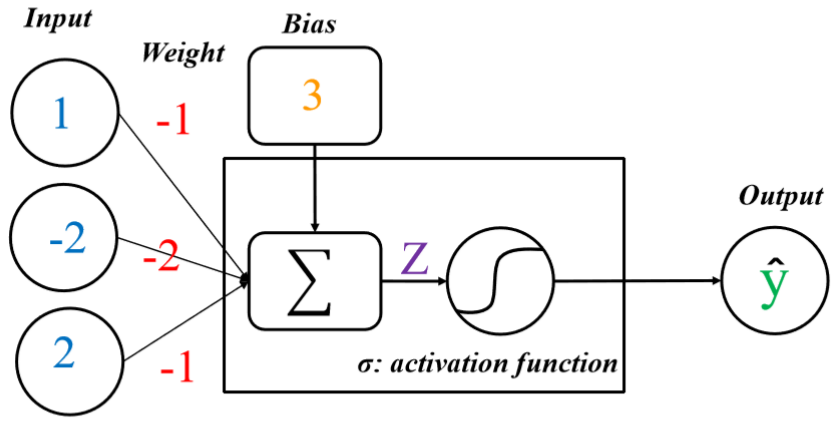
4.1.3 tanh

範圍 -1~1

公式：$f(x) = \frac{sinhx}{coshx} = \frac{e^x - e^-x}{e^x + e^-x}$

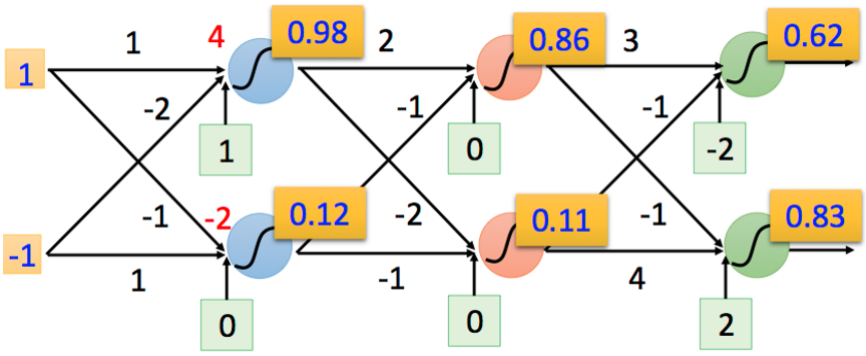


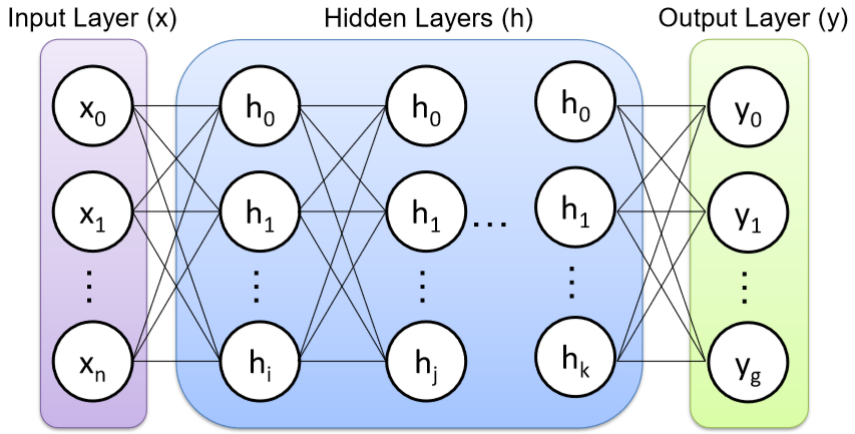
Forward Calculation (1/2)



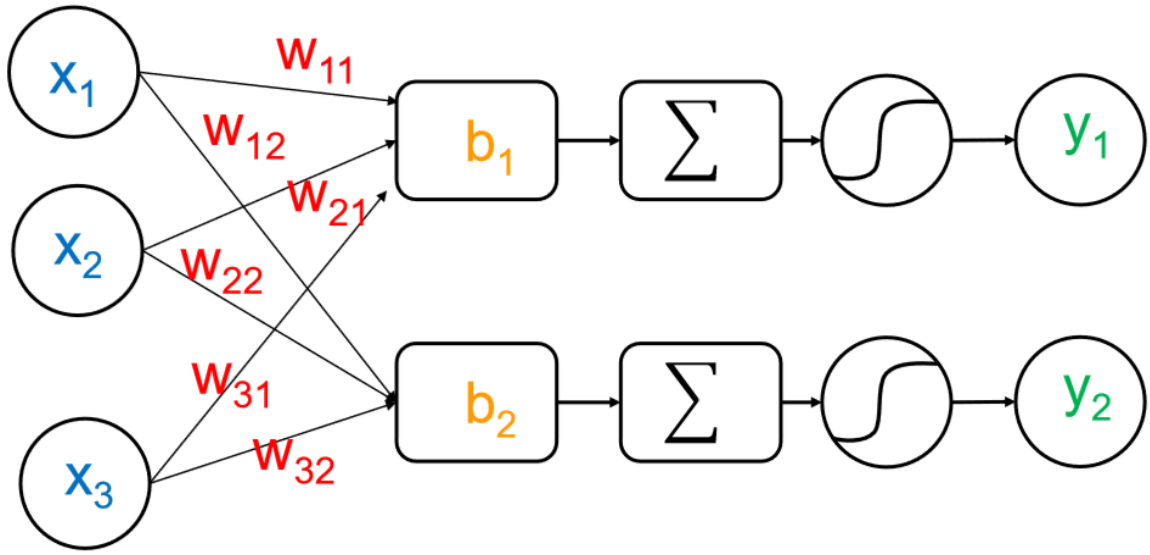
$$Z = (1 * (-1) + (-2) * (-2) + 2 * (-1) + 3)$$

$$y = \sigma(4) = \frac{1}{1.0183} = 0.98$$





4.2 Computation of metrix



$$y_1 = activation\ function(x_1 * w_{11} + x_2 * w_{21} + x_3 * w_{31} + b_1)$$
$$y_2 = activation\ function(x_1 * w_{12} + x_2 * w_{22} + x_3 * w_{32} + b_2)$$

$$[y_1, y_2] = activaton([x_1, x_2, x_3] \times \begin{bmatrix} W11 & W12 \\ W21 & W22 \\ W31 & W32 \end{bmatrix} + [b_1 b_2])$$

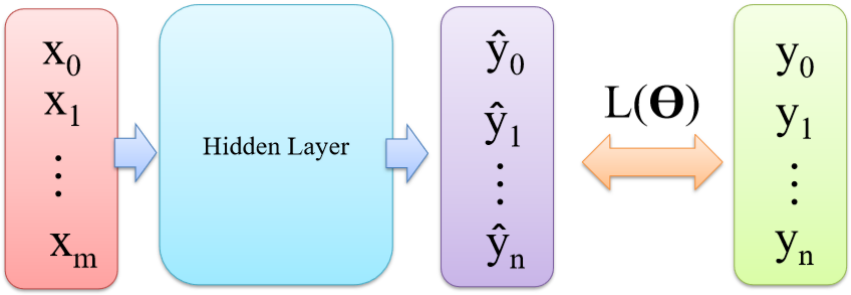
$$Y = activation(X \times W + b)$$

$$output = activation function(input \times weighting + bias)$$

4.3 How to evaluation model ?

4.3.1 loss function (L(θ)):

損失函數是量化網絡輸出與實際值之間的差距


x_0
x_1
x_m
Hidden Layer
$\hat{y}_0$
$\hat{y}_1$
$\hat{y}_n$
$L(\boldsymbol{\Theta})$
y_0
y_1
y_n