Logistic Regression

Jen-Ing Hwang
Department of Computer Science and
Information Engineering
Fu Jen Catholic University

Outline

- Introduction
  - Three Models for Binary Classification
  - 這三個都可以做二元分類
- Activation Function
  - Logistic Sigmoid Function
  - 注意它的觸發函數是誰&它的錯誤函數是誰將會決定它在更新w時會不一樣
- Error Function
  - Cross-Entropy Loss Function
- Review Gradients and Directional Derivatives
- Learning Algorithm
  - Gradient Descent Method

Linear regression
主要是做迴歸
但也可以做分類
若data裡面只有+1跟-1，也是
實數裡面的兩個值

Introduction

- Logistic regression
  - Like all regression analyses, logistic regression is a predictive analysis. 目的：預測一個值，跟linear regression是一樣的道理，差別是Logistic regression是預測機率，所以界在0~1之間，比較靠近1是第一種現象，比較靠近0是第二種現象
  - It predicts the probability of the occurrence of a binary outcome.預測機率!!!
  - It transforms its output using the logistic function to return a probability value.
  - The error function for logistic regression is the cross-entropy.
  - It is used to explain the relationship between binary output label and one or more input attributes.

Three Models for Binary Classification (1/2)

Recap so far

	Activation Function	Algorithm	Stop Criteria	Remarks
Linear Classification	Sign function	Perceptron Learning Algorithm	Stop when linearly separable examples are separated.	Converge for linearly separable problems; fail to converge for linearly inseparable problems.
Linear Regression (can be used in classification 也可以拿來做分類)	Linear function 預測的是一個實數值 ex:房子的價錢	Gradient Descent Method	Stop when the mean squared error is small enough.	Converge to hypothesis with minimum squared error; sensitive to outliers.

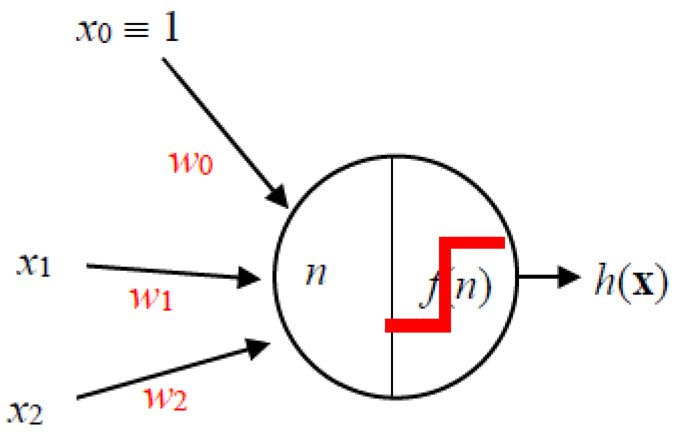
Linear Regression一開始不是拿來做分類，它是實數值

Three Models for Binary Classification (2/2)

Three simple LINEAR models for binary classification.

Linear Classification

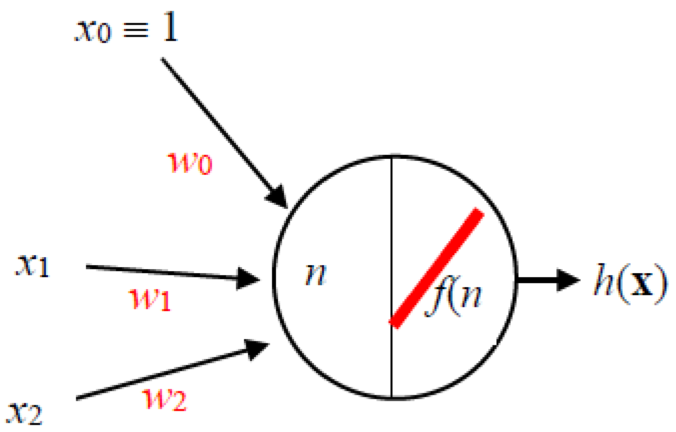
$h(\mathbf{x}) = f(n) = \text{sign}(n)$



Linear Regression

直線，斜率是1，通過原點

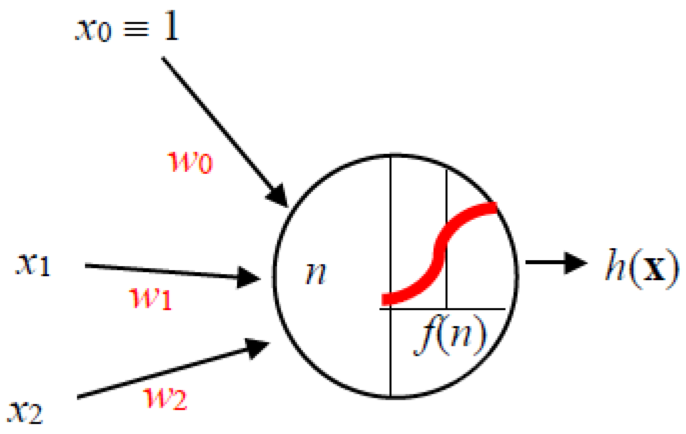
$h(\mathbf{x}) = f(n) = n$



Logistic Regression

預測是某個東西的機率有多大(就是在分類)

$h(\mathbf{x}) = f(n) = \sigma(n)$

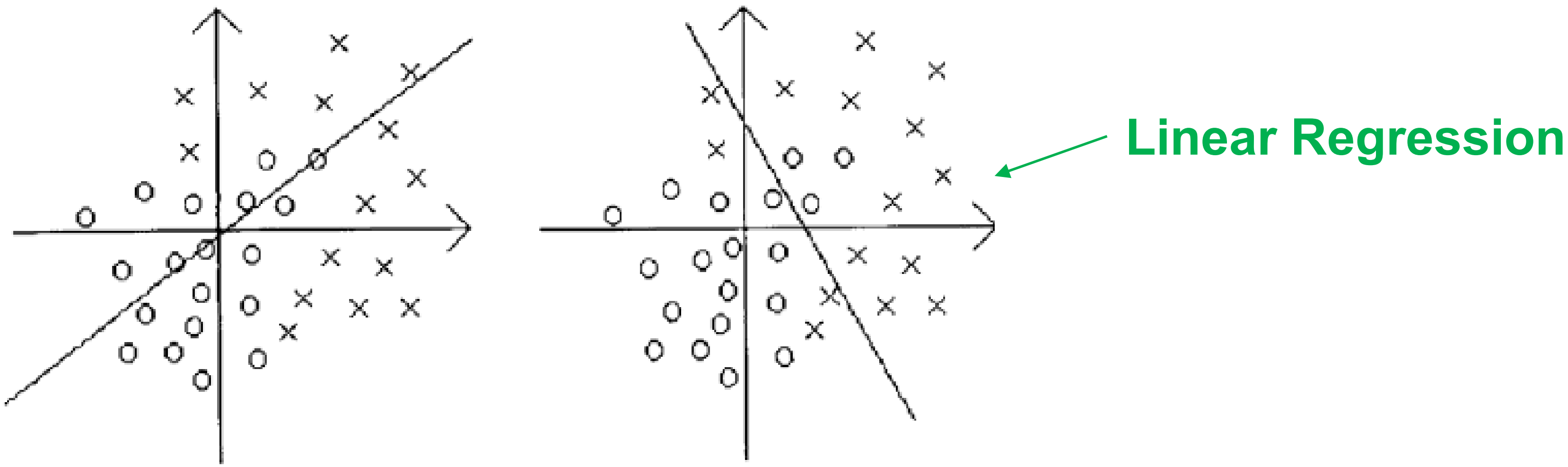


$\boxed{\text{Net input: } n = w_0 x_0 + w_1 x_1 + w_2 x_2}$

$\boxed{\text{Activation Function: } f(n)}$

Linear Classification Vs. Linear Regression

In the case of linearly INSEPARABLE(不可分離) problem

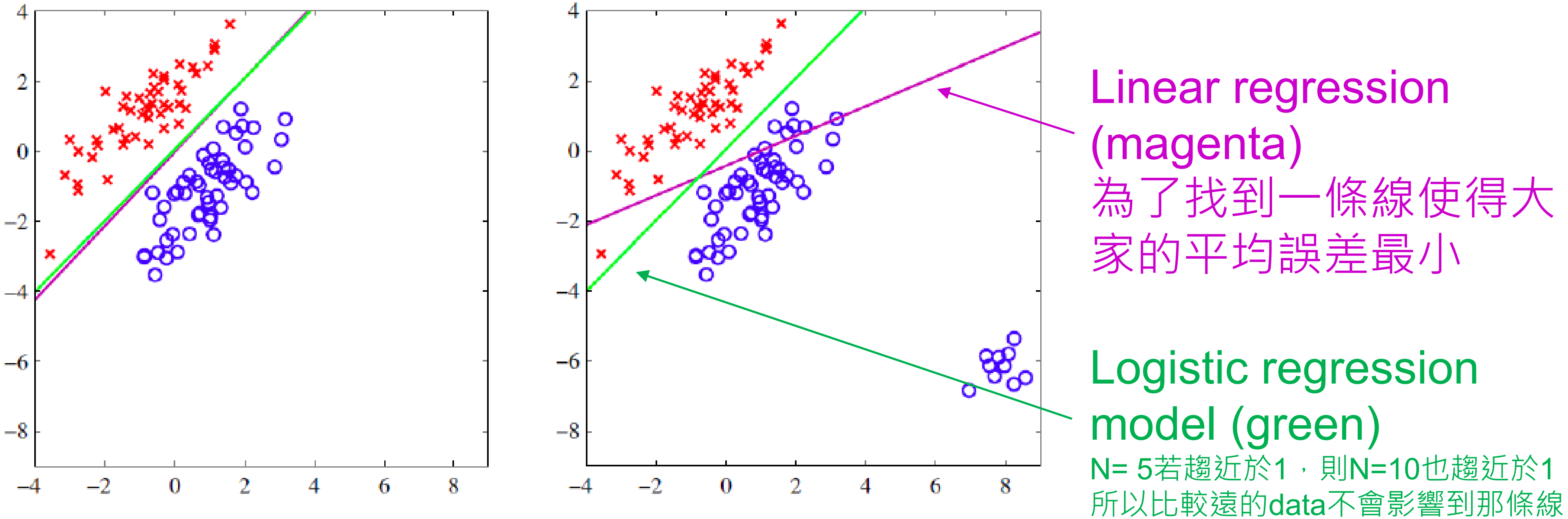


Perceptron Learning Algorithm fails to converge
如果遇到不可線性分離的，它不會停下來，只好用最大世代去停下來

Gradient Descent Method converges toward a best-fit approximation
還是可以找到最靠近的一條線
沒辦法分類，但可以找到誤差最小的，最吻合的

Linear Regression Vs. Logistic Regression (1/2)

In the case of linearly separable problem with outliers(離群值)(right figure)



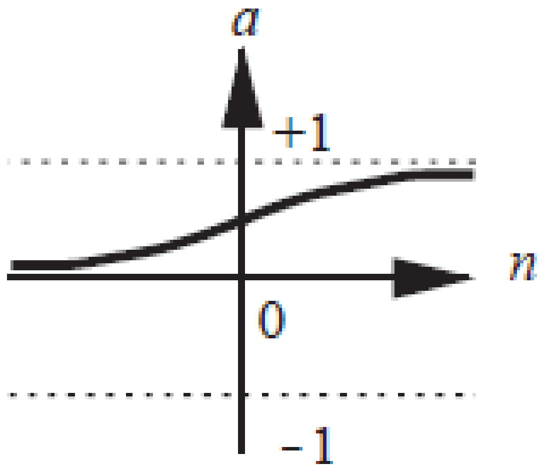
Two-class dataset (×, ○) Dataset (×, ○) with outliers

Linear regression is sensitive to outliers, but logistic regression, as well as linear classification (not shown in the figure), are usually not.

Figure adapted from Fig 4.4 in Ref. [3]

Logistic sigmoid function (1/2) 在算機率

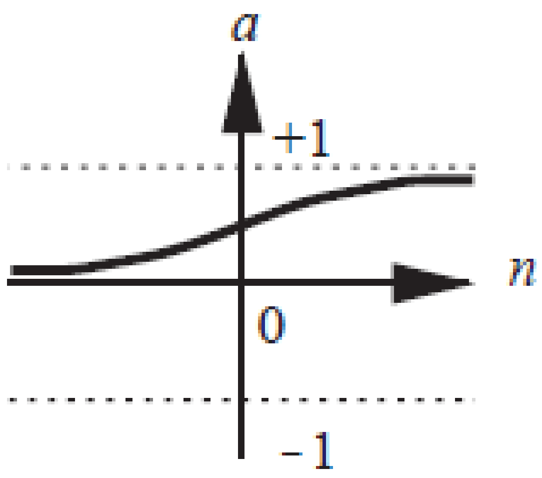
$$f(n) = \sigma(n) = \frac{1}{1+e^{-n}}$$



- $\sigma(0) = \frac{1}{1+\cancel{e^{-0}}_{1}} = 0.5$
- If n goes to infinity,
  then $\sigma(n) = \frac{1}{1+\cancel{e^{-n}}_{0}} \cong 1$
- If n goes to negative infinity,
  then $\sigma(n) = \frac{1}{1+\cancel{e^{-n}}_{\infty}} \cong 0$
- $\sigma(n) \in (0,1)$

Logistic sigmoid function (2/2)

- $f(n) = \sigma(n) = \frac{1}{1+e^{-n}}$
  - also known as sigmoid, logistic, or squashing function.
- The output value is continuous and between 0 and 1.
  - can be interpreted as a probability (機率)
- Note that $1 - \sigma(n) = \sigma(-n)$
  - Or $\sigma(n) + \sigma(-n) = 1$
  - Example: $\sigma(0.2) \cong 0.55$, $\sigma(-0.2) \cong 0.45$
- Also note that $\frac{\sigma(n)}{dn} = \sigma(n)(1 - \sigma(n))$



Logistic Regression for Binary Classification (1/2)

- A binary classification problem:

Students' Performance Dataset

Students	x_1 (Midterm)	x_2 (Final)	y (Pass/Fail)
A	80	60	1
B	50	50	0
C	90	80	1
D	30	60	0
E	40	90	1
F	90	50	1

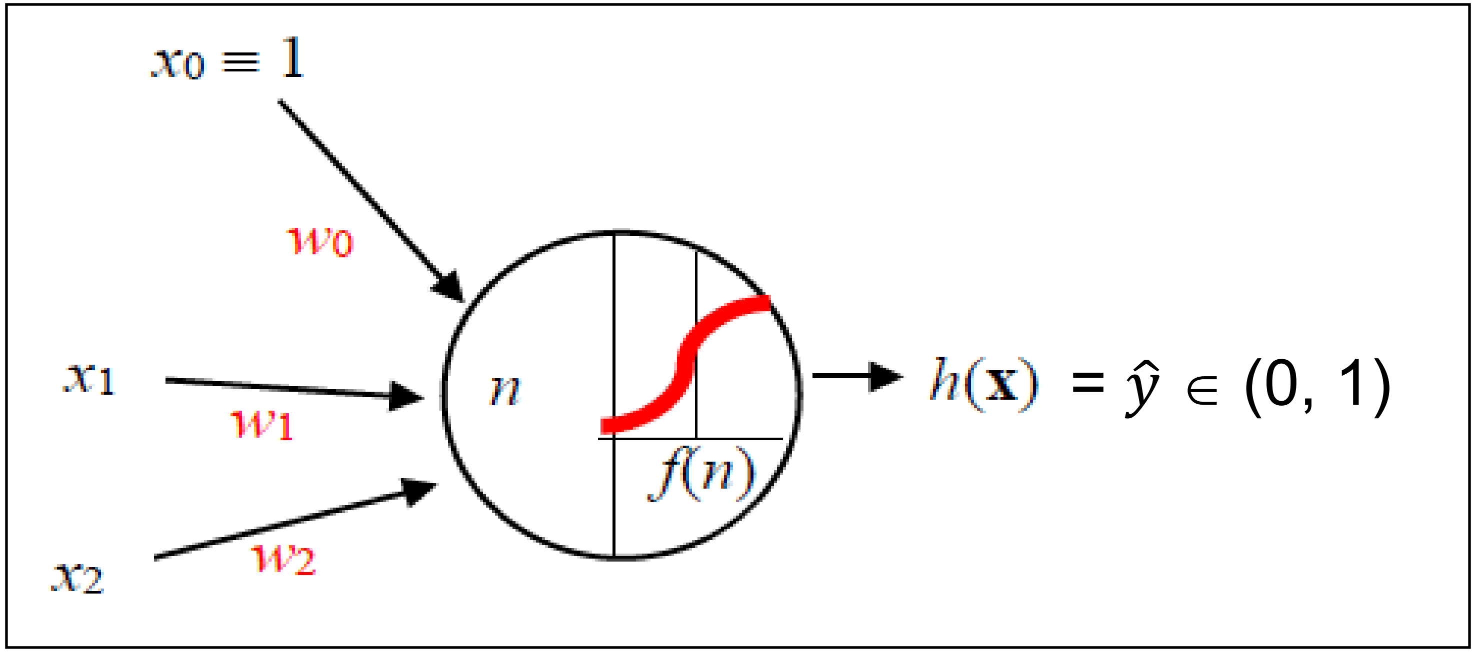
- Input: $\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} \text{Midterm} \\ \text{Final} \end{bmatrix}$
- Output: y = 1 or 0
- $h(\mathbf{x}) = \sigma(w_0 x_0 + w_1 x_1 + w_2 x_2)$

  $= \sigma(n) = f(n) \in (0, 1)$

  : Interpreted as a probability
- Given **x**, we want to predict
- $P(y|\mathbf{x}) = \begin{cases} \sigma(n), & \text{for } y = 1 \\ 1 - \sigma(n), & \text{for } y = 0 \end{cases}$

Logistic Regression for Binary Classification (2/2)

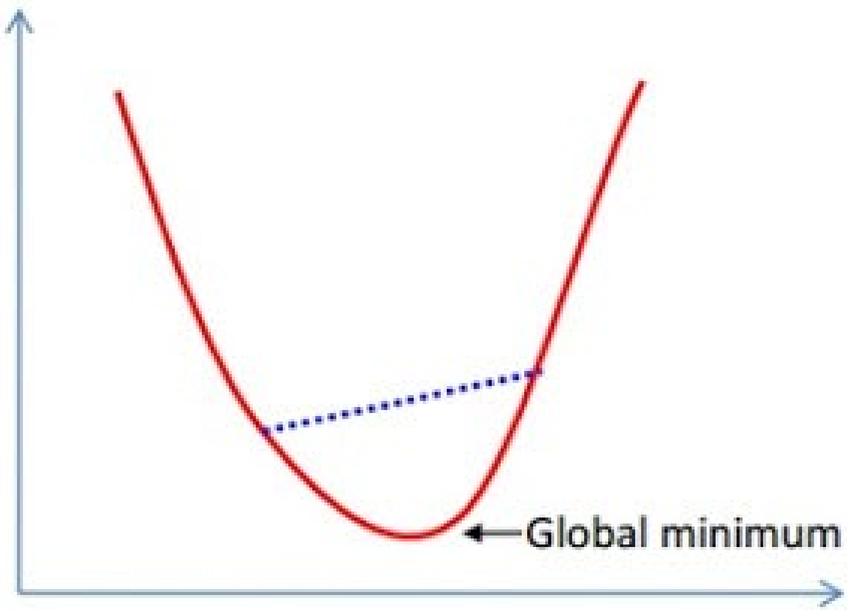
$$h\,(\mathbf{x}) = f(n) = \sigma\,(w_0 x_0 + w_1 x_1 + w_2 x_2) = \hat{y}$$



Interpret
$\hat{y} = \sigma\,(n) = \mathrm{P}(y=1|\mathbf{x})$
(條件機率：給定這張相片，他是貓的機率有多大，y = 1代表貓，x = 照片)

If $y = 1$: $\mathrm{P}(y|\mathbf{x}) = \hat{y}$
If $y = 0$: $\mathrm{P}(y|\mathbf{x}) = 1 - \hat{y}$

Note that the "logistic regression" is a model for classification(分類) rather than regression, though the model has the term "regression."



Convex function

Non-convex function

Activation： $\hat{y}$ = $\sigma(n)$
Error function：cross-entropy是convex
(它的Local minimum就是Global minimum)

$\frac{1}{2}(y-g)^2$ (sigmoid用half square是Non-convex)所以不敢保證是找到Global minimum

Error Measure for Logistic Regression (1/2)

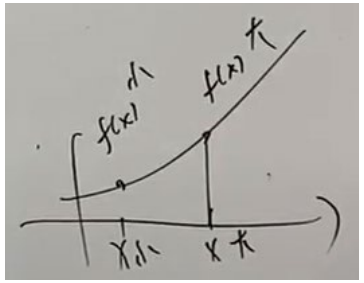
Our goal: $\boxed{\hat{y} = \mathrm{P}(y = 1|\mathbf{x})}$

$\Rightarrow$ Or we can say:
$$\boxed{\begin{array}{l} \text{If } y = 1\text{: } \mathrm{P}(y|\mathbf{x}) = \hat{y} \\ \text{If } y = 0\text{: } \mathrm{P}(y|\mathbf{x}) = 1 - \hat{y} \end{array}}$$

$\Rightarrow$ Equivalent to: $\boxed{\mathrm{P}(y|\mathbf{x}) = \hat{y}^{y}(1 - \hat{y})^{(1-y)}}$ Eq (*)

$$\left(\begin{array}{l} \text{If } y = 1\text{: } \mathrm{P}(y|\mathbf{x}) = \hat{y}^{1}(1 - \hat{y})^{(1-1)} = \hat{y}\,(1 - \hat{y})^{0} = \hat{y} \\ \text{If } y = 0\text{: } \mathrm{P}(y|\mathbf{x}) = \hat{y}^{0}(1 - \hat{y})^{(1-0)} = 1(1 - \hat{y})^{1} = 1 - \hat{y} \end{array} \right)$$

Error Measure for Logistic Regression (2/2)



We want to: Max (P(y|**x**))

$\Rightarrow$ We can ask for: Max (ln (P(y|**x**)))

(Since ln(•) is monotonically increasing遞增函數->把原來的數都放大

Ex：調分，把同學的分數都調高；用途：讓之後好微分)

$\Rightarrow$ Min (–ln (P(y|**x**))) (We prefer to use the term: minimize an error.我們都希望error越小越好)

Let (–ln (P(y|**x**))) be the error (cost or loss) function *E*.

Then $E(w_0, w_1, w_2)$ = –ln (P(y|**x**)) = –ln $(\hat{y}^y(1-\hat{y})^{(1-y)})$ (by Eq (*) on the previous slide)

= – (y ln $\hat{y}$ + (1 – y) ln (1 – $\hat{y}$)) (by the properties of ln)

$\Rightarrow E(w_0, w_1, w_2)$

= $\boxed{\text{– (y ln } \hat{y} \text{ + (1 – y) ln (1 – } \hat{y}\text{)): Cross-entropy loss function}}$

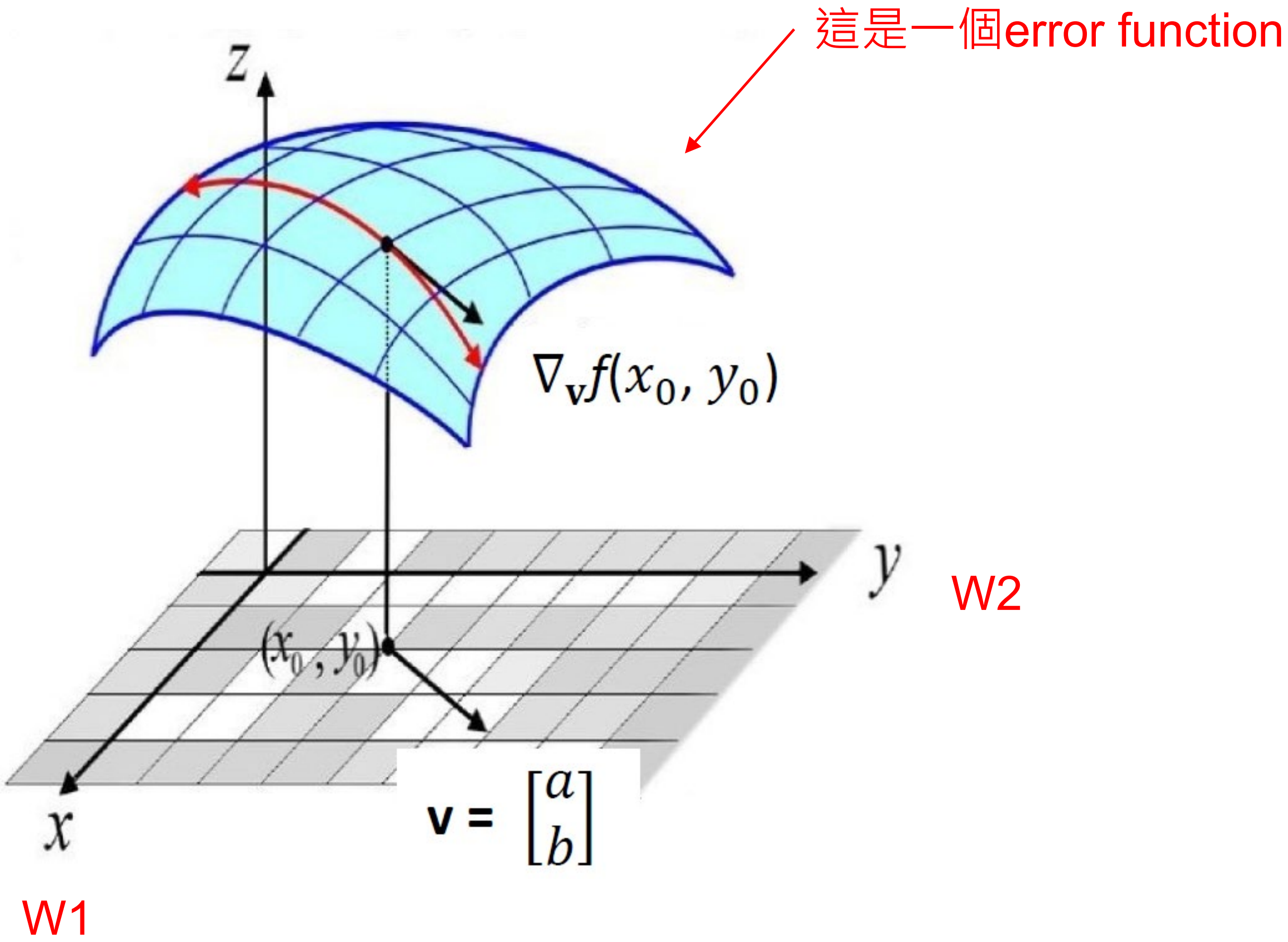
ln(x · y) = ln(x) + ln(y)

ln(x^y) = y · ln(x)

Gradients and Directional Derivatives (1/4)

Gradient：
->統籌的去看往x跟y的微分是什麼樣子，但是把它想成是一個山頂，我想要趕快的下山，我想知道怎樣可以最快的下降，因此我想知道的方向是沿著任何一個方向，不一定要往X軸或Y軸

Error function是w造成的(剛開始的，w是隨便猜的)，我隨便一開始的w就會對應到一個很高的error，那我想要趕快讓這個error下降，我要往哪個方向走呢?要找到一個V放向使得它下降最快



Gradients and Directional Derivatives (2/4)

看函數對各個維度的變化

Example: $f(x, y) = x^2y^3$

Gradient of $f(x, y) = \nabla f(x, y)$

$$= \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} = \begin{bmatrix} 2xy^3 \\ 3x^2y^2 \end{bmatrix}$$

Gradient：
看這個函數對第一個input的變化，
看這個函數對第二的input的變化

The rate of change $f(x, y)$ in direction (1, 0):

$$\frac{\partial f}{\partial x} = 1\,\frac{\partial f}{\partial x} + 0\,\frac{\partial f}{\partial y}$$

x的偏微：好像是算方向導數是沿著(1, 0)方向

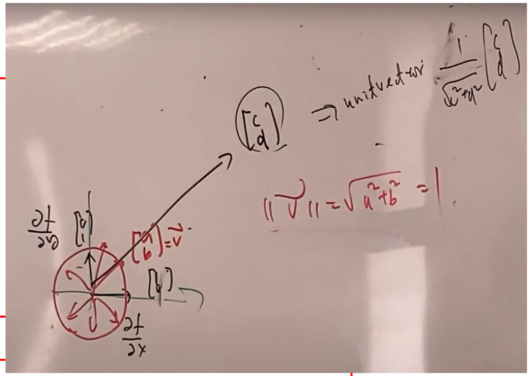
The rate of change $f(x, y)$ in direction (0, 1):

$$\frac{\partial f}{\partial y} = 0\,\frac{\partial f}{\partial x} + 1\,\frac{\partial f}{\partial y}$$

y的偏微：好像是算方向導數是沿著(0, 1)方向

Gradients and Directional Derivatives (3/4)

若今天我不是要沿著(1,0)或(0,1)我想要沿著(a, b)



Let **v** = $\begin{bmatrix} a \\ b \end{bmatrix}$ be a unit vector(單位長度的向量), then

The rate of change $f(x, y)$ in the direction **v** : $a\dfrac{\partial f}{\partial x} + b\dfrac{\partial f}{\partial y}$

The rate of change $f(x, y)$ in the direction of **v**

- called a directional derivative(方向導數)
- Denoted as $\nabla_{\mathbf{v}} f(x, y)$

如果V不是單位向量
還要除以||V||
$\|V\| = \sqrt{a^2 + b^2}$

- $\nabla_{\mathbf{v}} f(x, y) = \dfrac{1}{\sqrt{a^2+b^2}} \times \begin{bmatrix} a \\ b \end{bmatrix} \cdot \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} = \dfrac{a}{\sqrt{a^2+b^2}}\dfrac{\partial f}{\partial x} + \dfrac{b}{\sqrt{a^2+b^2}}\dfrac{\partial f}{\partial y} = \dfrac{V}{\|V\|} \cdot \nabla f$

確保這個一定是
一個長度為1的
方向

: inner product of **v** and ∇f

Gradients and Directional Derivatives (4/4)

Gradient是向量，但是算出來的值是數字，因為沿著任何一個地方下降，下降多快是一個數字。
注意!給你一個方向以後，你算出來的值一定是一個純量

The directional derivative of $f(x, y)$ in the direction of **v**

$\nabla_{\mathbf{v}} f(x, y) = \mathbf{v} \cdot \nabla f = \underbrace{|\cancel{\mathbf{v}}|}_{1} * |\nabla f| * \cos\theta = |\nabla f| * \cos\theta$

1 (因為V一定是單位向量)

先用f在帶出-f的觀念

$\Rightarrow$ The greatest value of $\nabla_{\mathbf{v}} f(x, y)$ is $|\nabla f|$ (if $\cos\theta = 1$, or $\theta = 0$)

$\Rightarrow$ The direction of greatest increase(上升最快) of f is the same direction as the gradient vector, ∇f.

這個方向就是Gradient的方向，使得這個方向變化的最快的，所以才會是最大值

我們會用到的

$\Rightarrow$ The greatest negative value of $\nabla_{\mathbf{v}} f(x, y)$ is $-|\nabla f|$ (if $\cos\theta = -1$, or $\theta = \pi$)

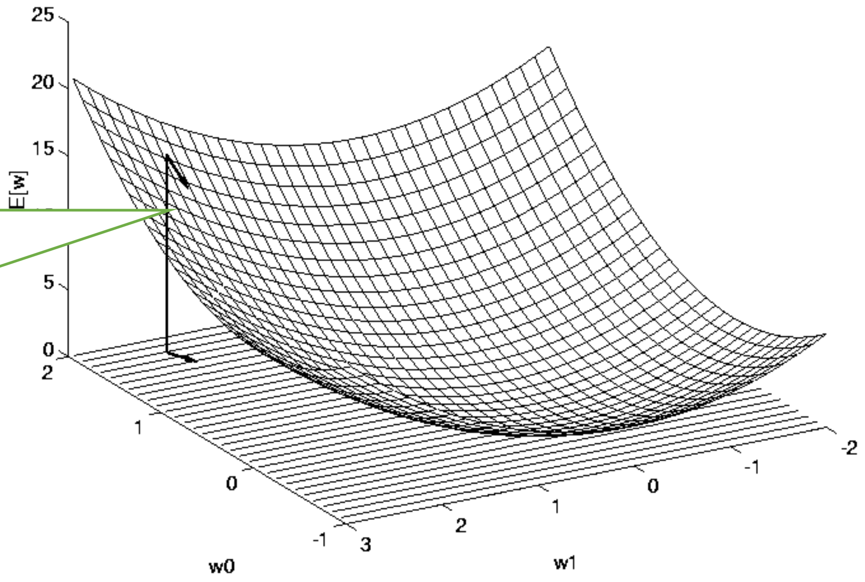
$\Rightarrow$ The direction of greatest decrease(下降最快) of f is the direction opposite(對面) to the gradient vector, $-\nabla f$.

我希望它的值是下降最快的，如果下降最快，那我的值就要最小，那就要沿著Gradient的負方向

Gradient Descent of Error Function

- Example: $E(\mathbf{w})$ is an error function, where $\mathbf{w} = (w_0, w_1)$
- Minimize $E(\mathbf{w})$ by iteratively moving in the direction of steepest descent
- Steepest descent: $-\nabla E(\mathbf{w})$

Arrow: negated gradient at one point

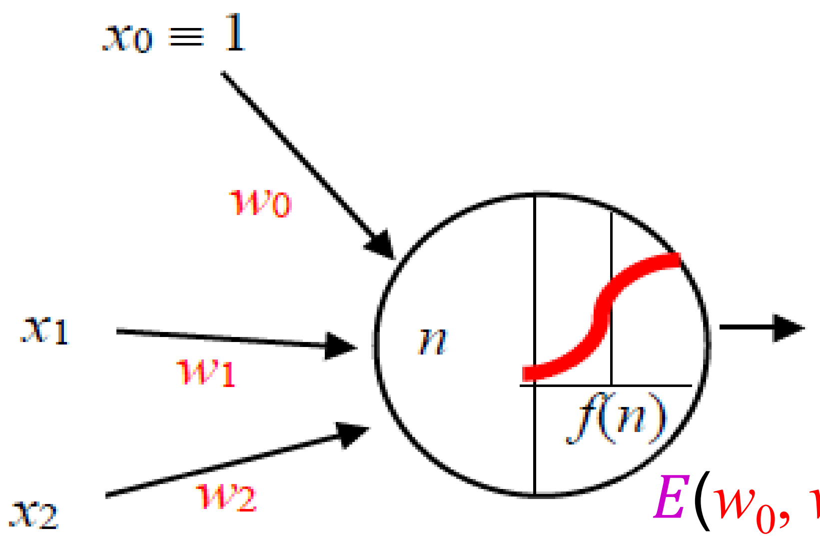


$$-\nabla E(\mathbf{w}) = -\nabla E(w_0, w_1) = \begin{bmatrix} -\dfrac{\partial E}{\partial w_0} \\ -\dfrac{\partial E}{\partial w_1} \end{bmatrix}$$

Figure adapted from Fig 4.4 in Ref. [4]

Gradient Descent Method (1/4)

$$\boxed{h(\mathbf{x}) = f(n) = \sigma(w_0 x_0 + w_1 x_1 + w_2 x_2) = a = \hat{y}}$$



$$h(\mathbf{x}) = a = \hat{y} \overset{?}{\cong} y$$

$$E(w_0, w_1, w_2) = -\left(y \ln \hat{y} + (1 - y) \ln (1 - \hat{y})\right)$$

Gradient Descent

$$\mathbf{w} \leftarrow \mathbf{w} + \eta\left(-\nabla E(\mathbf{w})\right)$$

Where $\mathbf{w} = \begin{bmatrix} w_0 \\ w_1 \\ w_2 \end{bmatrix}$

$-\nabla E(\mathbf{w})$：讓error function下降最快的方向

Gradient Descent Method (2/4)

How to find $\nabla E(\mathbf{w})$? First, find derivatives of *n*, *a*, and *E*.

Net input:
$n = w_0x_0 + w_1x_1 + w_2x_2$

Activation function:
$a = \hat{y} = \sigma(n) = \frac{1}{1+e^{-n}}$

Error function:
$E = -\left(y \boxed{\ln a} + (1-y)\boxed{\ln(1-a)}\right)$

$$\frac{\partial n}{\partial w_0} = x_0$$
$$\frac{\partial n}{\partial w_1} = x_1$$
$$\frac{\partial n}{\partial w_2} = x_2$$

$$\frac{da(n)}{dn} = a(1-a)$$

$\frac{d}{da}\ln a = \frac{1}{a}$　　$\frac{d}{da}\ln(1-a) = \frac{1}{1-a}$

$$\frac{dE(a)}{da} = -\frac{y-a}{a(1-a)}$$

神經元的輸出為什麼會變化，是因為神經元的輸入的值在做變化，所以Sigmoid應該要跟著net input做微分

Gradient Descent Method (3/4)

How to find $\nabla E(\mathbf{w})$? Next, use the chain rule to find $\frac{\partial E}{\partial w_0}, \frac{\partial E}{\partial w_1}, \frac{\partial E}{\partial w_2}$:

$$\boxed{\frac{\partial E}{\partial w_0} = \frac{dE}{da}\frac{da}{dn}\frac{\partial n}{\partial w_0} = -\frac{y-a}{a(1-a)}a(1-a)x_0 = -(y-a)\ x_0}$$

$$\boxed{\frac{\partial E}{\partial w_1} = \frac{dE}{da}\frac{da}{dn}\frac{\partial n}{\partial w_1} = -\frac{y-a}{a(1-a)}a(1-a)x_1 = -(y-a)\ x_1}$$

$$\boxed{\frac{\partial E}{\partial w_2} = \frac{dE}{da}\frac{da}{dn}\frac{\partial n}{\partial w_2} = -\frac{y-a}{a(1-a)}a(1-a)x_2 = -(y-a)\ x_2}$$

Gradient Descent Method (4/4)

$$\nabla E(\mathbf{w}) = \begin{bmatrix} \frac{\partial E}{\partial w_0} \\ \frac{\partial E}{\partial w_1} \\ \frac{\partial E}{\partial w_2} \end{bmatrix} = \begin{bmatrix} -(y-a)\ x_0 \\ -(y-a)\ x_1 \\ -(y-a)\ x_2 \end{bmatrix} = -(y-a)\begin{bmatrix} x_0 \\ x_1 \\ x_2 \end{bmatrix} = -(y-a)\mathbf{x}$$

Gradient Descent Rule

$\mathbf{w} \leftarrow \mathbf{w} + \eta\,(-\nabla E(\mathbf{w}))$

$\Rightarrow\ \mathbf{w} \leftarrow \mathbf{w} + \eta\,(y-a)\mathbf{x}$

: Vector Form

Or component form:

$w_0 \leftarrow w_0 + \eta\,(y-a)x_0$

$w_1 \leftarrow w_1 + \eta\,(y-a)x_1$

$w_2 \leftarrow w_2 + \eta\,(y-a)x_2$

Summary

- Logistic regression is a statistical model that uses a logistic function to model binary output labels.
- Logistic regression
  - The Gradient Descent Algorithm

$\Rightarrow \mathbf{w} \leftarrow \mathbf{w} + \eta\,(y - a)\mathbf{x}$: Vector Form

- $(\mathbf{x}, y)$: training example with binary output y
- a : Neuron output
- η : Learning rate

References

[1] Networks are like onions: Practical Deep Learning with TensorFlow, Barbara Fusinska, London Tensorflow Meetup, Jun 21, 2017.

[2] Neural Network Design (2nd Edition), Martin T Hagan , Howard B Demuth , Mark H Beale , and Orlando De Jesús, Martin Hagan, 2014.

[3] Pattern Recognition and Machine Learning, Christopher Bishop, Springer-Verlag New York, 2006.

[4] Machine Learning, Tom M. Mitchell, McGraw-Hill, 1997.