機器學習基礎簡介

Mark Chang

什麼是學習？

- 從已知的資料，根據資料的特徵，歸納出其中的規則（知識）。
- 用這個規則（知識）可預測未知的資料。

什麼是機器學習？

- 讓機器（電腦）根據已知資料的特徵，歸納出其中的規則。
- 用這個規則可預測未知的資料。
- 機器學習可分為兩大類：
  - 監督式學習：有老師，告訴答案
  - 非監督式學習：沒老師，自己學

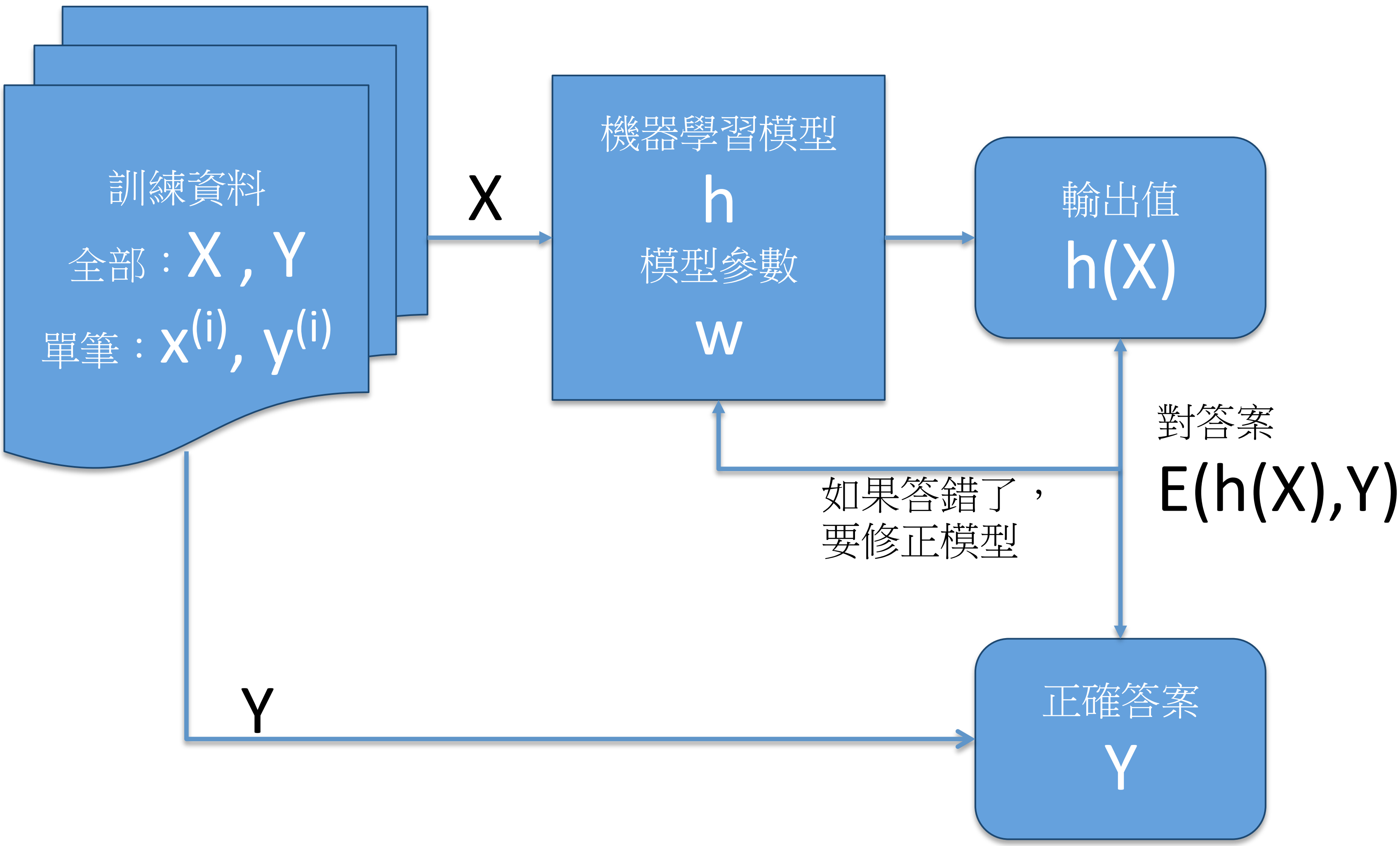
監督式學習種類

- 分類（Classification）：答案為一個類別
- 迴歸（Regression）：答案為一個數值

機器學習模型

- 迴歸：
  - 線性回歸（Linear Regression）
- 分類：
  - 邏輯迴歸（Logistic Regression）

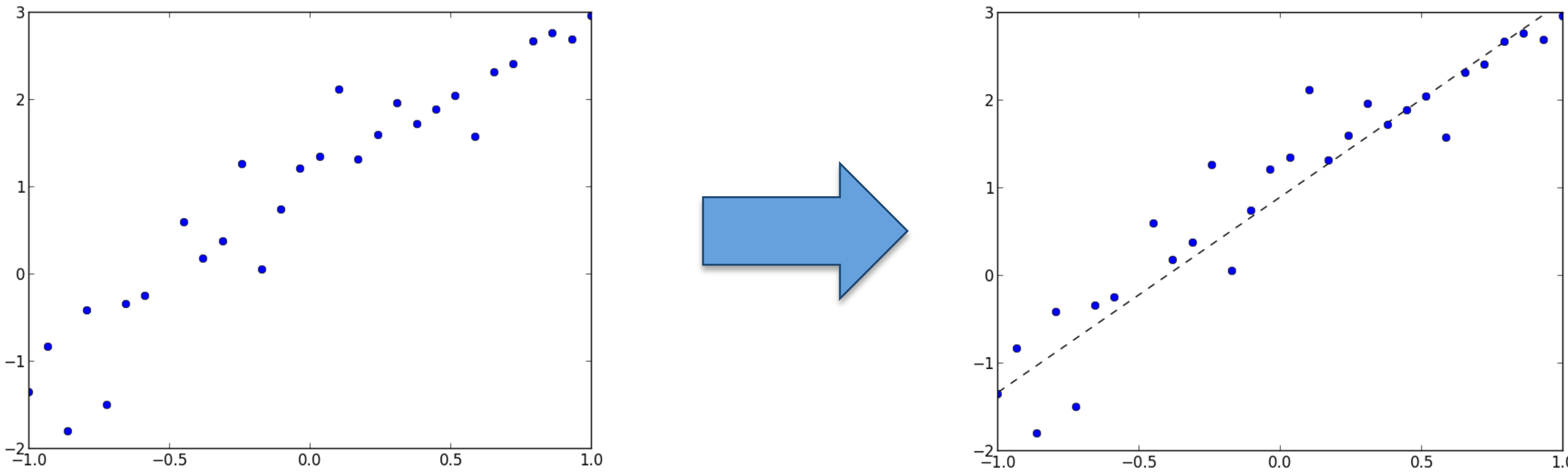
符號慣例



訓練資料
全部：X , Y
單筆：$x^{(i)}$, $y^{(i)}$
X
機器學習模型
h
模型參數
w
輸出值
h(X)
對答案
E(h(X),Y)
如果答錯了，
要修正模型
Y
正確答案
Y

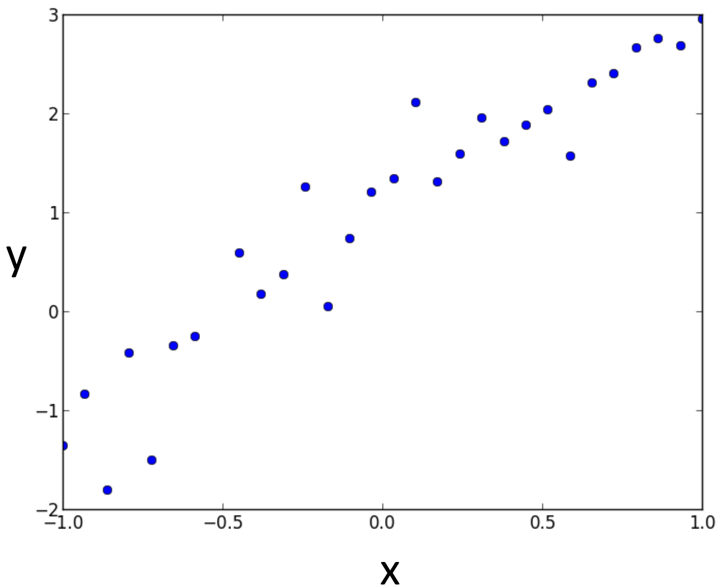
線性回歸（Linear Regression）

- 用直線去逼近資料分佈情形



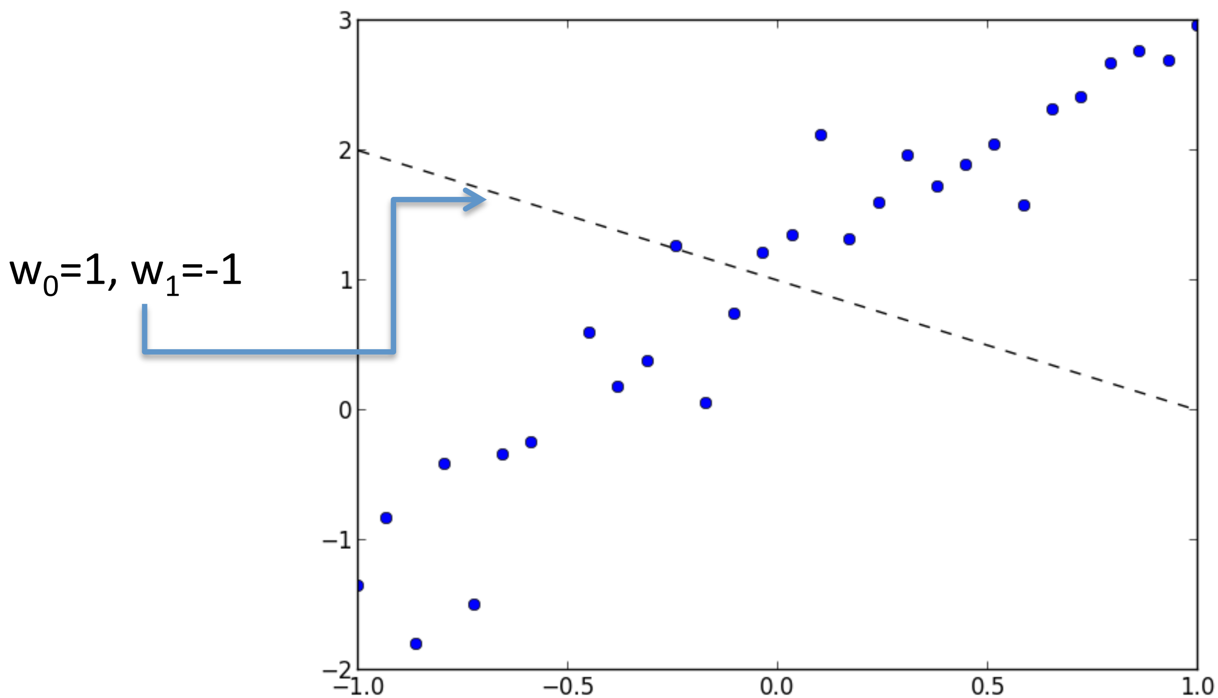
訓練資料

X	Y
-1	-1.35083488
-0.93103448	-0.832318
-0.86206897	-1.80594819
-0.79310345	-0.41854779
-0.72413793	-1.49916917
-0.65517241	-0.33899642
-0.5862069	-0.25423128
-0.51724138	-2.11349879
-0.44827586	0.59621997
-0.37931034	0.18047539
-0.31034483	0.37495828
…	…



機器學習模型

$$h(x) = w_0 + w_1 x$$



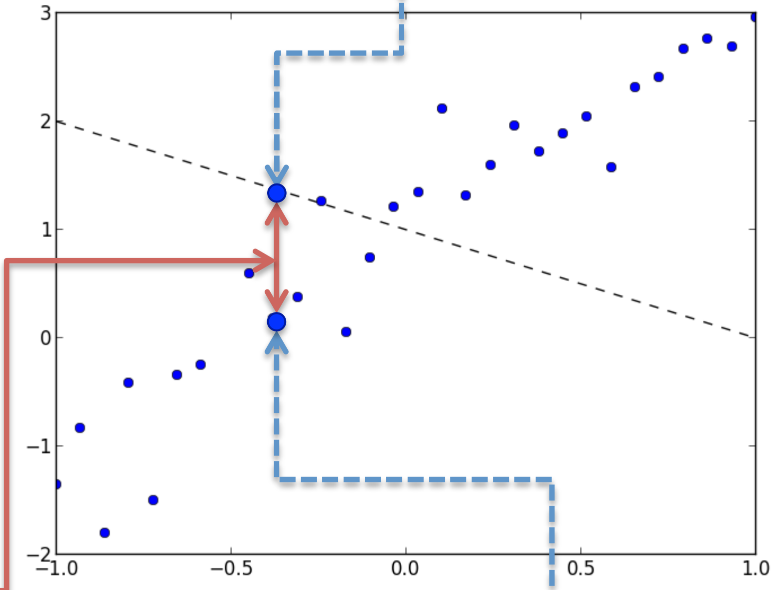
修正模型

- Error function:

$$E(h(X), Y)$$
$$= \frac{1}{2m} \sum_{i=1}^{m} (h(x^{(i)}) - y^{(i)})^2$$
$$= \frac{1}{2m} \sum_{i=1}^{m} ((w_0 + w_1 x^{(i)}) - y^{(i)})^2$$

$$(h(x^{(i)}) - y^{(i)})^2$$

$h(x^{(i)})$

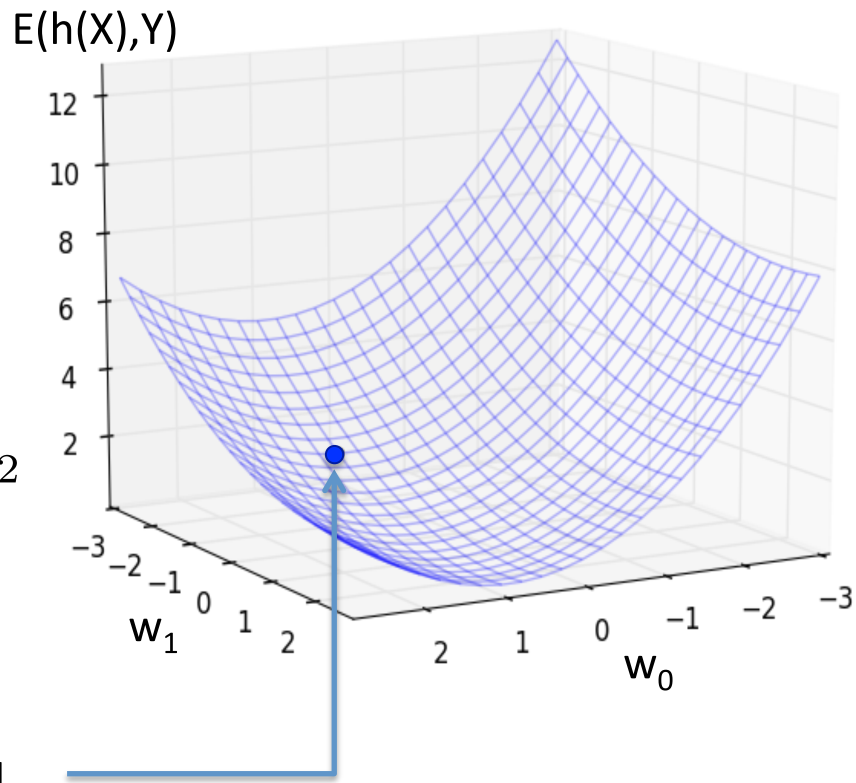


$y^{(i)}$

修正模型

- Error function:

$$
\begin{aligned}
E(h(X), Y) \\
&= \frac{1}{2m}\sum_{i=1}^{m}(h(x^{(i)}) - y^{(i)})^2 \\
&= \frac{1}{2m}\sum_{i=1}^{m}((w_0 + w_1 x^{(i)}) - y^{(i)})^2
\end{aligned}
$$



w_0=1, w_1=-1

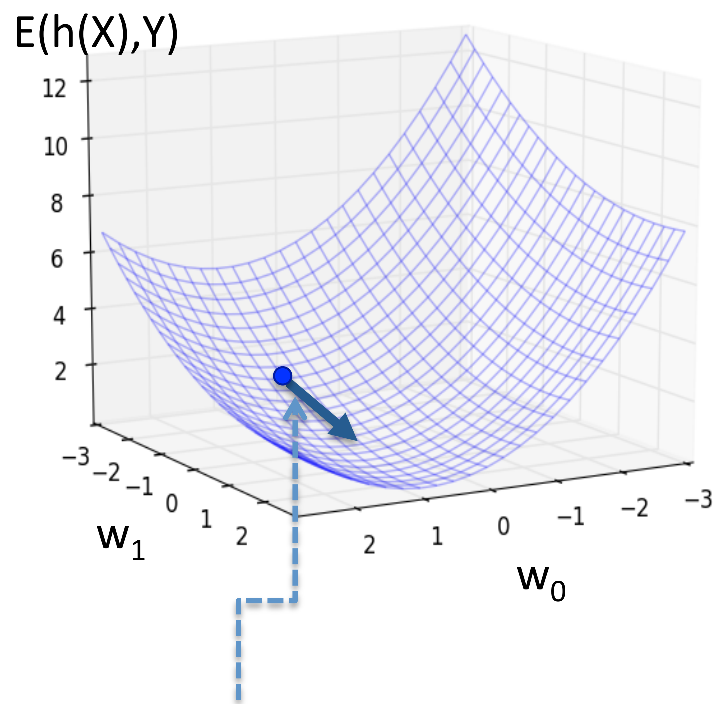
修正模型

- 梯度下降(Gradient Descent):

$$w_0 \leftarrow w_0 - \eta \frac{\partial E(h(X), Y)}{\partial w_0}$$

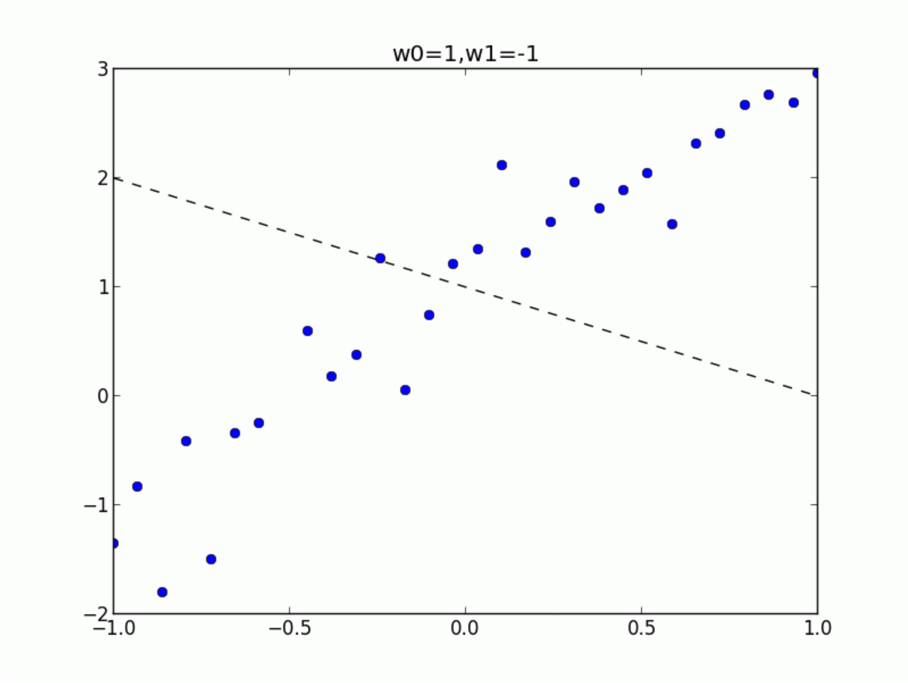
$$w_1 \leftarrow w_1 - \eta \frac{\partial E(h(X), Y)}{\partial w_1}$$

η : Learning Rate

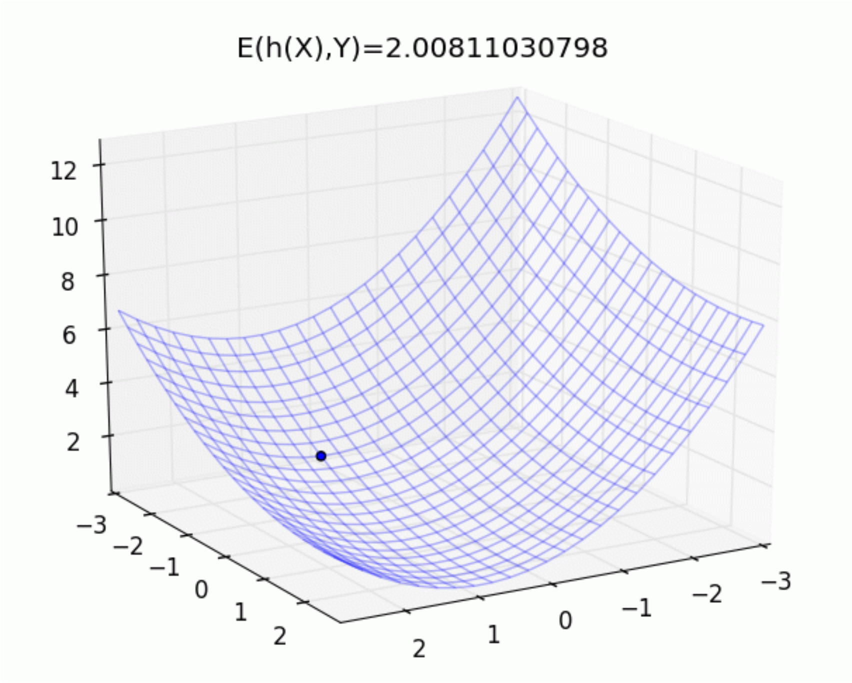


$$\left(-\frac{\partial E(h(X), Y)}{\partial w_0}, -\frac{\partial E(h(X), Y)}{\partial w_1}\right)$$

修正模型



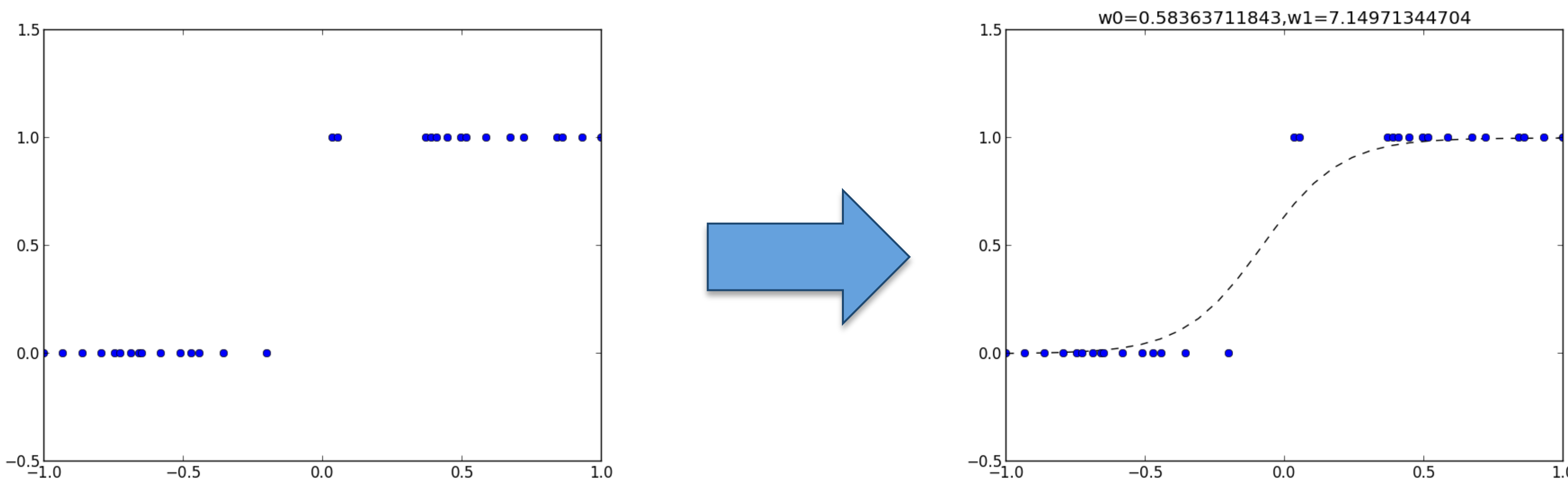
w0=1,w1=-1
3
2
1
0
−1
−2
−1.0
−0.5
0.0
0.5
1.0



E(h(X),Y)=2.00811030798
12
10
8
6
4
2
−3
−2
−1
0
1
2

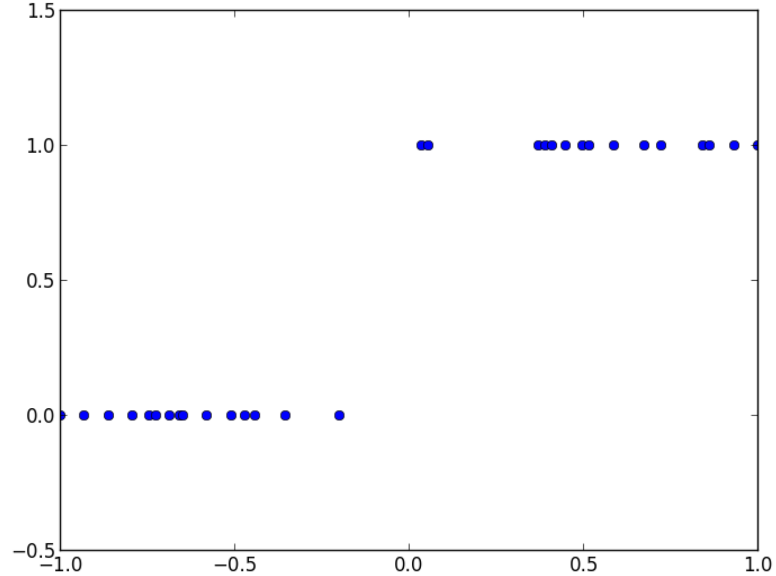
邏輯回歸（Logistic Regression）

- 用Sigmoid曲線去逼近資料的分佈情形



訓練資料

X	Y
-0.47241379	0
-0.35344828	0
-0.30148276	0
0.33448276	1
0.35344828	1
0.37241379	1
0.39137931	1
0.41034483	1
0.44931034	1
0.49827586	1
0.51724138	1
....	....



機器學習模型

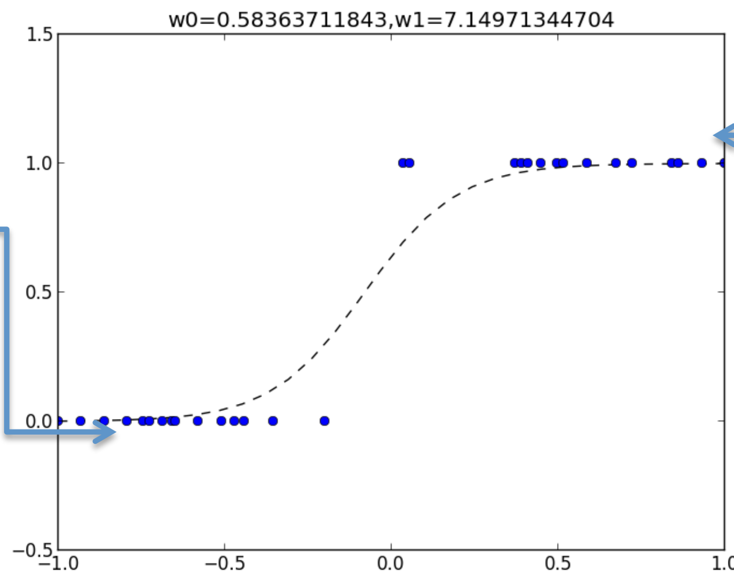
Sigmoid function $h(x) = \dfrac{1}{1 + e^{-(w_0 + w_1 x)}}$

$w_0 + w_1 x > 0$

$h(x) \approx 1$

$w_0 + w_1 x < 0$

$h(x) \approx 0$



修正模型

- Error function : Cross Entropy

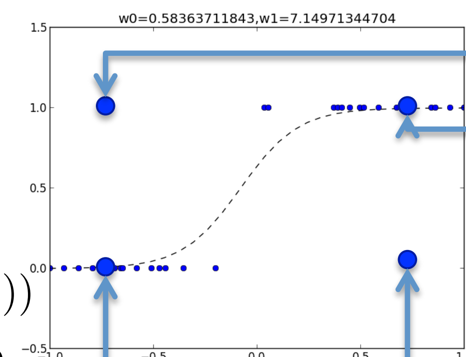
$$E(h(X),Y) = \frac{-1}{m}(\sum_{i}^{m} y^{(i)} log(h(x^{(i)})) + (1-y^{(i)}) log(1-h(x^{(i)})))$$

$$y^{(i)} = 1$$

$$E(h(x^{(i)}), y^{(i)}) = -log(h(x^{(i)}))$$

$$h(x^{(i)}) \approx 0 \Rightarrow E(h(x^{(i)}), y^{(i)}) \approx \infty$$

$$h(x^{(i)}) \approx 1 \Rightarrow E(h(x^{(i)}), y^{(i)}) \approx 0$$



$$y^{(i)} = 0$$

$$E(h(x^{(i)}), y^{(i)}) = -log(1-h(x^{(i)}))$$

$$h(x^{(i)}) \approx 0 \Rightarrow E(h(x^{(i)}), y^{(i)}) \approx 0$$

$$h(x^{(i)}) \approx 1 \Rightarrow E(h(x^{(i)}), y^{(i)}) \approx \infty$$

修正模型

- Error function : Cross Entropy

$$E(h(X),Y) = \frac{-1}{m}(\sum_{i}^{m} y^{(i)} log(h(x^{(i)})) + (1-y^{(i)}) log(1-h(x^{(i)})))$$

$$h(x^{(i)}) \approx 0 \text{ and } y^{(i)} = 0 \Rightarrow E(h(X),Y) \approx 0$$

$$h(x^{(i)}) \approx 1 \text{ and } y^{(i)} = 1 \Rightarrow E(h(X),Y) \approx 0$$

$$h(x^{(i)}) \approx 0 \text{ and } y^{(i)} = 1 \Rightarrow E(h(X),Y) \approx \infty$$

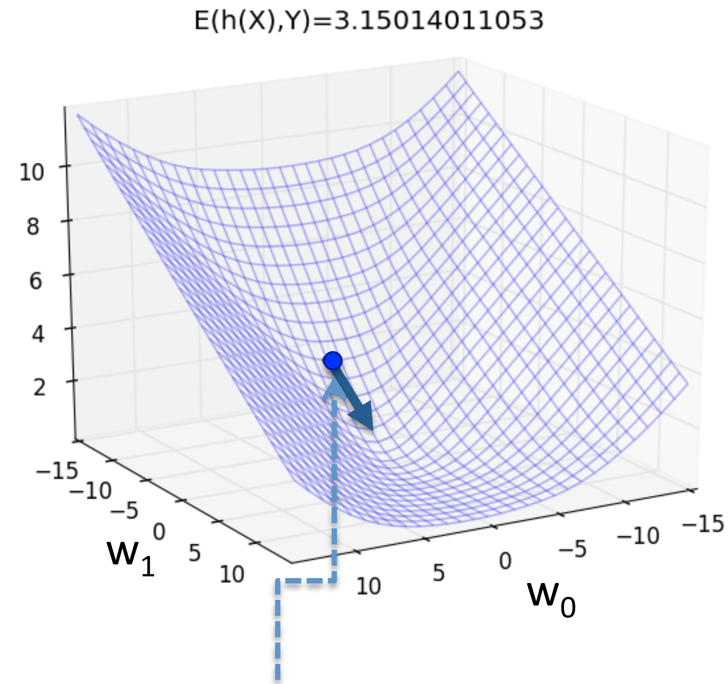
$$h(x^{(i)}) \approx 1 \text{ and } y^{(i)} = 0 \Rightarrow E(h(X),Y) \approx \infty$$

修正模型

- 梯度下降:

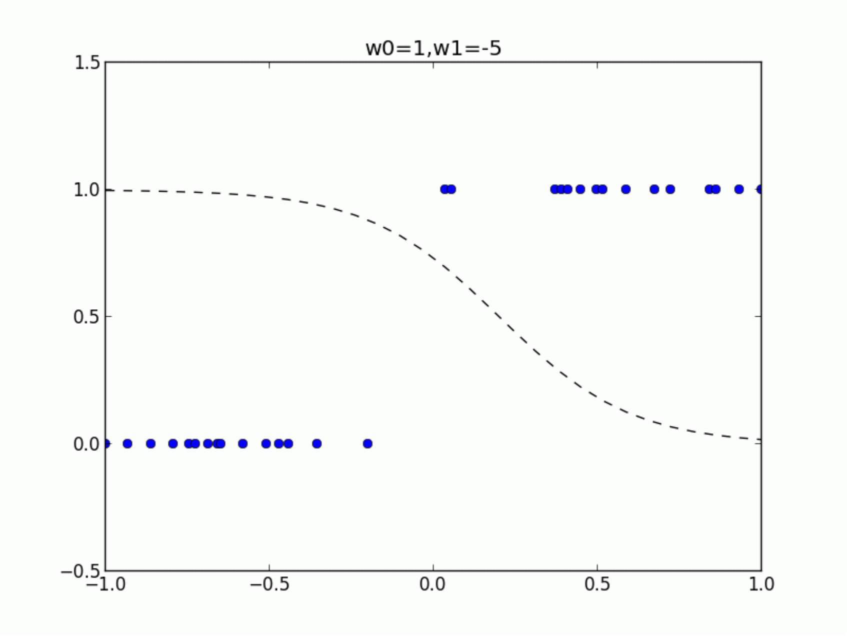
$$w_0 \leftarrow w_0 - \eta \frac{\partial E(h(X),Y)}{\partial w_0}$$

$$w_1 \leftarrow w_1 - \eta \frac{\partial E(h(X),Y)}{\partial w_1}$$

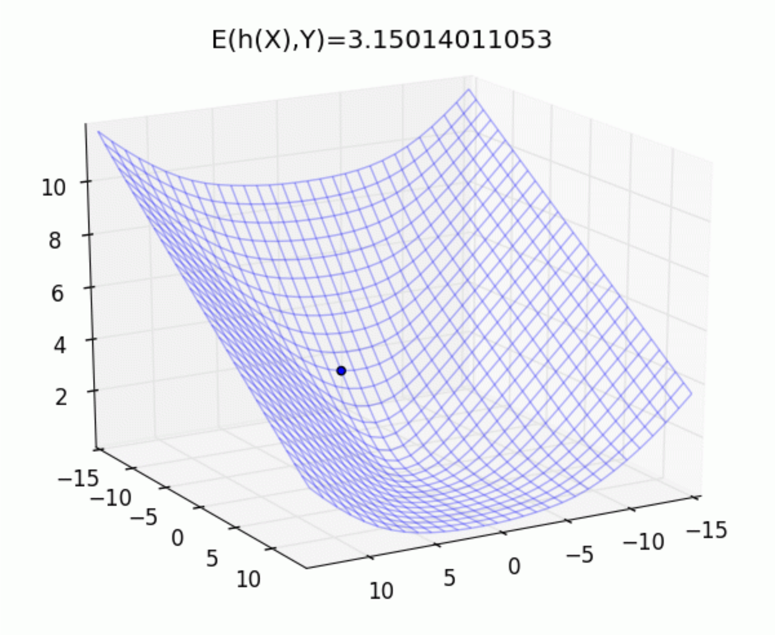


$$\left(-\frac{\partial E(h(X),Y)}{\partial w_0}, -\frac{\partial E(h(X),Y)}{\partial w_1}\right)$$

修正模型



w0=1,w1=-5
1.5
1.0
0.5
0.0
−0.5
−1.0
−0.5
0.0
0.5
1.0



E(h(X),Y)=3.15014011053
10
8
6
4
2
−15
−10
−5
0
5
10
10
5
0
−5
−10
−15

實作：邏輯回歸

延伸閱讀

- Logistic Regression 3D
  - http://cpmarkchang.logdown.com/posts/189069-logisti-regression-model
- OverFitting and Regularization
  - http://cpmarkchang.logdown.com/posts/193261-machine-learning-overfitting-and-regularization
- Model Selection
  - http://cpmarkchang.logdown.com/posts/193914-machine-learning-model-selection
- Neural Network Back Propagation
  - http://cpmarkchang.logdown.com/posts/277349-neural-network-backward-propagation