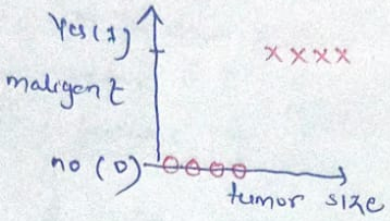
Week_1 Machine learning

Classification

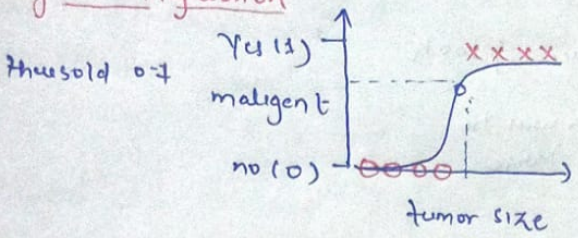
Question	Answer
Is this email spam	"Yes" "No"
Is this transaction fraud	"Y" "N"

"Binary classification"
Two classes or categories

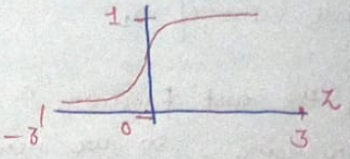
1 → positive class
0 → negative class



Logistic Regression



Sigmoid Function
or
Logistic Function



$$f_{\vec{w},b}(\vec{x})$$

$$z = \vec{w}\cdot\vec{x} + b$$

$$z$$

$$g(z) = \frac{1}{1+e^{-z}}$$

$$0 < g(z) < 1$$

$$f_{\vec{w},b}(\vec{x}) = g(\underbrace{\vec{w}\cdot\vec{x} + b}_{z}) = \frac{1}{1+e^{-(\vec{w}\cdot\vec{x}+b)}}$$

"logistic regression"

Interpretation of logistic regression output

$$f_{\vec{w},b}(\vec{x}) = \frac{1}{1+e^{-(\vec{w}\cdot\vec{x}+b)}}$$

"probability that class is 1"

$f_{\vec{w},b}(\vec{x}) = 0.7$ 70% chances that y is 1

when is $f_{\vec{w},b}(\vec{x}) \geq 0.5$?

$g(z) \geq 0.5$

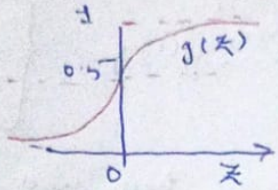
$z \geq 0$

$\vec{w}\cdot\vec{x} + b \geq 0$

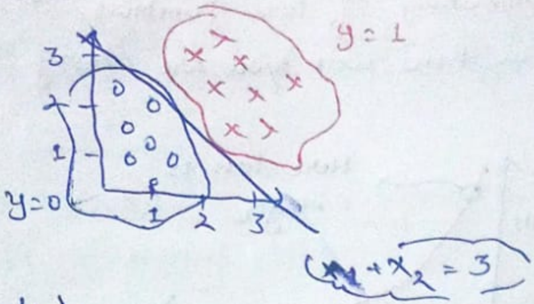
$\hat{y} = 1$

$\vec{w}\cdot\vec{x} + b < 0$

$\hat{y} = 0$



Decision Boundary



$f_{\vec{w},b}(\vec{x}) = g(z) =$

$g(w_1x_1 + w_2x_2 + b)$

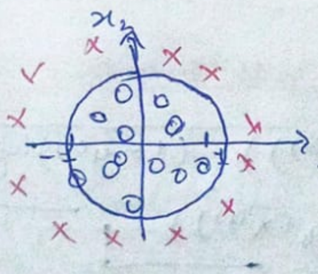
$w_1 = 1 \quad w_2 = 1 \quad b = -3$

Decision boundary $z = \vec{w}\cdot\vec{x} + b = 0$

$z = x_1 + x_2 - 3 = 0$

$x_1 + x_2 = 3$

Non-linear decision Boundary



$f_{\vec{w},b}(\vec{x}) = g(z) = g(w_1x_1^2 + w_2x_2^2 + b)$

$w_1 = 1 \quad w_2 = 1 \quad b = -1$

decision boundary $z = x_1^2 + x_2^2 - 1 = 0$

$x_1^2 + x_2^2 = 1$

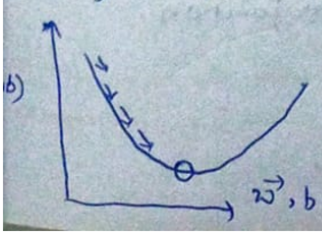
Cost function for Logistic Regression

Squared error cost

$$J(\vec{w},b) = \frac{1}{m}\sum_{i=1}^{m}\frac{1}{2}\left(f_{\vec{w},b}(\vec{x}^{(i)}) - y^{(i)}\right)^2$$

} Not good choise

Linear Regression

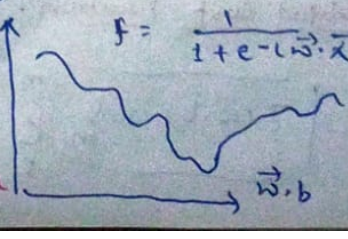


Logistic Regression

$f = \frac{1}{1+e^{-(\vec{w}\cdot\vec{x}+b)}}$

$J(\vec{w},b)$

So many local minima

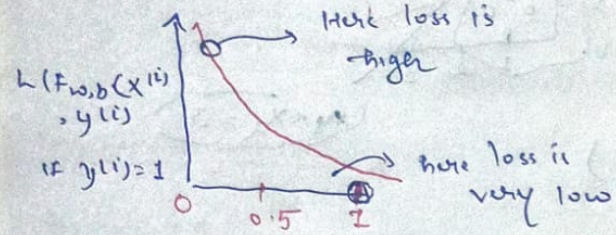


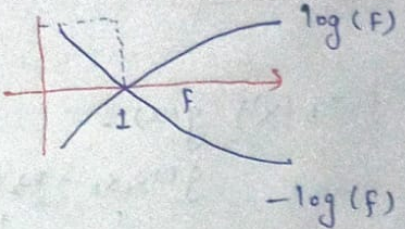
Logistic loss function

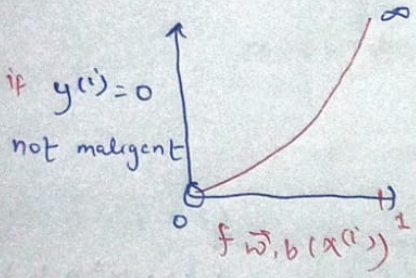
$$L(f_{\vec{w},b}(\vec{x}^{(i)}), y^{(i)}) = \begin{cases} -\log(f_{\vec{w},b}(\vec{x}^{(i)})) & \text{if } y^{(i)} = 1 \\ -\log(1 - f_{\vec{w},b}(\vec{x}^{(i)})) & \text{if } y^{(i)} = 0 \end{cases}$$

Loss function tells us how well you are doing in one training sample

After summation of loss function we get cost function which tells us how well you are doing in whole training set.







The further prediction $f_{\vec{w},b}(x^{(i)})$ is from target $y^{(i)}$ the higher the loss.

Simplified loss function.

$$L(f_{\vec{w},b}(\vec{x}^{(i)}), y^{(i)}) = -y^{(i)} \log(f_{\vec{w},b}(\vec{x}^{(i)})) - (1 - y^{(i)}) \log(1 - f_{\vec{w},b}(\vec{x}^{(i)}))$$

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

$$L(f_{\vec{w},b}(\vec{x}^{(i)}), y^{(i)}) = -1 \log(f(\vec{x}))$$

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

$$L(f_{\vec{w},b}(\vec{x}^{(i)}), y^{(i)}) = -\log(1 - f(\vec{x}))$$

Simplified cost function.

upper function.

$$J(\vec{w},b) = \frac{1}{m} \sum_{i=1}^{m} L(f_{\vec{w},b}(\vec{x}^{(i)}), y^{(i)})$$

Training logistic regression

find $\vec{w}, b$ given new $\vec{x}$, output $f_{\vec{w},b}(\vec{x}) = \frac{1}{1+e^{-(\vec{w}\cdot\vec{x}+b)}}$

$$P(y=1|\vec{x};\vec{w},b)$$

Minimizing the cost function {gradient descent algorithm}

$$J(\vec{w},b) = -\frac{1}{m}\sum_{i=1}^{m}\left[y^{(i)}\log\left(f_{\vec{w},b}(\vec{x}^{(i)})\right) + (1-y^{(i)})\log\left(1-f_{\vec{w},b}(\vec{x}^{(i)})\right)\right]$$

repeat {

$$w_j = w_j - \alpha\frac{d}{dw_j}J(\vec{w},b) \qquad \frac{d}{dw_j}J(\vec{w},b) = \frac{1}{m}\sum_{i=1}^{m}(f_{\vec{w},b}(\vec{x}^{(i)}) - y^{(i)})x_j^{(i)}$$

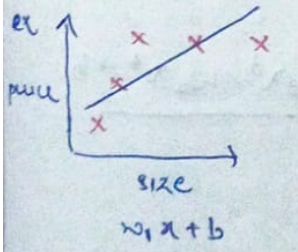
$$b = b - \alpha\frac{d}{db}J(\vec{w},b) \qquad \frac{d}{dw}J(\vec{w},b) = \frac{1}{m}\sum_{i=1}^{m}(f_{\vec{w},b}(\vec{x}^{(i)}) - y^{(i)})$$

This all looks like same as linear regression but here definition of $f(x)$ is different

Linear regression = $f_{\vec{w},b}(\vec{x}) = \vec{w}\cdot\vec{x} + b$

Logistic regression = $f_{\vec{w},b}(\vec{x}) = \frac{1}{1+e^{-(\vec{w}\cdot\vec{x}+b)}}$

The Problem of Overfitting

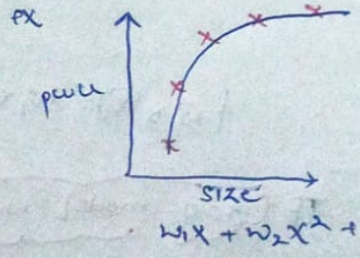


$w_1x + b$

Underfit

Does not fit the training set well

high Bias

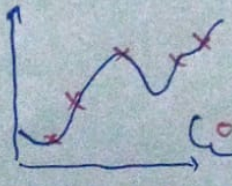


Just right

$w_1x + w_2x^2 + b$

- fits training set pretty well

generalization: predict well in new test set

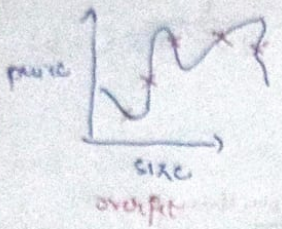


Fits the training set extremely well

overfitting ↔ high variance

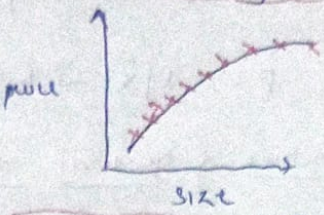
$w_1x + w_2x^2 + w_3x^3 + w_4x^4 + b$

• Addressing Overfitting



overfit

→ Collecting more training examples ★



* Select features to include/exclude) → feature selection

Reduce features if you have less data

★

* Regularization: Reduce the size of parameters w_j eliminating features by setting value 0 {or small}

for ex $w_3 \approx 0$

$w_4 \approx 0$

$$\min_{\vec{w},b} J(\vec{w},b) = \min_{\vec{w},b} \left[\frac{1}{2m} \sum_{i=1}^{m} (f_{\vec{w},b}(\vec{x}^{(i)}) - y^{(i)})^2 + \frac{\lambda}{2m} \sum_{j=1}^{n} w_j^2 \right]$$

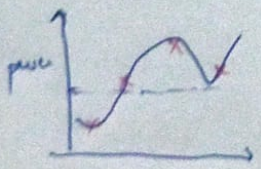
mean squared error

regularization term

fit data ↘ ↙ keep w_j small

↓ balance both goals.

$$f_{\vec{w},b}(\vec{x}) = w_1 x + w_2 x^2 + w_3 x^3 + w_4 x^4 + b$$



If $\lambda = 0$ model will overfit

If $\lambda = 10^{10}$ model will underfit.

$w_1 \approx w_2 \approx w_3 \approx w_4 = 0$