Day-26, Oct-28, 2024.

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

So, we want to find parameters w and b that minimizes the regularized cost function.

Cost Function of Linear Regression:

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

So finding min $\vec{w}, b$

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

We have Gradient descent Repeat {

$$w_j = w_j - \alpha \frac{\partial}{\partial w_j} J(\vec{w},b) \quad \Big| \quad b = b - \alpha \frac{\partial}{\partial b} J(\vec{w},b)$$

Simultaneous updates }

So, $\frac{\partial}{\partial w_j} J(\vec{w}, b)$ is given by the equation $\left(\frac{d}{dw_j}\right)$

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

and $\frac{\partial}{\partial b} J(\vec{w}, b)$ is given by the equation

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

Now, for Regularized Regression we add $\frac{\lambda}{m} w_j$

Since $\frac{\partial}{\partial w_j} J(\vec{w}, b)$ is obtained w.r.t $\frac{\partial}{\partial w_j}$ to Cost function

So,

$$\frac{\partial}{\partial w_j} J(\vec{w}, b) \Rightarrow \frac{1}{m} \sum_{i=1}^{m} \left(f_{\vec{w}, b}(\vec{x}^{(i)}) - y^{(i)} \right) x_j^{(i)} + \frac{\lambda}{m} w_j$$

Whereas b is not Regularized

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

Regularized linear regression

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

Gradient descent

repeat {

$$w_j = w_j - \alpha \frac{\partial}{\partial w_j} J(\vec{w},b) \qquad = \frac{1}{m}\sum_{i=1}^{m}\left(f_{\vec{w},b}(\vec{x}^{(i)}) - y^{(i)}\right)x_j^{(i)} + \frac{\lambda}{m} w_j$$

$j = 1, \ldots, n$

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

} simultaneous update

don't have to regularize b

Stanford ONLINE DeepLearning.AI Andrew Ng

Regularized Gradient Descent

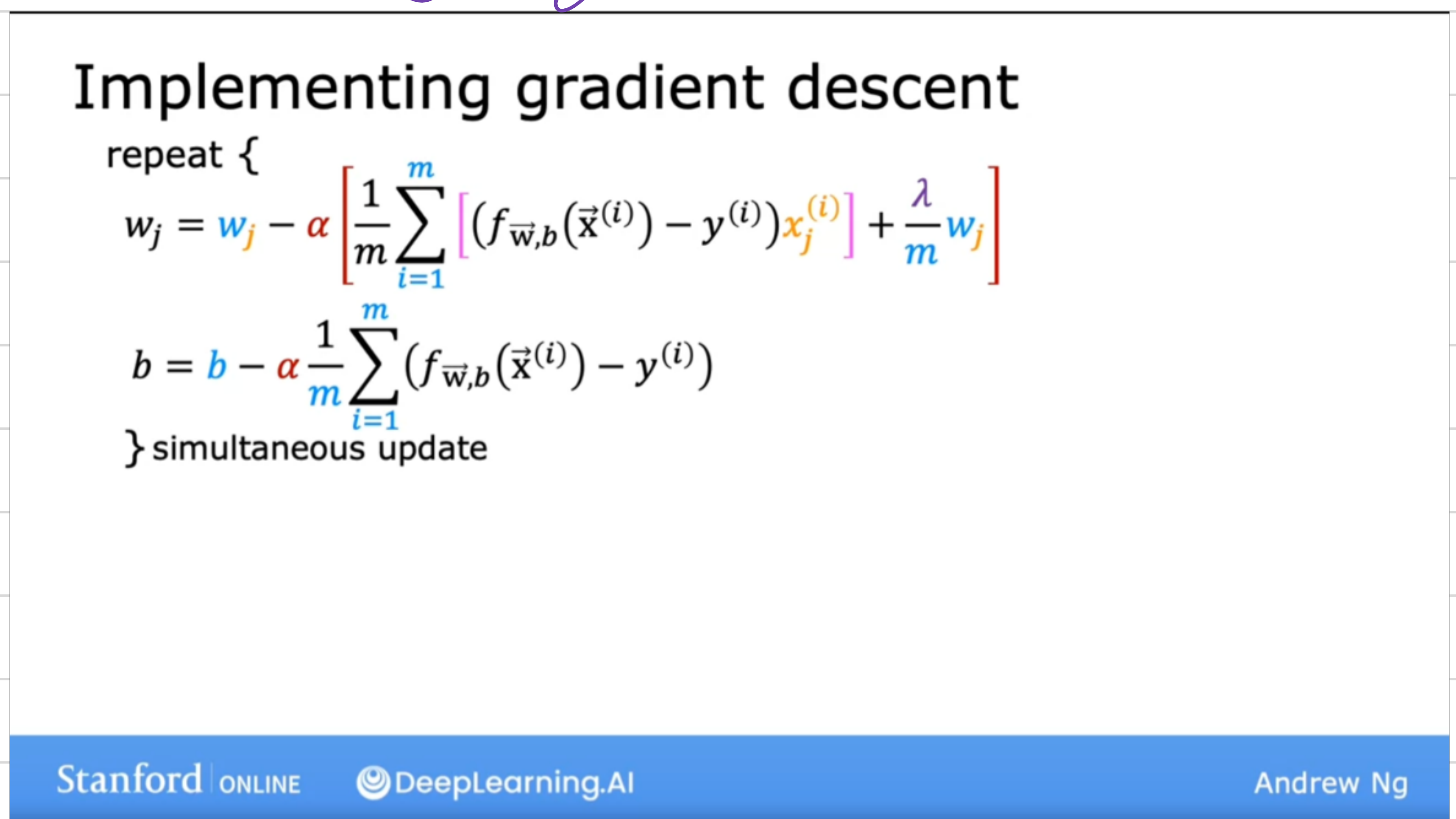
$$\frac{\partial}{\partial w_j} J(\vec{w},b) \Rightarrow \frac{1}{m}\sum_{i=1}^{m}\left(f_{\vec{w},b}(\vec{x}^{(i)}) - y^{(i)}\right)x_j^{(i)} + \lambda/m \, w_j$$

Implementing them in linear Regularized Regression

$$\{ \; w_j = w_j - \alpha \left[\frac{1}{m} \sum_{i=1}^{m} \left[\left(f_{\vec{w},b}(\vec{x}^{(i)}) - y^{(i)} \right) x_j^{(i)} \right] + \frac{\lambda}{m} w_j \right]$$

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

We expect them to reduce the overfitting problem.



See How α, λ, m Plays role in penalizing.

Implementing gradient descent

repeat {

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

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

} simultaneous update $j = 1 \ldots n$

$$w_j = \underbrace{1 w_j - \alpha \frac{\lambda}{m} w_j}_{w_j \underbrace{\left(1 - \alpha \frac{\lambda}{m}\right)}_{\text{shrink } w_j}} \underbrace{- \alpha \frac{1}{m} \sum_{i=1}^{m} (f_{w,b}(\vec{x}^{(i)}) - y^{(i)}) x_j^{(i)}}_{\text{usual update}}$$

$$\alpha \frac{\lambda}{m}$$

$$0.01 \frac{1}{50} = 0.0002$$

$$w_j \underbrace{(1 - 0.0002)}_{0.9998}$$

$\alpha \Rightarrow$ Learning Rate and λ is the Regularization value where m is the Sample repeat{

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

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

} Simultaneous updates.

$$w_j = \underbrace{1 w_j - \alpha \frac{\lambda}{m} w_j}_{\text{Regularized Added term}} \underbrace{- \alpha \frac{1}{m} \sum_{i=1}^{m} \left(f_{\vec{w},b}\left(\vec{x}^{(i)}\right) - y^{(i)} \right) \cdot x_j^{(i)}}_{\text{usual update for UnRegularized}}$$

So, $w_j = 1 w_j - \alpha \frac{\lambda}{m} w_j$

Where, $\alpha \Rightarrow 0.01$ or $\alpha = 0.1$.

$\lambda \Rightarrow 1$

$m = 50$.

$(1 w_j - 0.0002)$ i.e. $(1 - 0.0002)$ $j = 1$

$\Rightarrow 0.9998$

So, Computation value decreases - Regularization is the effect of shrinking w_j.

How we get the derivative term (optional)

$$\frac{\partial}{\partial w_j} J(\vec{w},b) = \frac{d}{dw_j}\left[\frac{1}{2m}\sum_{i=1}^{m}\left(\underbrace{f(\vec{x}^{(i)})}_{\vec{w}\cdot\vec{x}^{(i)}+b} - y^{(i)}\right)^2 + \frac{\lambda}{2m}\sum_{j=1}^{n} w_j^2\right]$$

$$= \frac{1}{\cancel{2}m}\sum_{i=1}^{m}\left[\left(\vec{w}\cdot\vec{x}^{(i)} + b - y^{(i)}\right)\cancel{2}\,x_j^{(i)}\right] + \frac{\lambda}{\cancel{2}m}\cancel{2}\,w_j$$

No $\sum_{j=1}^{n}$

$$= \frac{1}{m}\sum_{i=1}^{m}\left[\left(\underbrace{\vec{w}\cdot\vec{x}^{(i)} + b}_{f(\vec{x})} - y^{(i)}\right)x_j^{(i)}\right] + \frac{\lambda}{m}w_j$$

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

Stanford ONLINE DeepLearning.AI Andrew Ng

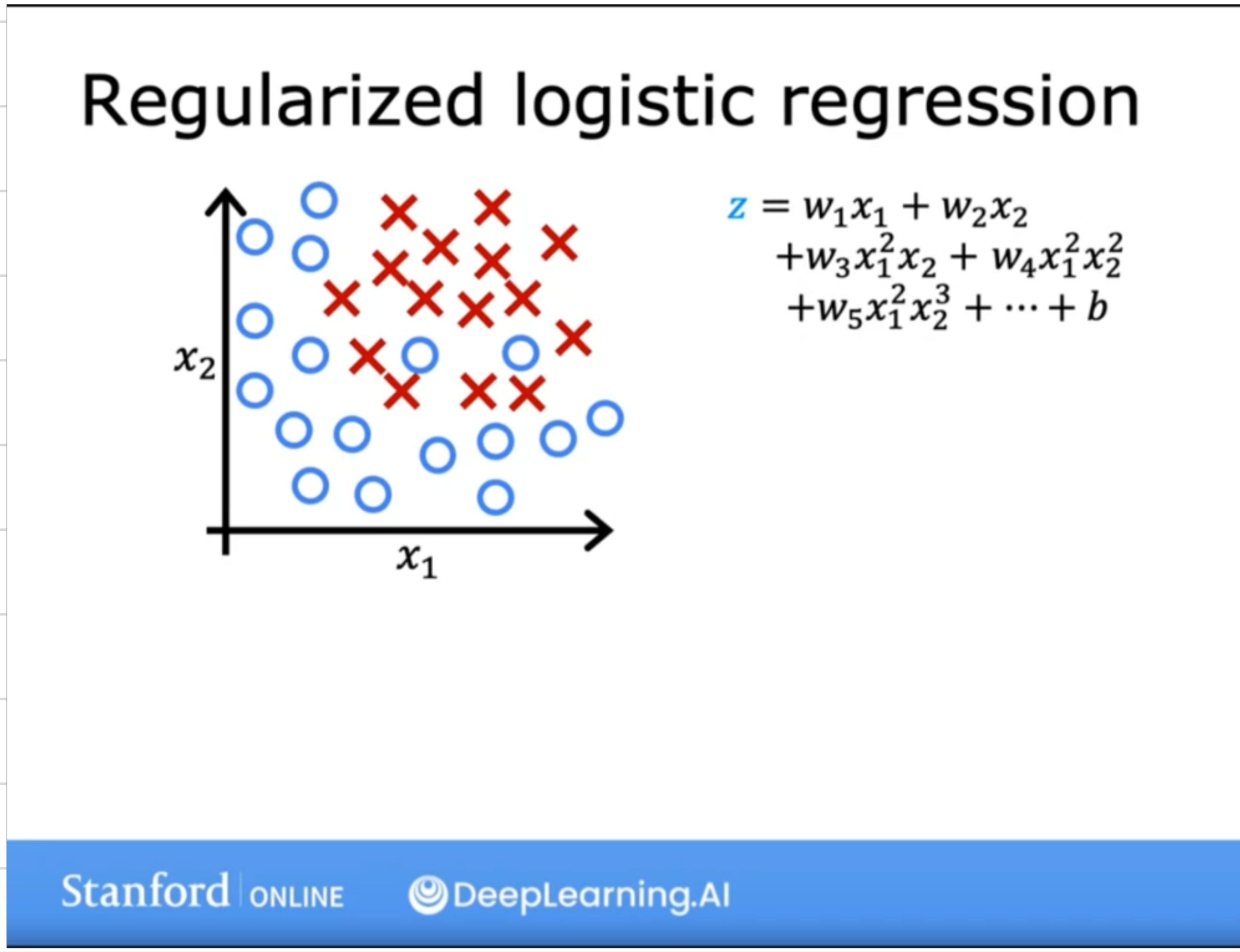
Derivation part → How we got $\frac{\partial}{\partial w_j} J(\vec{w},b)$.

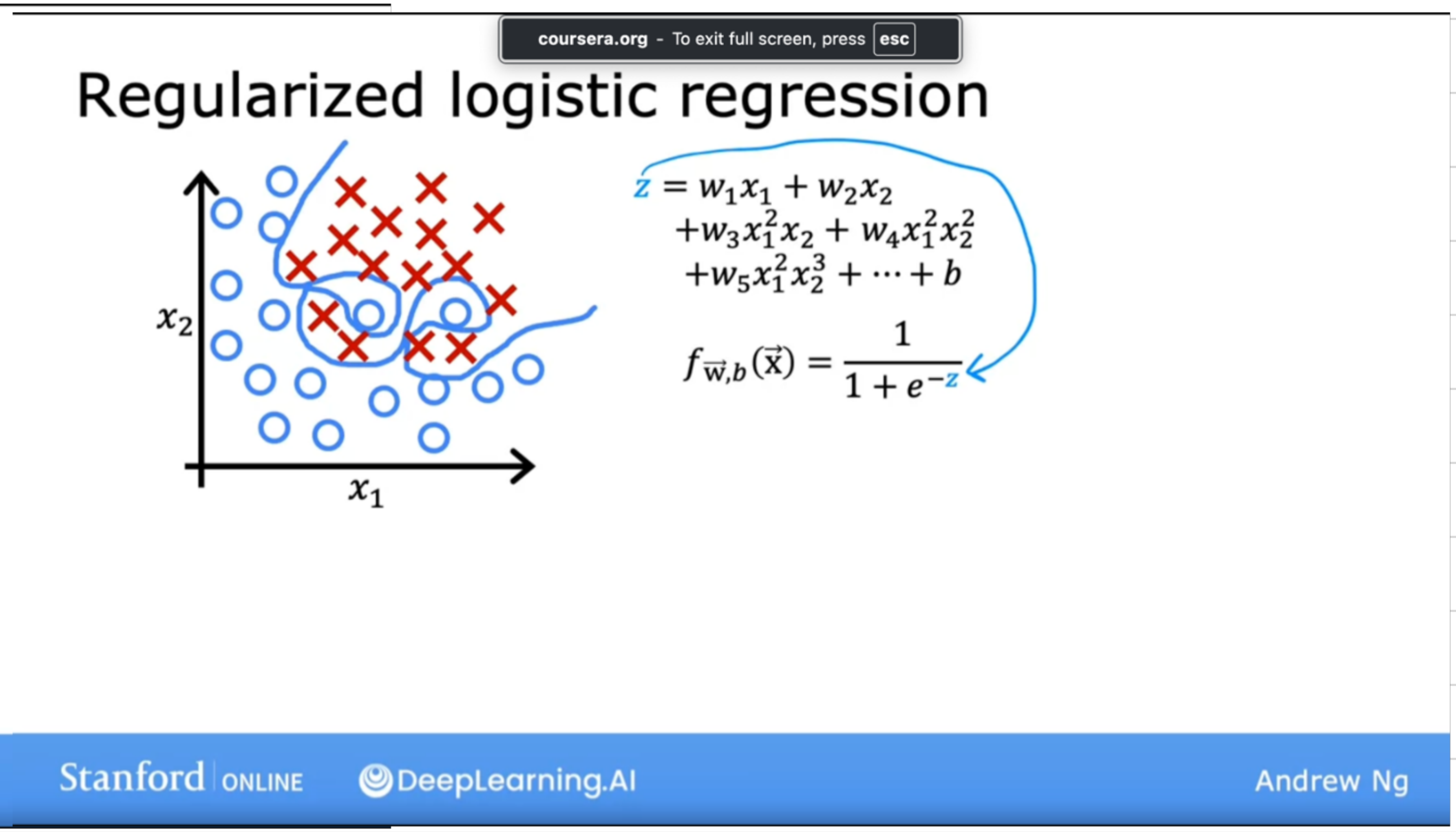
$$\frac{\partial}{\partial w_j} J(\vec{w}, b) \Rightarrow \frac{\partial}{\partial w_j}\left[\frac{1}{2m}\sum_{i=1}^{m}\left(f(\vec{x}^{(i)}) - y^{(i)}\right)^2 + \frac{\lambda}{2m}\sum_{j=1}^{n} w_j^2\right]$$

$$\Rightarrow \frac{1}{\cancel{2}m}\sum_{i=1}^{m}\left[\left(\vec{w}\cdot\vec{x}^{(i)} + b - y^{(i)}\right)\cancel{2}x_j^{(i)}\right] + \frac{\lambda}{\cancel{2}m}\cancel{2}w_j$$

$$\Rightarrow \frac{1}{m}\sum_{i=1}^{m}\left[\left(\underbrace{\vec{w}\cdot\vec{x}^{(i)} + b}_{f(\vec{x})} - y^{(i)}\right)x_j^{(i)}\right] + \frac{\lambda}{m}\cdot w_j$$

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





Higher Order Numerous Polynomial Features will likely to make logistic Regression Overfitting See how it is being overfitted.

Cost Function of Logistic Regression:

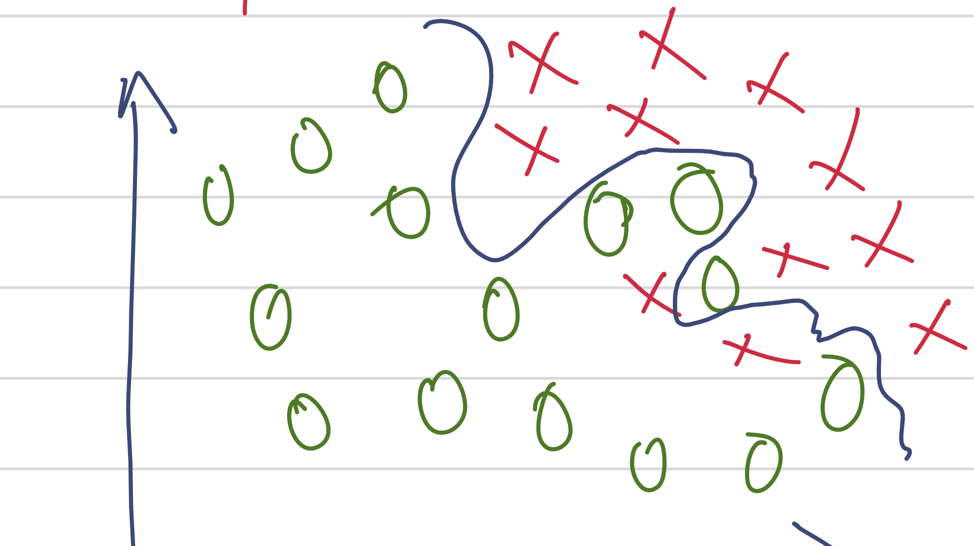
$$J(\vec{w}, b) = -\frac{1}{m}\sum_{i=1}^{m}\left[y^{(i)}\log\left(f_{\vec{w},b}\left(\vec{x}^{(i)}\right)\right) + \left(1-y^{(i)}\right)\log\left(1-f_{\vec{w},b}\left(\vec{x}^{(i)}\right)\right)\right]$$

Add term $\frac{\lambda}{2m}\sum_{j=1}^{n} w_j^2$ – Regularized terms λ = Regularized values

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

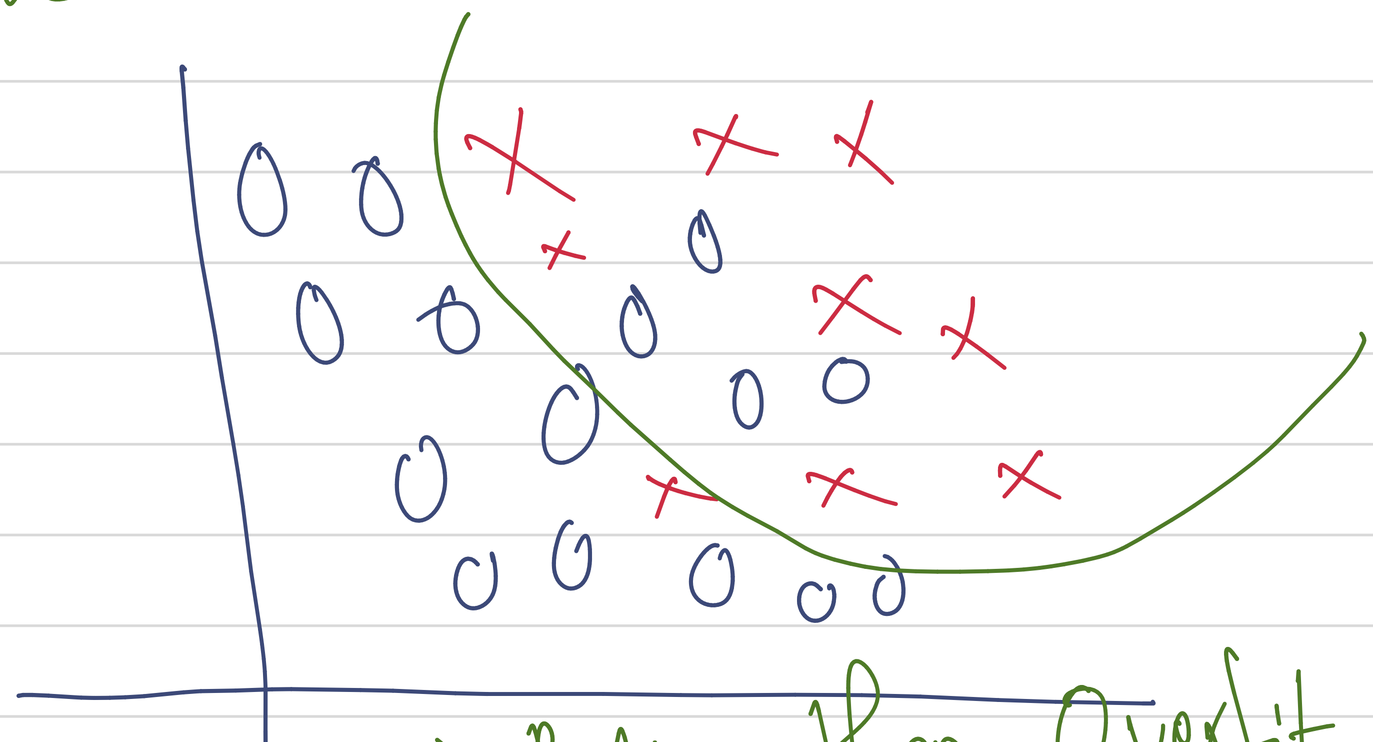
$$J(\vec{w},b) = \left[-\frac{1}{m}\sum_{i=1}^{m}\left[y^{(i)}\log\left(f_{\vec{w},b}(\vec{x}^{(i)})\right) + (1-y^{(i)})\log\left(1 - f_{\vec{w},b}(\vec{x}^{(i)})\right)\right]\right] + \frac{\lambda}{2m}\sum_{j=1}^{n} w_j^2$$

$$\min_{\vec{w},b} J(\vec{w},b) \longrightarrow w_j \downarrow$$



Overfit

$\longrightarrow$



$\rightarrow$ Better than Overfit –

$J(\vec{w}, b) \rightarrow$ Cost Function Minimize it.

How Logistic Regression $J(\vec{w}, b)$ is Regularized?

coursera.org - To exit full screen, press esc

Regularized logistic regression

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

Gradient descent

Looks same as for linear regression!

repeat {

$$w_j = w_j - \alpha \frac{\partial}{\partial w_j} J(\vec{w}, b) \qquad j=1\ldots n$$

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

$$b = b - \alpha \frac{\partial}{\partial b} J(\vec{w}, b)$$

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

}

logistic regression

don't have to regularize

Stanford ONLINE DeepLearning.AI Andrew Ng

Gradient Descent

repeat {

$$w_j = w_j - \alpha \boxed{\frac{\partial}{\partial w_j} J(\vec{w}, b)}$$

$j = 1, 2, 3 \ldots n$

$$b = b - \alpha \boxed{\frac{\partial}{\partial b} J(\vec{w}, b)}$$

}

Where, $\frac{\partial}{\partial w_j} J(\vec{w}, b)$ is $\frac{1}{m} \sum_{i=1}^{m} \left(f_{\vec{w}, b}(\vec{x}^{(i)}) - y^{(i)} \right) x_j^{(i)} + \frac{\lambda}{m} w_j$

& $\frac{\partial}{\partial b} J(\vec{w}, b) \Rightarrow \frac{1}{m} \sum_{i=1}^{m} \left(f_{\vec{w}, b}(\vec{x}^{(i)}) - y^{(i)} \right)$ → No Logic Regression Regularized since.