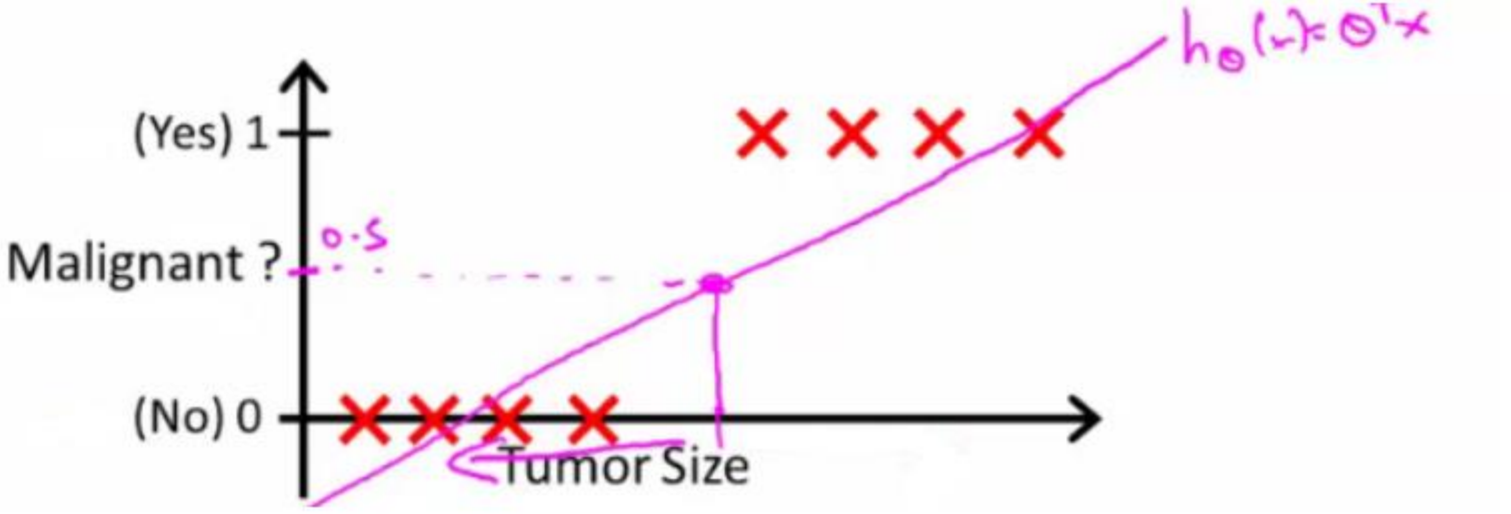
Advanced Topic In Deep Learning

Dr. Muhammad Safyan

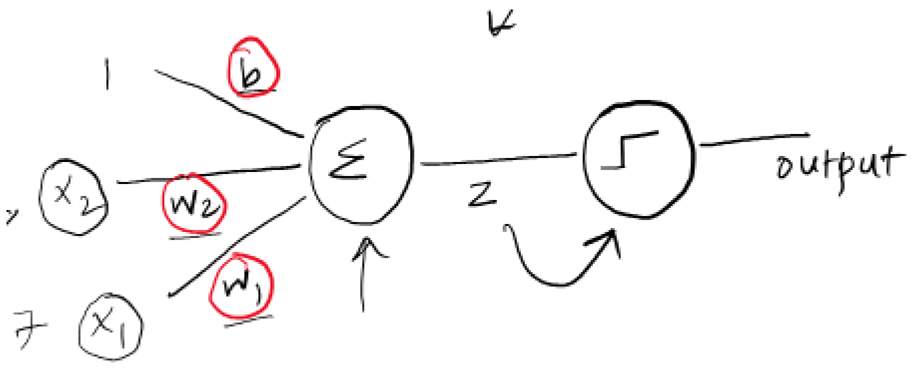
Logistic Regression

- <u>**Classification**</u>Where y is a discrete value
  - Develop the logistic regression algorithm to determine what class a new input should fall into
- Classification problems
  - Email -> spam/not spam?
  - Online transactions -> fraudulent?
  - Tumor -> Malignant/benign

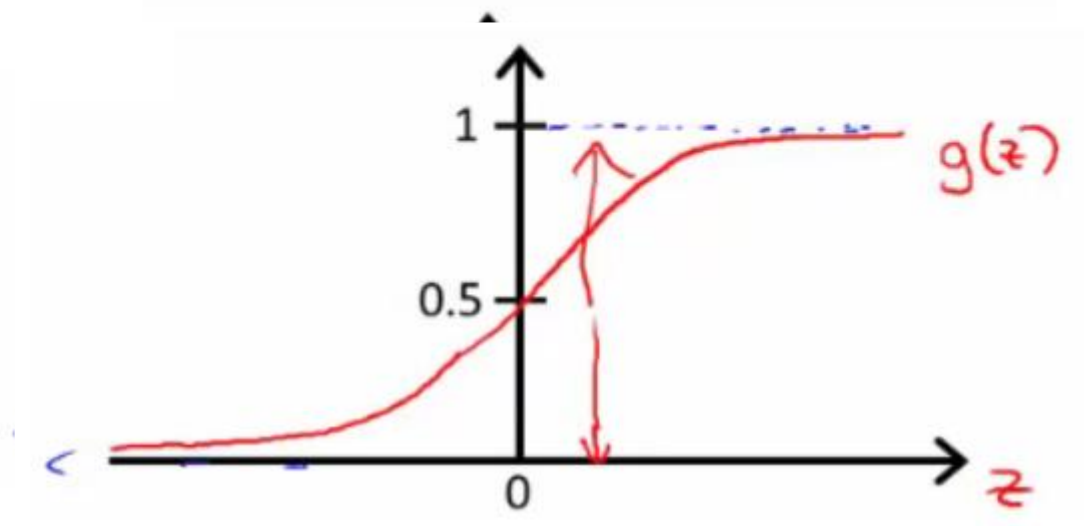


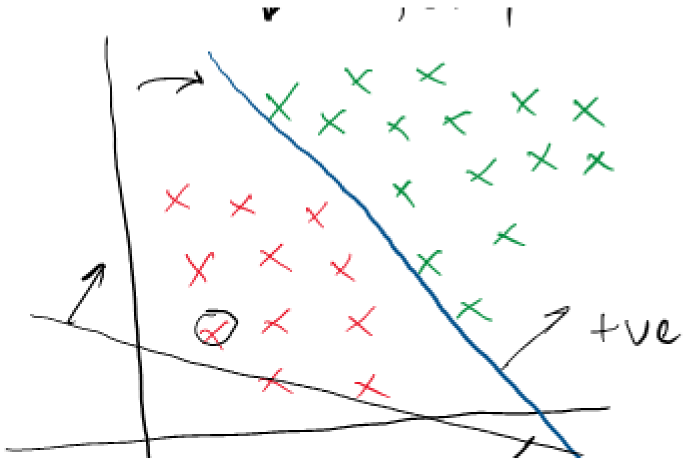
Logistic Regression

$$h_\theta(x) = \frac{1}{1 + e^{-\theta^T x}}$$



$$z = w_1 x_1 + w_2 x_2 + b$$





Cost Function-Logistic Regression

$$\mathrm{Cost}(h_\theta(x), y) = \begin{cases} -\log(h_\theta(x)) & \text{if } y = 1 \\ -\log(1 - h_\theta(x)) & \text{if } y = 0 \end{cases}$$

$$J(\theta) = -\frac{1}{m}[\sum_{i=1}^{m} y^{(i)} \log h_\theta(x^{(i)}) + (1 - y^{(i)}) \log (1 - h_\theta(x^{(i)}))]$$

GD-Backpropgation

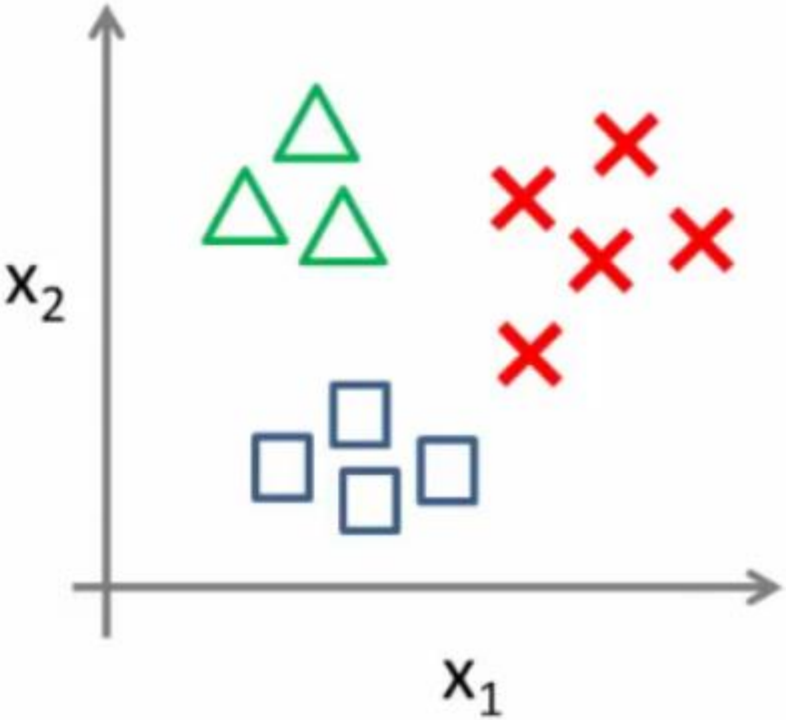
Repeat $\{\ \theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)\ \}$

Multiclass classification

One vs. all classification

SoftMax

Multi-class classification:



- Hypothesis/cost/prediction

$$P(y = k \mid x) = \frac{e^{z_k}}{\sum_{j=1}^{K} e^{z_j}}$$

$$z = Wx + b$$

Where:

- $z = [z_1, z_2, \ldots, z_K]$ are the logits for each of the K classes

Then, apply the **softmax function** to convert logits into class probabilities:

$$\hat{y} = \arg\max_k P(y = k \mid x)$$

yes → 1

no → 2

opt → 3

$$\sigma(z)_i = \frac{e^{z_i}}{\sum_{j=1}^{k} e^{z_j}}$$

$$\sigma(z)_3 = \frac{e^{z_3}}{e^{z_1} + e^{z_2} + e^{z_3}}$$

(Yes)

$$\sigma(z)_1 = \frac{e^{z_1}}{e^{z_1} + e^{z_2} + e^{z_3}}$$

No

$$\sigma(z)_2 = \frac{e^{z_2}}{e^{z_1} + e^{z_2} + e^{z_3}}$$

Training-Multiclassificatin

- One-Hot Encoding

Multiple Models to be trained
Increased Parameters
Increased computation Time

Loss Function-Multi-classification

$$L=-\frac{1}{m}\sum_{i=1}^{m}\sum_{k=1}^{K}y_k^{(i)}\log\left(\hat{y}_k^{(i)}\right)$$

X_1	X_2	Y	Y_{k-1}	Y_{k-2}	Y_{k-3}
X_{11}	X_{12}	1	1	0	0
x_{21}	x_{22}	2	0	1	0
x_{31}	x_{32}	3	0	0	1

$$\underbrace{y_1^{(1)}\log(\hat{y}_1^{(1)})+y_2^{(1)}\log(\hat{y}_2^{(1)})+y_3^{(1)}\log(\hat{y}_3^{(1)})}$$

$$+\underbrace{y_1^{(2)}\log(\hat{y}_1^{(2)})+y_2^{(2)}\log(\hat{y}_2^{(2)})+y_3^{(2)}\log(\hat{y}_3^{(2)})}$$

$$+\underbrace{y_1^{(3)}\log(\hat{y}_1^{(3)})+y_2^{(3)}\log(\hat{y}_2^{(3)})+y_3^{(3)}\log(\hat{y}_3^{(3)})}$$

$$L=y_1^{(1)}\log(\hat{y}_1^{(1)})+y_2^{(2)}\log(\hat{y}_2^{(2)})+y_3^{(3)}\log(\hat{y}_3^{(3)})$$

- Now we have to calculate:

$$\hat{Y}_1{}^{(1)},\quad \hat{Y}_2{}^{(2)},\quad \hat{Y}_3{}^{(3)}$$

$$\hat{Y}_1{}^{(1)} = \sigma - (W_1{}^{(1)}\ X_{11} + W_2{}^{(1)}\ X_{12} + W_0{}^{(1)})$$
$$Y_2{}^{(2)} = \sigma - (W_1{}^{(2)}\ X_{21} + W_2{}^{(2)}\ X_{22} + W_0{}^{(2)})$$
$$Y_3{}^{(3)} = \sigma - (W_1{}^{(3)}\ X_{31} + W_2{}^{(3)}\ X_{32} + W_0{}^{(3)})$$

$$\mathbf{W} = \begin{bmatrix} w_1^{(1)} & w_2^{(1)} & w_0^{(1)} \\ w_1^{(2)} & w_2^{(2)} & w_0^{(2)} \\ w_1^{(3)} & w_2^{(3)} & w_0^{(3)} \end{bmatrix}$$

09- weights to be updated

$$w_1^{(i)} = w_1^{(i)} - \eta \frac{\partial L}{\partial w_1^{(i)}}$$

$$w_2^{(i)} = w_2^{(i)} - \eta \frac{\partial L}{\partial w_2^{(i)}}$$

Derivative Chain Rule:

$$\frac{\partial \mathcal{L}}{\partial w_k} = \frac{\partial \mathcal{L}}{\partial z_k} \cdot \frac{\partial z_k}{\partial w_k}$$

09- weights to be updated

$$w_1^{(i)} = w_1^{(i)} - \eta \frac{\partial L}{\partial w_1^{(i)}}$$

$$w_2^{(i)} = w_2^{(i)} - \eta \frac{\partial L}{\partial w_2^{(i)}}$$

$$y_k^{(i)} \log\left(\hat{y}_k^{(i)}\right)$$

Derivative Chain Rule:

$$\frac{\partial \mathcal{L}}{\partial w_k} = \frac{\partial \mathcal{L}}{\partial z_k} \cdot \frac{\partial z_k}{\partial w_k}$$

- Derivative chain rule

$$\mathcal{L} = -\sum_{j=1}^{K} y_j \log(\hat{y}_j)$$

$$\frac{\partial \mathcal{L}}{\partial w_k} = \frac{\partial \mathcal{L}}{\partial z_k} \cdot \frac{\partial z_k}{\partial w_k}$$

$$\frac{\partial \mathcal{L}}{\partial z_k} = \sum_{j=1}^{K} \frac{\partial \mathcal{L}}{\partial \hat{y}_j} \cdot \frac{\partial \hat{y}_j}{\partial z_k}$$

$$\frac{\partial \mathcal{L}}{\partial \hat{y}_j} = -\frac{y_j}{\hat{y}_j}$$

$$\frac{\partial \mathcal{L}}{\partial z_k} = \sum_{j=1}^{K} \frac{\partial \mathcal{L}}{\partial \hat{y}_j} \cdot \frac{\partial \hat{y}_j}{\partial z_k}$$

$$\hat{y}_j = \frac{e^{z_j}}{\sum_{l=1}^{K} e^{z_l}}$$

$$\frac{\partial \hat{y}_j}{\partial z_k} = \begin{cases} \hat{y}_k(1 - \hat{y}_k) & \text{if } j = k \\ -\hat{y}_j \hat{y}_k & \text{if } j \neq k \end{cases}$$

✳️ Step 3: Apply Chain Rule

Now combine both:

$$\frac{\partial \mathcal{L}}{\partial z_k} = \sum_{j=1}^{K} \left(-\frac{y_j}{\hat{y}_j} \right) \cdot \frac{\partial \hat{y}_j}{\partial z_k}$$

Now we split the sum into two cases:

Case 1: When $j = k$

$$\text{Term} = -\frac{y_k}{\hat{y}_k} \cdot \hat{y}_k(1 - \hat{y}_k) = -y_k(1 - \hat{y}_k)$$

Case 2: When $j \neq k$

$$\text{Term} = -\frac{y_j}{\hat{y}_j} \cdot (-\hat{y}_j \hat{y}_k) = y_j \hat{y}_k$$

$$\frac{\partial \mathcal{L}}{\partial z_k} = -y_k(1 - \hat{y}_k) + \sum_{j \neq k} y_j \hat{y}_k$$

Now factor out $\hat{y}_k$:

$$\frac{\partial \mathcal{L}}{\partial z_k} = -y_k + y_k \hat{y}_k + \hat{y}_k \sum_{j \neq k} y_j$$

But in one-hot encoding:

- Only one $y_j = 1$, the rest are 0
- So $\sum_{j \neq k} y_j = 1$ if $y_k = 0$, otherwise 0

Thus, in all cases:

$$\boxed{\frac{\partial \mathcal{L}}{\partial z_k} = \hat{y}_k - y_k}$$

✅ Final Result:

$$\frac{\partial \mathcal{L}}{\partial z_k} = \hat{y}_k - y_k$$

Simplification of $\frac{\partial z_k}{\partial w_k}$

$$z_k = w_k^\top x$$

$$\frac{\partial z_k}{\partial w_k} = x$$

Then, by chain rule:

$$\frac{\partial \mathcal{L}}{\partial w_k} = \frac{\partial \mathcal{L}}{\partial z_k} \cdot \frac{\partial z_k}{\partial w_k} = (\hat{y}_k - y_k) \cdot x$$

Final Formula:

$$\boxed{\frac{\partial \mathcal{L}}{\partial w_k} = (\hat{y}_k - y_k) \cdot x}$$