WEEK - 3

⇒ We cant use linear regression for classification models.

⇒ LOGISTIC REGRESSION
(CLASSIFICATION PROBLEM)

⇒ we want $0 \le h_\theta(x) \le 1$ (binary classification)

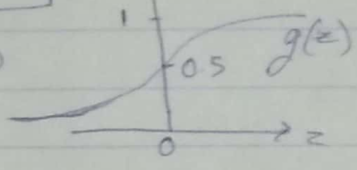
$$h_\theta(x) = g(\theta^T x)$$

$$g(z) = \frac{1}{(1+e^{-z})}$$ (Logistic / Sigmoid Function)

$\therefore$ $\boxed{z = \theta^T x}$

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

⇒



⇒ For +ve output, $P(y=1 | x; \theta)$
-ve output, $P(y=0 | x; \theta)$] $+ = 1$

Probability

⇒ In sigmoid f^n → for $h_\theta(x)$ to be ≥ 0.5,
(+ve outcome)
$z \ge 0$
$\therefore \theta^T x \ge 0$

⇒ Decision Boundary ⇒ The line which separates the +ve area from the -ve area.

$5 - x_1 > 0$
$5 > x_1$

⇒ Decision Boundary ⇒ line that separates the region where the hypothesis predicts Y equals 1 from the region where the hypothesis predicts that y is equal to 0.

⇒ If we use the cost f^n of linear regression as the cost f^n for logistic regression
→ we get a non-convex f^n.
↓
It is very difficult to reach the local minima.

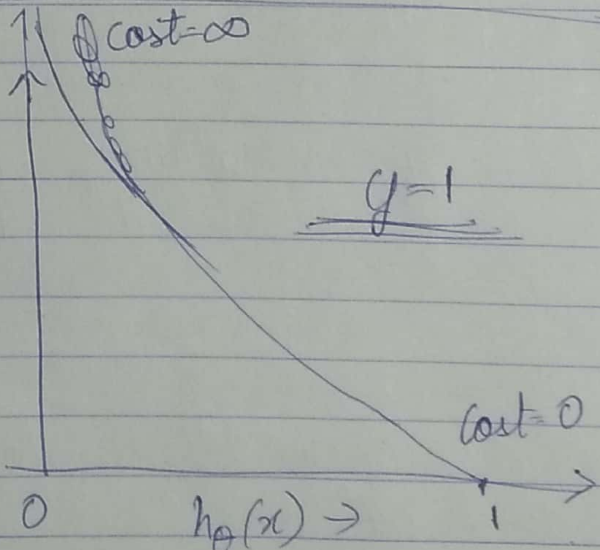
⇒ Cost f^n of Logistic Regression $\boxed{J(\theta) = \frac{1}{m} \sum_{i=1}^{m} \text{Cost}}$

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

for y = 1

⇒ Cost = 0, if y = 1 & if $h_\theta(x) = 1$

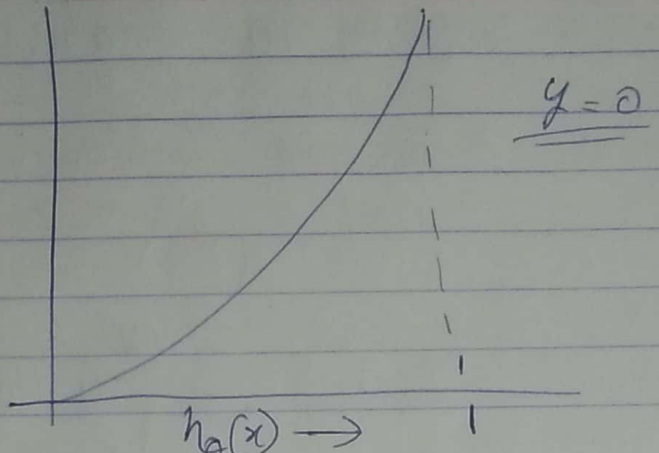
⇒ for $h_\theta(x) = 0$ & $y = 1$
⟶ Cost → ∞



$h_\theta(x) = 0$ → -ve outcome

⇒ for $\underline{y = 0}$

⇒ if $y = 0$ & $h(x) \to 1$
then Cost $\to \infty$



⇒ If $y = 0$ & $h(x) = 0$
Cost = 0

⇒ <u>Logistic Regression Cost f^n</u>

$$J(\theta) = \frac{1}{m} \sum_{i=1}^{m} \text{Cost}(h_\theta(x^i), y^i)$$

$$\boxed{\text{Cost}(h_\theta(x), y) = -y \log(h_\theta(x)) - (1-y) \log(1 - h_\theta(x))}$$

Compact version cost = (1/m) * sum(-y .* log(hypo) - (1-y) .* log(1-hypo));

⇒ This cost f^n can be derived from statistics using the principle of maximum likelihood estimation.

⇒ To minimise $J(\theta)$, we use gradient descent as it has the same eq as in linear regression.

Repeat {

$$\theta_j := \theta_j - \alpha \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_j^{(i)}$$

for all θ_j

} ⇒ (1/m) * ((hypo - y)' * X);

temp = theta

~~theta~~ temp(1) = 0

⇒ We can use feature scaling in logistic Reg.

⇒ Vectorized Implementation of $J(\theta)$ is

$$h = g(X\theta)$$

$$J(\theta) = \frac{1}{m} \cdot \left(-y^T \log(h) - (1-y)^T \log(1-h)\right)$$

⇒ Vectorized Implementation of Gradient descent

$$\theta := \theta - \frac{\alpha}{m} X^T \left(g(X\theta) - \vec{y}\right)$$

⇒ Other ways to minimize $J(\theta)$ →

1) = Conjugate Gradient
2) = BFGS
3) = L-BFGS

} — we do not need to pick α, to do it themselves

& they are faster than gradient descent
But they are more complex.

⇒ For Multiclass Classification Problems,
we use one-~~vs~~-all classifier.

We use binary logistic Reg. on each class to find its probability.

We are basically choosing one class and then lumping all the ~~others~~ others into a single second class.

Cost f^n (Regularized)

⇒ cost = (1/m)*sum(-y.*log(hypo) - (1-y).*log(1-hypo)) + (1/2*m)*sum(temp(2:end).^2);

⇒ Overfitting → If we have too many features the learned hypothesis may fit the training set very well, but fails to generalize to new examples (predict) aka High Variance

⇒ Underfitting or High Bias → not a very good fit to the data (Using too few features)

⇒ To Address overfitting —
1) = Reduce no. of features
2) = Regularization (keep all features, but reduce the magnitude/value of θ_j

⇒ Regularized Cost f^n → keeping all parameters small

$$J(\theta) = \frac{1}{2m}\left[\sum_{i=1}^{m}\left(h_\theta(x^i) - y^i\right)^2 + \underbrace{\lambda \sum_{j=1}^{n} \theta_j^2}_{\text{extra term for regularization}}\right]$$

regularization parameter (helps balance the two sides)

⇒ If the value of λ is very high, then algo results in underfitting

⇒ Gradient Descent with Regularization :-

$$\boxed{\theta_j := \theta_j \underbrace{\left(1 - \frac{\alpha\lambda}{m}\right)}_{\text{this no. is slightly less than 1}} - \alpha\frac{1}{m}\sum_{i=1}^{m}\left(h_\theta(x^i) - y^i\right)x_j^i}$$

for theta ≠ 0

grad = $\frac{1}{m}$*(X'*(hypo-y)) + (λ/m)*temp;

⇒ Normal eq with Regularization

Normal eq ⇒ $\theta = (X^T X)^{-1} X^T y$.

if with Regularization

if $\lambda > 0$,

$$\theta = \Big(\overbrace{X^T X + \lambda \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}_{(n+1) \times (n+1)}}^{\text{this makes it invertible}} \Big)^{-1} X^T y$$

(Identity matrix with 0 at first place)

⇒ Regularized Logistic Regression

$$J(\theta) = \left[\frac{-1}{m} \sum_{i=1}^{m} y^i \log(h_\theta(x^i)) + (1-y^i) \log(1 - h_\theta(x^i)) \right] + \frac{\lambda}{2m} \sum_{j=1}^{n} \theta_j^2$$

⇒