Normal Equations

- Solves for optimum value directly
- Take derivative and equal to zero
- Defining X as a matrix containing all features and Y as the solution vec

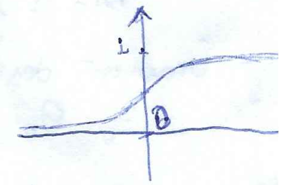
$$\theta = (X^T X)^{-1} X^T y$$

- No need to choose α or to iterate
- Does not scale well to a large number of features

Week 3

Logistic Regression

- Used in classification problems → Linear regression is ineffective
- Model (Binary case)



  - $0 \leq h_\theta(x) \leq 1$
  - Logistic function: $g(z) = \frac{1}{1+e^{-z}}$ → $h_\theta(x) = g(\theta^T x)$
  - h_θ will output the probability of the correct prediction being 1
- Decision Boundary
  - if $\theta^T x \geq 0$ → $y = 1$ else $y = 0$
  - $\theta^T x$ can represent all kinds of polynomials by adding extra features
- Cost
  - Original cost function is not convex in logist regression → gradient descent wouldn't work
  - Defining:

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

case $y=1$: $= -\log(h_\theta(x))$

case $y=0$: $= -\log(1-h_\theta(x))$

    In LR, we have:
    cost zero when prediction correct
    cost blows up when prediction incorrect
  - Simplified: $Cost(h_\theta(x), y) = -y\log(h_\theta(x)) - (1-y)\log(1-h_\theta(x))$
- Gradient Descent
  - Same update rule:

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

  - Vectorized:

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

- Advanced Optimization Algorithms
  - Conjugate Gradient
  - BFGS
  - L-BFGS

  } provided by language
  * fminunc()
- Multiclass Classification
  - ↳ One-vs-all: find h_θ^i for each class and predict whichever maximizes the result

Overfitting

Hypothesis fits training data very well but does not generalize well to new examples

Solutions

- Reduce number of features manually / via model selection
- Regularization by reducing magnitudes of θs
  - ↳ works well with many slightly useful features

Regularization in Linear Regression

- Cost:

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

↳ parametro de regularização

- Gradient descent:

$$\theta_0 := \theta_0 - \alpha\frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)})-y^{(i)})x_0^{(i)}$$

$$\theta_j := \theta_j - \alpha\left[\left(\frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)})-y^{(i)})x_j^{(i)}\right)+\frac{\lambda}{m}\theta_j\right]$$

- Normal equation:

$$\theta = (X^TX+\lambda L)^{-1}X^Ty \qquad L = \begin{bmatrix}0 & & & \\ & 1 & & \\ & & 1 & \\ & & & 1\end{bmatrix}$$

Regularization in Logistic Regression

- Cost:

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

- Gradient Descent:

Same rule as Linear Regression