**Saisha Kashyap - Progress Report**

**17.12.18**

**Machine Learning**

-Completed tutorials till week 4.

-Implemented gradient descent for linear regression and multivariant linear regression. Relevant codes are attached.

**Algorithm for normal equation:**

Cost function J is minimised explicitly by taking it’s derivative with respect to theta(j)s and setting them to zero.



Vectorised:

Theta=pinv(X\*X’)\*X’\*y

However, normal equation method becomes slow when n exceeds 10,000 and iterative process is preferred.

**Logistic regression**

Logistic regression is used for classification problems.

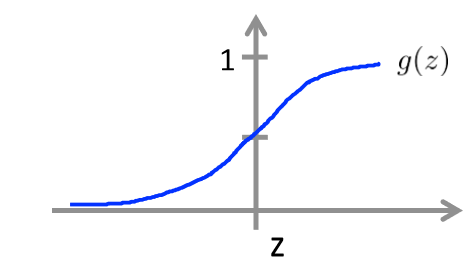
Hypothesis function: g(theta’\*X)

g(z)=1/(1+e^-z)

Thus,

h(x)=1/(1+e(-theta’\*X))

This is called the sigmoid/logistic function.



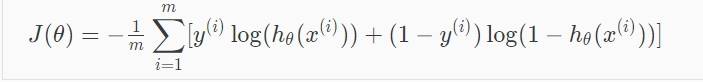
From the graph of sigmoid function, it is clear that when theta’\*X is greater than 0, g(theta’\*X) is greater than/equal to 0.5 which gives y=1.

On the other hand, when theta’\*X is less than 0, g(theta’\*X) is less than 0.5 which gives y=0.

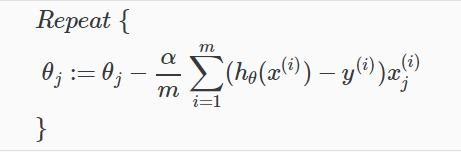
Decision boundary: It is the line which separates the area where y=0 and where y=1. It depends on the hypothesis function and not the training set.

A different cost function is defined for logistic regression since the cost function used for linear regression is not a convex function.

Simplified cost function:



Gradient descent for logistic regression:



**Multiclass Classification:**

One-vs-all: In one-vs-all, we choose one class at a time and lump all the other classes into a single second class. Doing this repeatedly, we apply binary logistic regression to each case and then use the hypothesis which returns the highest value as the prediction.

**Overfitting:**

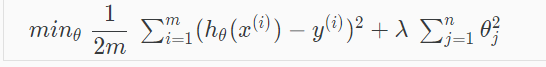
If we have too many features, the learned hypothesis may fit the training set very well, but fail to generalise to new examples. This situation is called overfitting and can be addressed by:

i)Reducing the number of features: Either by manually selecting which features to keep or by model selection algorithm.

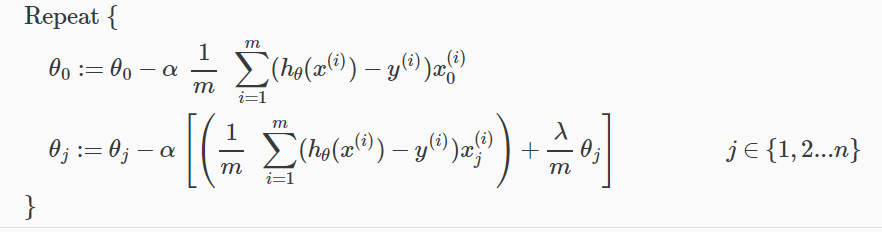
ii)Regularisation

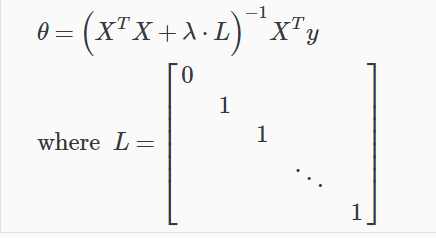
**Regularisation:**

Small values for theta1, theta2..thetan are chosen and the cost function is modified:

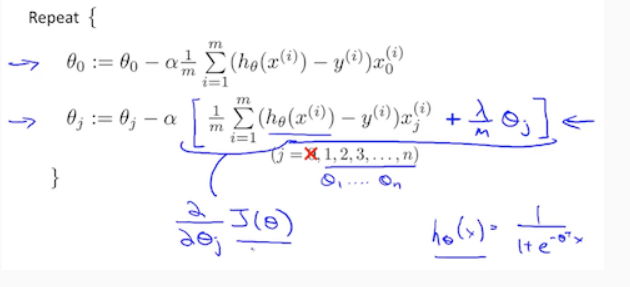


The term lambda is called the regularisation parameter. If lambda is chosen to be too large, it may smooth out the function too much and cause underfitting. Lambda determines how much the costs of our parameters are inflated.

1. Regularised linear regression:
   1. Gradient descent:
   2. Normal Equation:

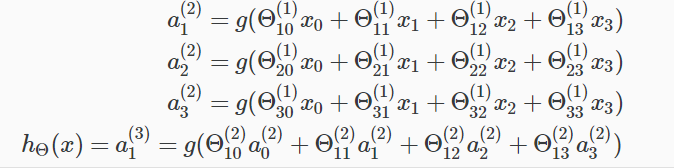


1. Regularised logistic regression:
   1. Gradient descent:



* 1. Optimisation algorithm: (BFGs, L-BFGs)

**Week 4** talks about non-linear hypothesis and is a brief introduction to Neural Networks. The various layers of the neural networks algorithm: input, hidden and output; are discussed along with bias nodes.

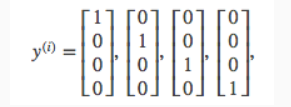


Here h(X) is the output while a is the activation node.

Neural networks can be used to simulate logical operations like AND, OR, NOT, XNOR etc.

Neural networks: Multiclass classification

To classify data into multiple classes, the hypothesis function must return a vector of values rather than a single value.



Where each vector represents a different class.

**Python**

Completed 4 weeks of 7 of the second course of Python offered by University of Michigan on Coursera. Relevant codes are attached.