Progress so far- started with the Andrew NG machine learning course and the Michigan university course for basics in python. Things learnt so far-

Week 1 of ML course-

1] model representation: Examples of supervised and unsupervised learning. In supervised learning we see that with the input training examples, we also feed in the true values or feed in if the condition is satisfied or not. Also defining problems of regression type or classification type.

2] linear regression:

Linear Regression is a supervised machine learning algorithm where the predicted output is continuous and has a constant slope. It’s used to predict values within a continuous range, rather than trying to classify them into categories (e.g. cat, dog). There are two main types:

Simple regression: Simple linear regression uses traditional slope-intercept form, where m and b are the variables our algorithm will try to “learn” to produce the most accurate predictions. x represents our input data and y represents our prediction.

y=mx+b

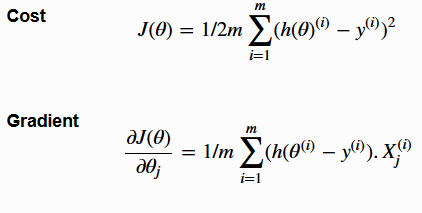
Multivariable regression: It is complex, multi-variable linear equation might look like this, where w represents the coefficients, or weights, our model will try to learn.

f(x,y,z)=w1x+w2y+w3z

3] Cost function & Gradient descent: To map the data on a graph using parameters to correctly predict the values of data points outside our training examples. The algorithm explained is the linear regression with one variable. So in the cases seen, we have one variable and two parameters. Theta0 and theta1 are the basic parameters. So to improve our accuracy and find the best suitable parameters theta0 and theta1, we use cost function.

*J*(*θ*0​,*θ*1​)=2*m*1​*i*=1∑*m*​(*y*^​*i*​−*yi*​)2=2*m*1​*i*=1∑*m*​(*hθ*​(*xi*​)−*yi*​)2

So given below is the cost function and the gradient descent:



Gradient descent: We need it for the estimation of the parameters in the hypothesis. In a graph we put the theta0 and theta1 on the x axis and the y axis. The points on our graph will be the result of the cost function using our hypothesis with those specific theta parameters. We find the minimas in the graph:



Code for the gradient descent in python is ready. I tried writing the code for cost function, but it still doesn’t work fine. There is some bug which I’m failing to find. Will upload it as soon as I’m done with it.