Week1:

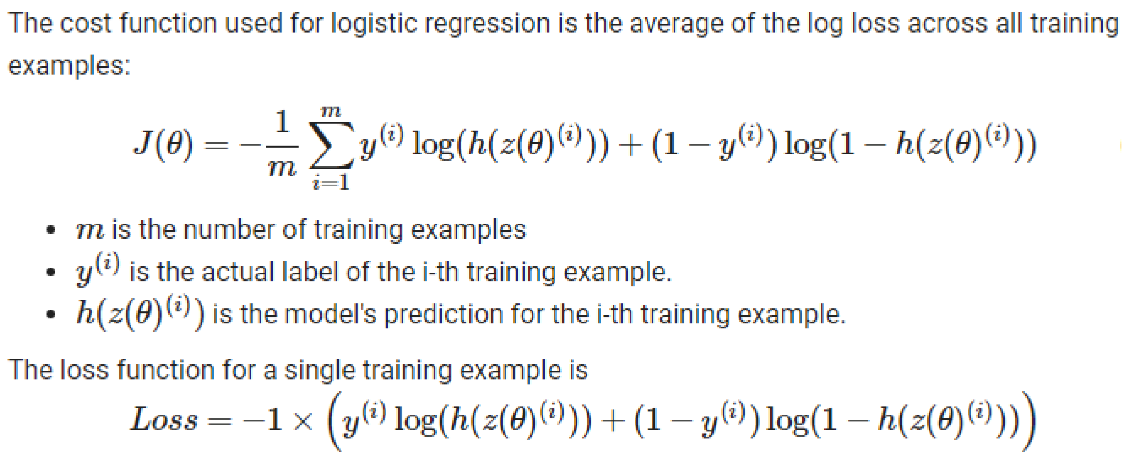
Notebook:	Natural Language Processing		
Created:	31-07-2020 18:28	**Updated:**	31-07-2020 19:11
Author:	Zaid Hasib		
Tags:	Week1		

Week1:

In this week , we learnt about implementation of logistics regression
for sentiment analysis.

Below are the steps that has been done to build the model:

i) Split the data into train and test.
ii) Tokenize the tweets and removed the stop words.
iii) Implemented the sigmoid function for fetching th value 0 and 1
for the sentiment. The sigmoid function is basically convert the
values between 0 and 1.
iv) Logistics regression takes the regular linear regression and
applies a sigmoid to the output of linear regression.
Lets say z=w0x0+w1x+..., then the logistics function will
become:
h(z)=1/(1+e^-z)
v) The cost function in logistics regression is the average of the log
loss across all training examples. The larger the predicted value is away from the actual more ,
the more the loss



The cost function used for logistic regression is the average of the log loss across all training examples:

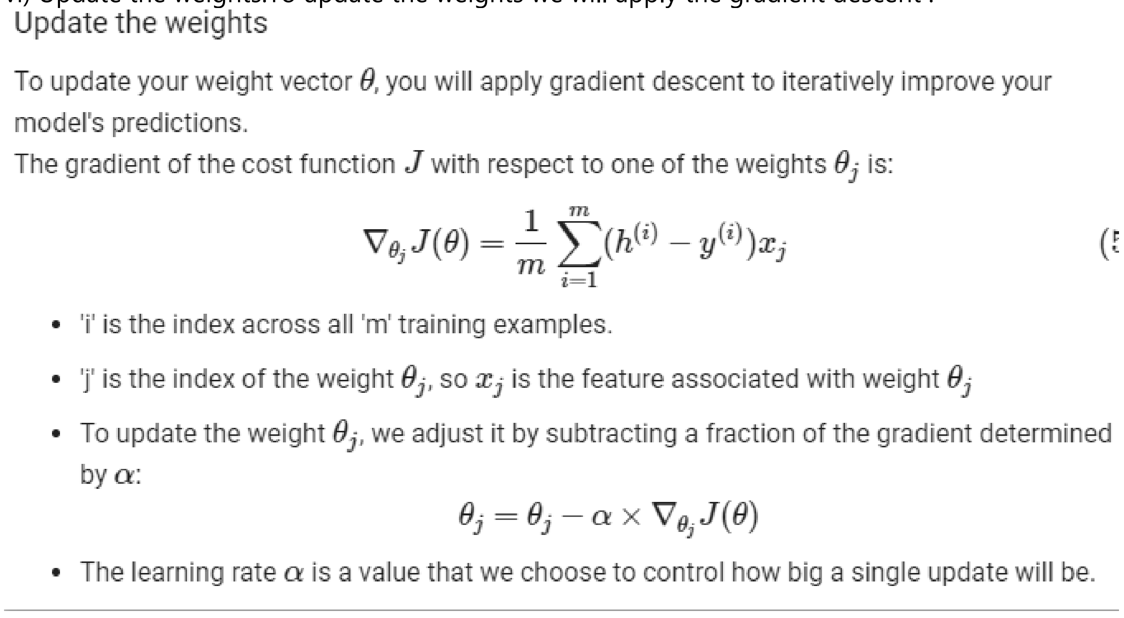
$$J(\theta) = -\frac{1}{m} \sum_{i=1}^{m} y^{(i)} \log(h(z(\theta)^{(i)})) + (1 - y^{(i)}) \log(1 - h(z(\theta)^{(i)}))$$

- m is the number of training examples
- $y^{(i)}$ is the actual label of the i-th training example.
- $h(z(\theta)^{(i)})$ is the model's prediction for the i-th training example.

The loss function for a single training example is

$$Loss = -1 \times \left(y^{(i)} \log(h(z(\theta)^{(i)})) + (1 - y^{(i)}) \log(1 - h(z(\theta)^{(i)}))\right)$$

vi) Update the weights:To update the weights we will apply the gradient descent .

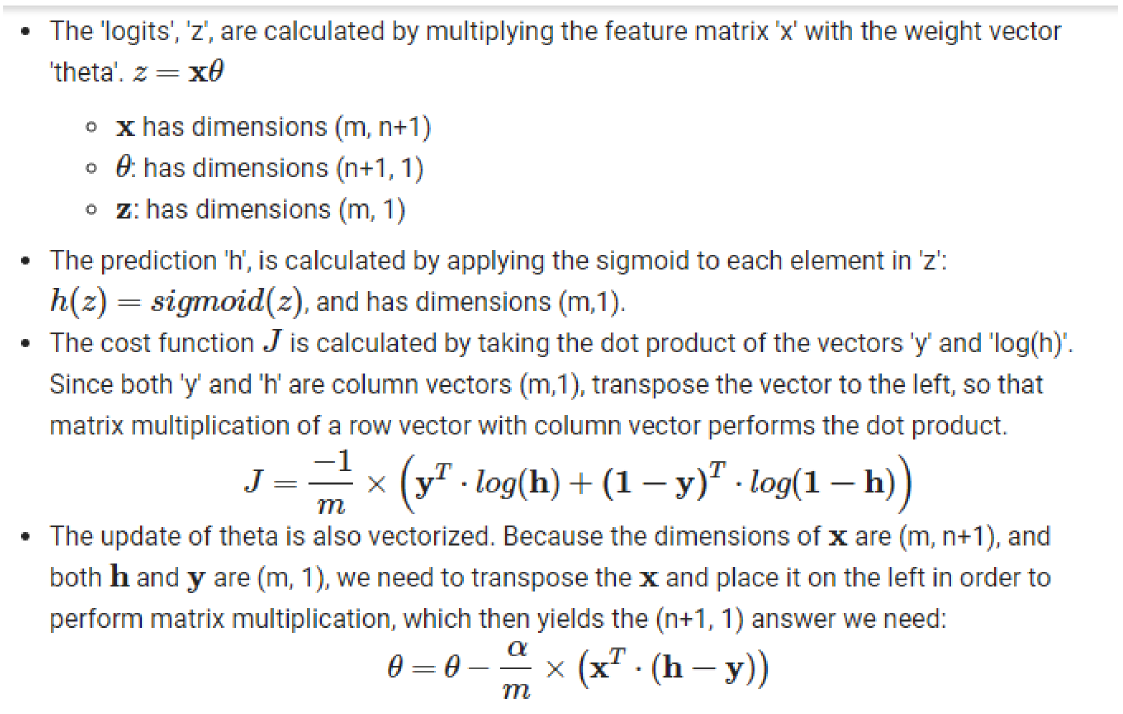


Update the weights

To update your weight vector θ, you will apply gradient descent to iteratively improve your model's predictions.
The gradient of the cost function J with respect to one of the weights θ_j is:

$$\nabla_{\theta_j}J(\theta) = \frac{1}{m}\sum_{i=1}^{m}(h^{(i)} - y^{(i)})x_j \qquad (5$$

- 'i' is the index across all 'm' training examples.
- 'j' is the index of the weight θ_j, so x_j is the feature associated with weight θ_j
- To update the weight θ_j, we adjust it by subtracting a fraction of the gradient determined by α:
$$\theta_j = \theta_j - \alpha \times \nabla_{\theta_j}J(\theta)$$
- The learning rate α is a value that we choose to control how big a single update will be.



- The 'logits', 'z', are calculated by multiplying the feature matrix 'x' with the weight vector 'theta'. $z = \mathbf{x}\theta$
  - $\mathbf{x}$ has dimensions (m, n+1)
  - θ: has dimensions (n+1, 1)
  - $\mathbf{z}$: has dimensions (m, 1)
- The prediction 'h', is calculated by applying the sigmoid to each element in 'z': $h(z) = sigmoid(z)$, and has dimensions (m,1).
- The cost function J is calculated by taking the dot product of the vectors 'y' and 'log(h)'. Since both 'y' and 'h' are column vectors (m,1), transpose the vector to the left, so that matrix multiplication of a row vector with column vector performs the dot product.
$$J = \frac{-1}{m} \times \left(\mathbf{y}^T \cdot log(\mathbf{h}) + (\mathbf{1} - \mathbf{y})^T \cdot log(\mathbf{1} - \mathbf{h})\right)$$
- The update of theta is also vectorized. Because the dimensions of $\mathbf{x}$ are (m, n+1), and both $\mathbf{h}$ and $\mathbf{y}$ are (m, 1), we need to transpose the $\mathbf{x}$ and place it on the left in order to perform matrix multiplication, which then yields the (n+1, 1) answer we need:
$$\theta = \theta - \frac{\alpha}{m} \times \left(\mathbf{x}^T \cdot (\mathbf{h} - \mathbf{y})\right)$$

In this snippet, we basically updated the value of the weights, the stopping criteria to stop the iteration can be the number of iteration, till we will find the convergence.

vii) Next step we will extract the features from the tweets, basically what we are doing, we are taking the words , and put it in a dictionary in a manner like:
{(word,1),count of that word for the class}

viii) Now comes the training part for the model. We will call the function for gradient descent , and pass our test and train data set. By this we will find the parameters.

viii) Then we will test the model with test data set and find the log loss(cost function)