Changes to be done in Q1

1 Analyse the data

- Data is comprised of 1460 rows with 81 features. 38 continuous features and 43 categorical features.

2 Handle Categorical Separately

- Categorical data must be encoded in numeric form i.e one hot encoding.

Let the 43 categorical features in step to be converted into 200 features than:

- Concat numeric and one hot categorial to form dataframe of 1460*238
- Make hidden layer 3 in init function of Neural network class.
- Handle numerical and categorical data separately.
- For numerical data use the same process as used in Q1.
- For categorical data there is separate activation function and error function as described below:

- **Activation Function for Categorical**:Softmax

- **Reason behind Softmax**: Suppose if there are only N categorical values being predicted you'd use one hot encoding on it. Now with this output encoding you want the neural network output layer to have N neurons. In order to compare an output with an input like (0 1 0) instead of using a normal sigmoid or step function, you want the output to be three values between 0.0 and 1.0 that sum to 1.0. In order to understand reason behind this we can suppose that an output is (1.0 1.0 1.0) ,it's not entirely clear how to evaluate that against a categorical value of (1 0 0). To get output that sums to 1.0 you can use the softmax function, where you allow arbitrary output values but then compute the sum of their Math.Exp() values and then divide each output by that sum. For example if some output from the neural net is (2.0 -3.0 0.0) their Math.Exp values are (7.39 0.05 1.00) which sum to 8.44 and dividing each by that sum gives a softmax activation output of (0.87 0.01 0.12). By the way, this computation is tricky and you have to guard against numeric overflow.

- **Error Function to use**:Cross Entropy Function:

- **Reason behind Cross Entropy Function**:You can think of softmax outputs as probabilities. But now comparing a softmax output with a training output becomes somewhat of a problem if you use a standard sum of squared deviations (SSD) approach. For example suppose the softmax output is (0.87 0.01 0.12) as above and the training value is (1 0 0). The SSD would be computed as (1 – 0.87)^2 + (0 – 0.05)^2 + (0 – 1.00)^2. You'd compute all these values and sum them to get the total SSD error. On contrary cross entropy error will be calculated as :(1 * Ln(0.87)) + (0 * Ln(0.01)) + (0 * Ln(1.00) = -0.06 + 0 + 0 = -0.06.You would add all the cross entropies for each training vector up and then multiply by -1. There's a lot to understand here but in essence we only look at the 1 term in the training data (multiplying by the 0s has no effect on the sum). If the output is close to 1.0 the Ln is close to zero.
- To summarize, when using a neural network to classify categorical data, encode the output categorical data using one hot encoding use the softmax activation function to generate output, and use cross entropy to measure error.
- Rest all process is same.