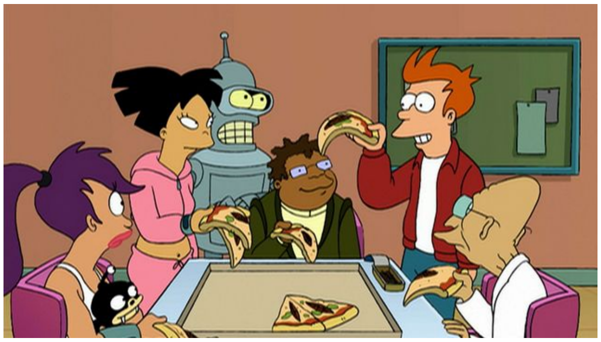
HOMEWORK #2:

Logistic Regression, and pizzas.... and cats.

Due Date: Tuesday, October the 24th, 11:59:59pm

Problem:

Dr. Farnsworth is exploring a new venture to fund his research: Pizzas for Cats! Earth cats that is, not evil space cats. In order to develop the most tasty and acceptable toppings for his pizza-for-cats, Dr. Farnsworth has tested multiple different topping combinations against test subject cats, and recorded if the combination was or was not accepted by the test subjects. Now, he wants to train a predictor from the gathered data. Help Dr. Farnsworth build a predictor to know whether a topping combination will be accepted by cats. You will train a "single neuron" to perform logistic regression on the data collected by Dr. Farnsworth.



Everybody loves pizza, isnt it?

Input:

The given file `pizzacatdata.txt` contains the data for this logistic regression problem. Each row of the data file consists of 5 boolean [0,1] values. The first 4 values indicate weather the pizza tested had Pepperoni, Sausage, Mushroom and Cheese, the last value indicates whether the cat subject accepted the pizza (1) or not (0).

Gradient Descent:

Once again, you will implement gradient descent, and will use Least Squares as the error measurement for your neuron. However, this time the output of the neuron $Y\hat{}(e)$ is given by:

$$Y\hat{}(e) = \mathrm{sigmoid}\left(\Sigma_{i=0}^{n} (W_i * X_i(e)) \right)$$

Where sigmoid(x) is the well known function:

$$\text{sigmoid}(x) = 1 / (1 + e^{-x}) \qquad \text{sigmoid}'(x) = \text{sigmoid}(x)*(1 - \text{sigmoid}(x))$$

You shall implement **incremental** gradient descent. This means that you will be updating the weights after **each** example. Your program should perform **5000** iterations of the gradient descent. Initialize the weights randomly **in the range [-100...100]**. Experiment with different learning rates η to find the best fit possible.

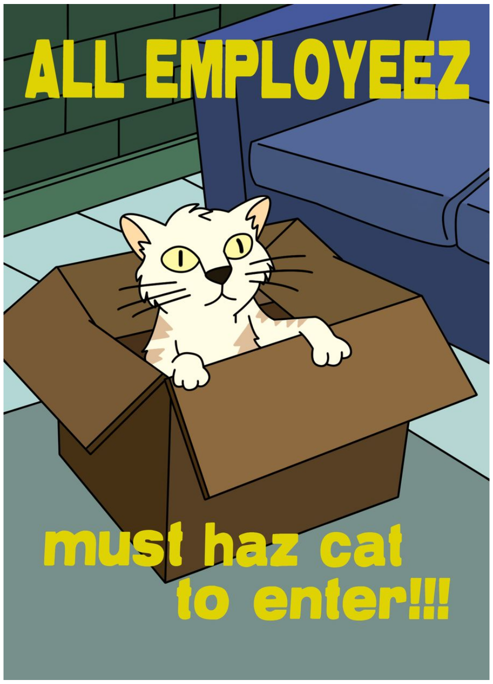
Submission Guidelines:

You will submit through the department's Unix machines using the command:

```
cssubmit 5001 a 2
```

Your submission will consists of the following components:

1. Your program files.- Submit all necessary files. Your program should compile and run in the Unix systems. Your program should read the input data from a file called 'pizzacatdata.txt', (in the same location as your program), and produce output formatted like in the sample shown below.
2. A make file that should build and run your program.
3. A report, in PDF format, in which you state the final weights of your neuron, the learning rate you used to achieve such performance, state the sum of squares error of your final weights.



The internet was built for them!

Sample Output:

Output

```
CS-5001 : HW#2 : Logistic Regression.
Programmer: Dr. Hubert J. Farnsworth
No cats were hurt gathering this data.

Using learning rate eta = 0.050
After 5000 iterations:
Sum of Squares Errors = 0.1234
Weights:
w0 = 3.1415
w1 = 42.0
w2 = 1.0
w3 = 777.0
w4 = 1.6180
```

Pseudocode:

```
PROCEDURE Logistic_Regression_Learner
GIVEN:
    Ex[0..n] : examples, each a <$X_0$ .. $X_n$, Y> tuple.
LOCAL:
    w[n] : weights
    Randomize w[0 .. n].
    REPEAT 5000 times
        FOR EACH example e = <$X_0$ .. $X_n$, Y> DO
            compute yCap ( from w[0 .. n] and $X_0$ .. $X_n$ )
            Error := Y - YCap
            FOR EACH weight w[k] DO
                update w[k] using eta, Error, Ycap, and  $X_k$

    output w[]
```

For Logistic Regression [link]

END.