README for Image Classification

by Matt Bower, Armin Grossrieder

OUR ALGORITHMS

Naive Bayes

- for each image in the training set, extract features (pixels) and store in frequency table
- for each test image, P(D=d I E) = P(E I D=d) P(D=d)

Perceptron

- for each image
    - for each pixel
        - P(D=d I E) = (#samples(total) * #samples(XIY)) / (#samples(X) * #samples(Y))
- took this from a graphical perspective; saw there were x amount of input nodes and y amount of output nodes
- made predictor a matrix function; weights of connections between inputs and outputs * inputs matrix
- we took this matrix _____ and we made a cost function where we took the predicted Y values subtracted by the actual Y
- then we made a cost y function to find the gradient of the function
- then we passed the objective function and the derivative to the pystat optimize function to give us the optimal set of weights that would give the best predictions

Neural Network

- saw that the perceptron performed much worse than Bayes formula
- sought out to improve it to add a hidden layer into the perceptron model in order to increase the accuracy of the model
- added a signmoid function in order to increase the accuracy- this would highlight the values that had the most effect on the output
- after doing this, it dramatically increased the run time, so we had to limit the amount of call backs the optimizer could make such that we wouldn’t overload main memory
    - this led to impractical run times
    - this model can only be used with a GPU to take advantage of the matrix multiplications

PROBLEMS

- probabilities not adding up
- probabilities equating 0 (before smoothing)
- running out of memory
- debugging… unsure whats going on
    - solution: struct and print_struct methods
- accuracy
- formatting data for algorithms
- everything just runs slower on the second iteration on any given loop…