<div align="center">CS181 Assignment 2</div>

Joy Ming and Alisa Nguyen (25 February 2013)

1. Perceptron activation rule

   (a) Bright-or-dark is not able to be recognized by a perceptron. One possibility is to have the weights of all $n = 9$ pixels be initialized to $1/n = 1/9$. Then, multiplying the value if the pixel is one (1) or off (0) with the weights will give something greater than 0.75 if at least 75% pixels are on and will give something less than 0.25 if at least 75% pixels are off. However, this function of greater that 0.75 and less than 0.25 is piecewise and not differentiable. There cannot exist a perceptron that does.

   (b) Top-bright is also able to be recognized by a perceptron. This perceptron would have $1/3$ weights for the top three pixels and $-1/6$ weights for the bottom pixels. If both have the same fraction of pixels on in both rows then the result should be 0. However, if the top has a larger fraction it would be positive and if the bottom has a larger fraction it would be negative. Therefore, simply checking to see if the result is greater than zero can tell if a larger fraction of pixels is on the top row than in the bottom two rows.

   (c) It is not possible to present a perceptron that can check to see if a set of pixels that are on is connected because it is not possible to model when different points of data are correlated. Suppose for contradiction that there is a perceptron that could check to see if a set of pixels that are on is connected. Then for PROOF.

2. Four different possible learning algorithms

   (a) Decision trees
   (b) Boosted decision stumps
   (c) Perceptrons
   (d) Multi-layer feed-forward neural networks

3. Implementing a neural network

   5. We normalize the input values from $[0, 255]$ to between 0 and 1 so we are able to better utilize the logistic sigmoid function.

   6. Experiment wit simple networks

   (a) We used a learning rate of 0.1 based on this DATA:
   (b)   i. Yes/No we are/are not in danger of overfitting by training for too many epochs.
        ii. A good number of ephocs to train for is
            It is important that we use a validation set rather than the actual test set to tune the number of epochs because
   (c) The training performance is
       The validation performance is
       The test performance is

   7. Experiment with networks with a single hidden layer

   (a) We decided that we wanted to use a learning rate of FILLIN for 15 fully connected units in the hidden layer and FILLIN for 30 fully connected units in the hidden layer based on DATA.

<div align="center">1</div>

(b) We determine when to stop training by

(c) GRAPH HERE

(d) We used FILLIN epochs for 15 hidden units and FILLIN epochs for 30 hidden units.

(e) Based on these experiements, we would use the network structure with FILLIN hidden units because FILLIN

(f) The test set performance of the network we chose is FILLIN. This compares to the committee of perceptrons by FILLIN.

8. CustomNetwork

(a) We chose to implement

(b) Our new network's test performance

4. Changing error function

(a) Error function C is implementing the idea of regularization, or that the less complex solution, or one that has less dimensions or degrees, is more desirable others that might work. In the new error function, different values are added onto the original loss function that will increase the sum depending on the increased complexity or layers of the network. This will penalize the loss function by squared weight values, making it more crucial to create less complicated graphs to minimize the loss function.

(b) Our weight update rule for the error function C derived from gradient descent is: