Phillip Martin – 9782-2936

Homework 2 Solutions CAP 6610

Problem 1.

Part A.

When my code is initially run, it splits up the dataset given from bank.csv into a 50 / 50 training/test set split. The 16 features for a given sample are made into a feature vector consisting of floats based on two cases:

1. If the feature is an integer, convert it to a float.
2. If the feature is a string, consult a data structure which will assign it a unique value from 1 to the number of possible different responses for a given feature. (E.g., we might assign, for the "Marital" feature, "married" = 1, "single" = 2, "divorced" = 3)

The data structure used is a hashmap containing (key, value) pairs where the keys are the index of the feature ("Job" = 1, "Marital" = 2, "Education" = 3, etc) and the values are another hashmap, where the (key, value) pairs are the responses and unique indices as outlined above in the second case. The output values of the dataset are assigned as "yes" = 1 and "no" = -1.

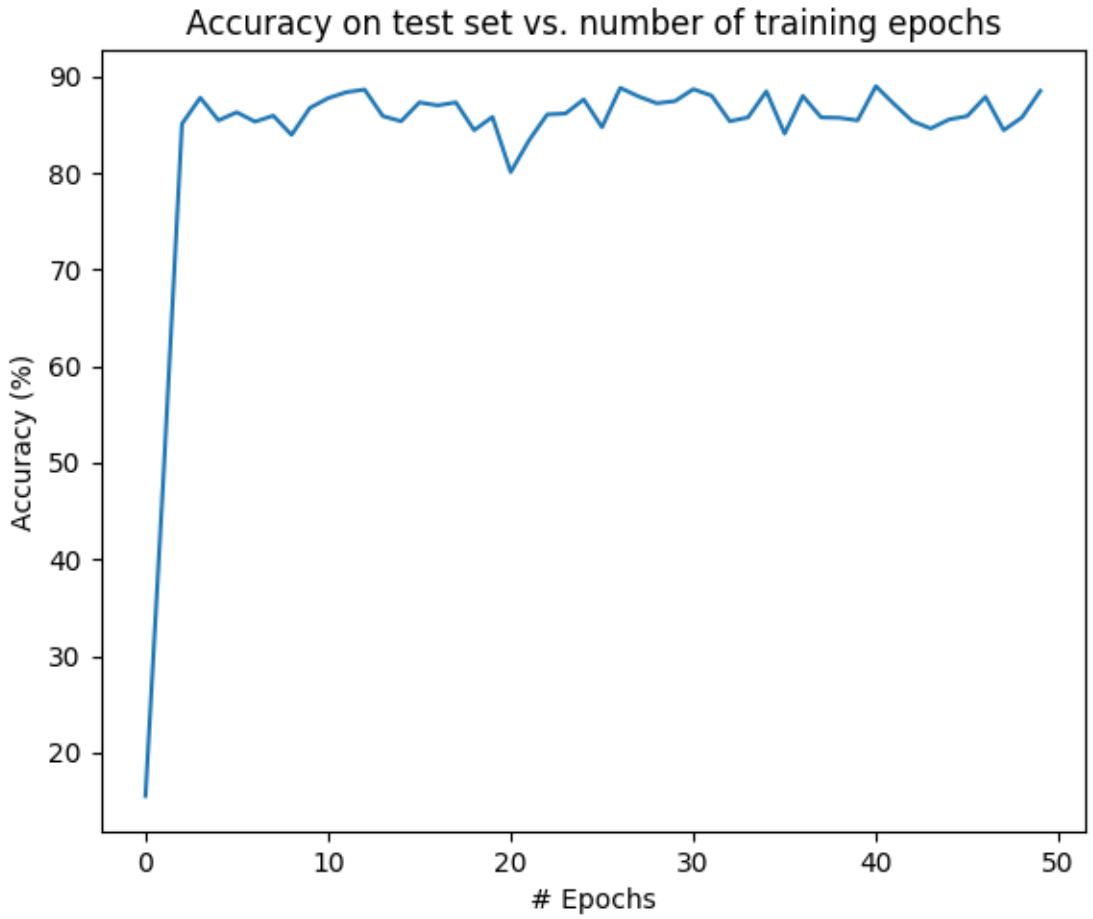
A single sign-activation perceptron then has its weights trained on all the samples in the training set, with the following update rule:

if (sign($W \cdot X^i$) != Y_i):

$$W = W + \eta Y_i X^i$$

where W is the weight vector, η is the learning rate, Y_i is the output for a given sample *i*, and X^i is the feature vector for a given sample *i*.

The following are the results from a sample run of the sign-activation perceptron. The learning rate was set to be 0.02, and the max number of training epochs was set to be 50.



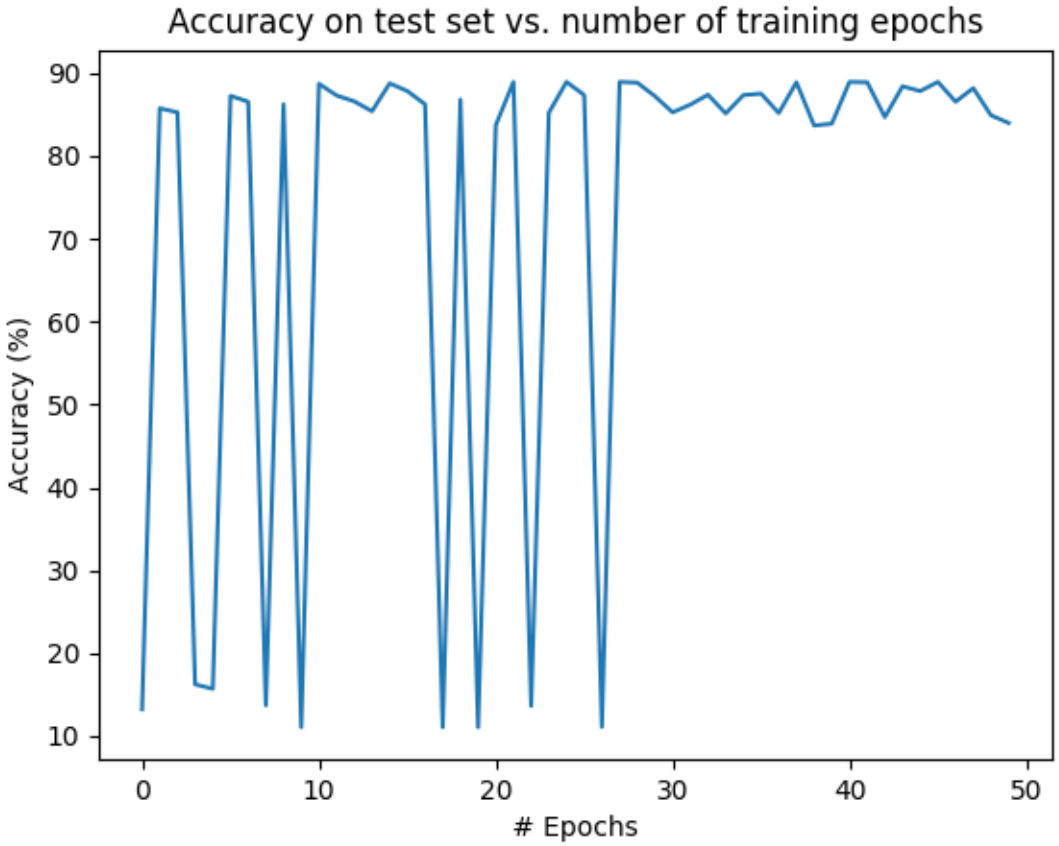
Best accuracy on Test Set: 88.982300885 %

Part B.

The only change here is that we use a sigmoid-activation function rather than a sign-activation. In this case, the update rule is:

$$output = \sigma(W \cdot X^i)$$
$$W = W + \eta(Y_i - output) * (1 - output) * output * X^i$$

The following is a sample run of the algorithm. The learning rate is kept the same along with the maximum number of training epochs as in part A.



Best accuracy on Test Set : 88.9380530973 %

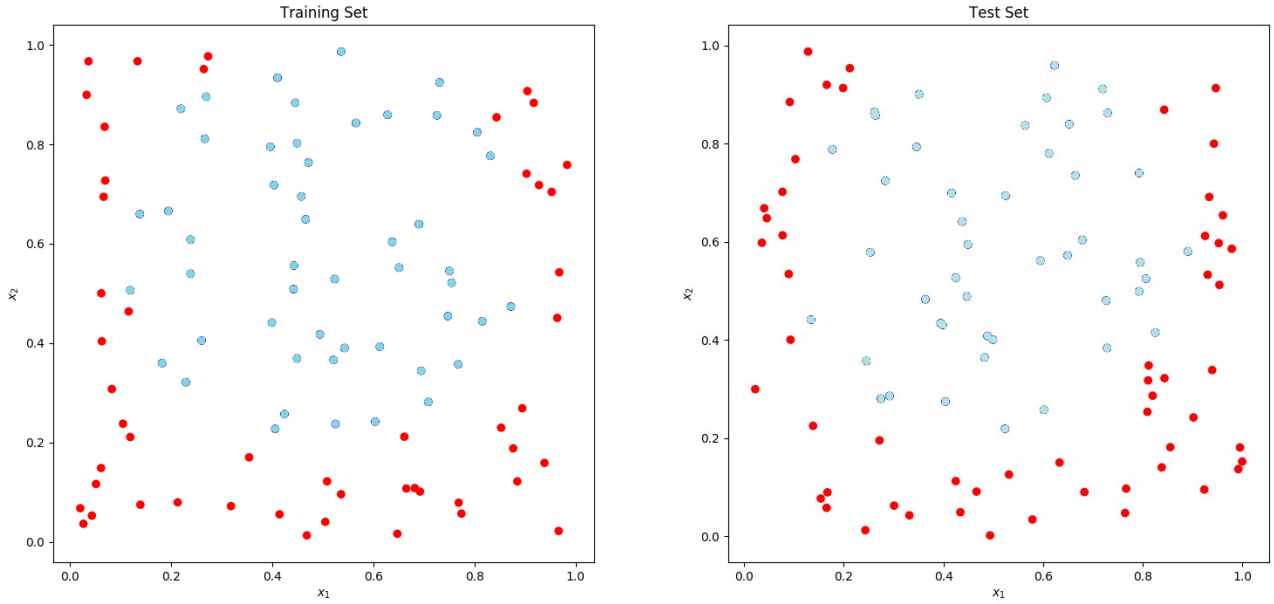
Phillip Martin – 9782-2936

Homework 2 Solutions

CAP 6610

Problem 2.

Initially a training and test set of 100 samples each are randomly generated on the unit square:



The weights corresponding to the hidden nodes in the first layer are stored in a 10 x 3 matrix Θ_1, where each column represents the set of weights on the biases, x_1 inputs, and x_2 inputs, respectively. The weights for the node in the output layer are stored in a 10 x 1 matrix Θ_2, where each entry is the weight on the respective hidden node input. A feedforward pass is computed using sigmoidal activations at each node. Then, the weights are updated in the following manner via backpropagation:

$$\begin{aligned} \mathrm{h} &= \sigma(\Theta_1 \cdot X^i) \\ \mathrm{output} &= \sigma(\Theta_2^T \cdot h) \\ \delta_2 &= (Y_i - output) * (1 - output) * output * h \\ \delta_1 &= (\delta_2 * \Theta_2 * (1 - h)) \cdot (X^{iT}) \\ \Theta_1 &= \Theta_1 + \eta\delta_1 \\ \Theta_2 &= \Theta_2 + \eta\delta_2 \end{aligned}$$

Where h is the output of the first hidden layer, output is the output of the node in the second layer, and δ_1 are the derivatives of the cost function with respect to the weights in Θ_1 - likewise for δ_2. Note that in calculating δ_1, we do the element-wise multiplication for the terms δ_2, Θ_2, and $(1 - h)$ before doing a dot product with the input sample.

To find the hyperplanes for each hidden node, we notice that we wish to solve:

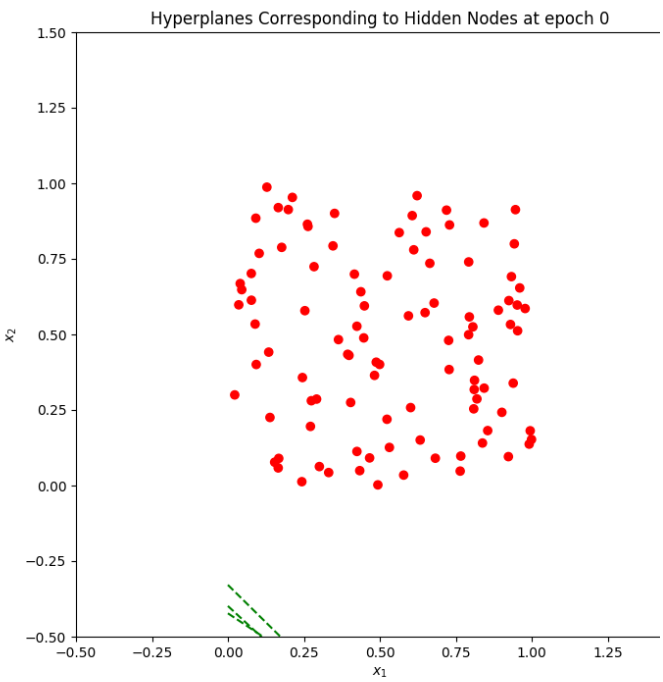
$$\theta_0 + \theta_1 x_1 + \theta_2 x_2 = 0$$

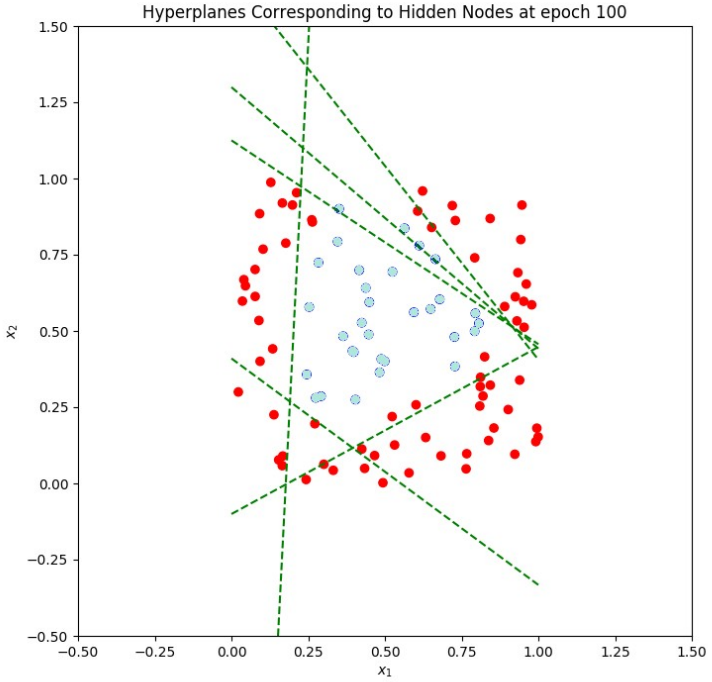
Since this easily takes the form of a line:

$$x_2 = -\frac{1}{\theta_2}(\theta_0 + \theta_1 x_1)$$

we solve for x_2 when $x_1 = 0$ and $x_1 = 1$ for every hidden node, since these are the bounds of our unit square.

The following is a sample run of the algorithm. Accuracies were recorded as a function of increasing training epochs, with initial weights randomly initialized at each new training session. The hyperplanes and resulting classifications by the model at each number of training epochs on the test set were arbitrarily selected, albeit with one showing the best results achieved. The maximum number of training epochs was set to 200, and the learning rate was set to 1.

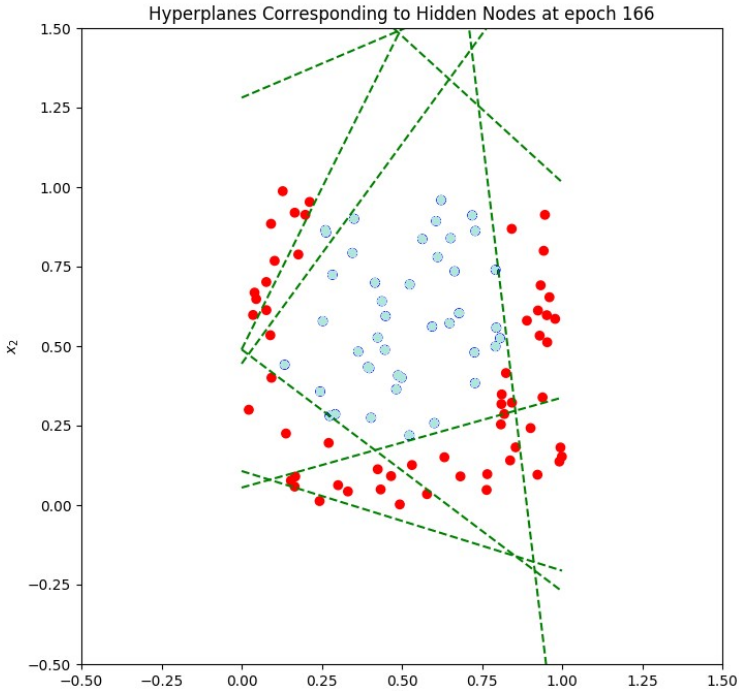


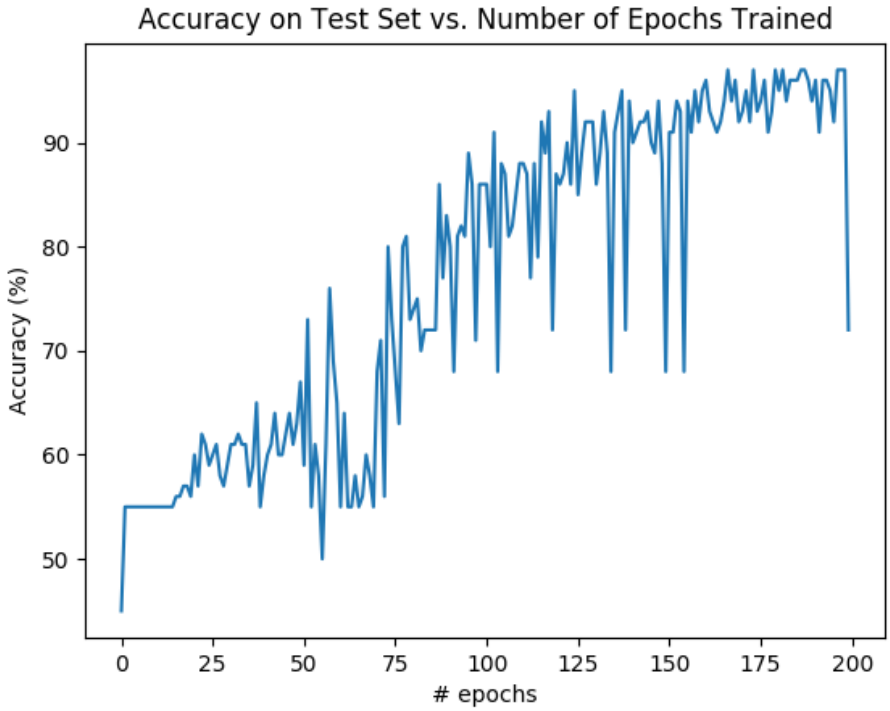


Phillip Martin – 9782-2936

Homework 2 Solutions

CAP 6610





Best Accuracy: 97.0 %

Note that not all of the hyperplanes are visible. This is because they likely lie outside of the unit square.

Problem 3.

The structure of the RNN is outlined in the problem and is unrolled into a feedforward network. The gradients are similar to those of the backpropagation derivation above, except that the gradients accumulate with each timestep in the network.

When trained for 20,000 epochs at a learning rate of 2.5, the network was able to get as close to the desired output as was possible – the difference between the output of the network and the desired output was of order 10^{-3} at most. A sample run comparing 20 samples of output vs. 20 samples of desired output is shown below. The corresponding weights and biases at this result are listed as well.

Desired output:
[[0.6424751 0.33593409]
 [0.65104343 0.33566692]
 [0.63592755 0.35266305]
 [0.60248829 0.33702818]
 [0.62827465 0.34557695]
 [0.62488345 0.33730368]
 [0.71685975 0.34885799]
 [0.60481151 0.35905157]
 [0.70889782 0.33254788]
 [0.69343781 0.35771957]
 [0.60051987 0.3447645]
 [0.69328937 0.36744683]
 [0.67601045 0.36427006]
 [0.61793377 0.37782069]
 [0.68004156 0.35582717]
 [0.62621264 0.37803151]
 [0.59732616 0.34227722]
 [0.61407857 0.3498357]
 [0.61611321 0.36907926]
 [0.6294281 0.36957359]]

Model output:
[[0.64459339 0.33612006]
 [0.65337536 0.33636505]
 [0.63605348 0.35035875]
 [0.60613848 0.33486881]
 [0.62929501 0.34341289]
 [0.62693129 0.33627367]
 [0.7170189 0.35200344]
 [0.60586481 0.35445172]
 [0.71136016 0.33696254]
 [0.69414169 0.35923502]
 [0.60359649 0.34108193]
 [0.69297253 0.36950898]
 [0.67582875 0.3648056]
 [0.61577201 0.37510828]
 [0.6808964 0.35638665]
 [0.62360531 0.37599083]
 [0.60109164 0.33881234]
 [0.6152624 0.34635438]
 [0.61503796 0.36548388]
 [0.62768574 0.36699887]]

$b_1 = -0.862522129216$
$w_1 = 3.25851876326$
$\hat{w}_1 = 1.63501458012$

$b_2 = 0.803378044789$
$w_2 = -1.8639978978$
$\hat{w}_2 = -0.726998050301$