CS6140 Machine Learning Fall 2014 Homework 2, Wei Luo

PROBLEM 1

B) Training and testing performance across all four learning algorithms is:

	Decision or Regression Tree	Linear Regression (Normal Equations)	Linear Regression (Gradient Descent)	LogisticRegression (Gradeint Descent)
Spambase	Train ACC: 0.91 Test ACC: 0.90	Train ACC: 0.92 Test ACC: 0.88	Train ACC: 0.91 Test ACC: 0.90	Train ACC: 0.92 Test ACC: 0.91
Housing	Train MSE: 26.49 Test MSE: 24.28	Train MSE: 22.08 Test MSE: 22.63	Train MSE: 22.13 Test MSE: 22.58	N/A.

C)
Confusion Matrix for Decision Tree:

		Prediction 1	Prediction 0
Label	1	152	29
	0	14	264

Confusion Matrix for Linear Regression:

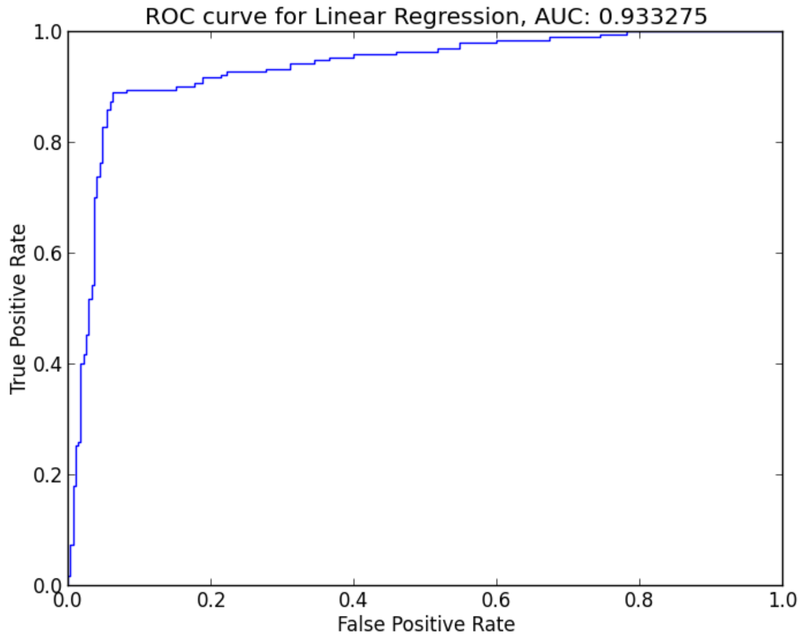
		Prediction 1	Prediction 0
Label	1	162	19
	0	22	256

Confusion Matrix for Logistic Regression:

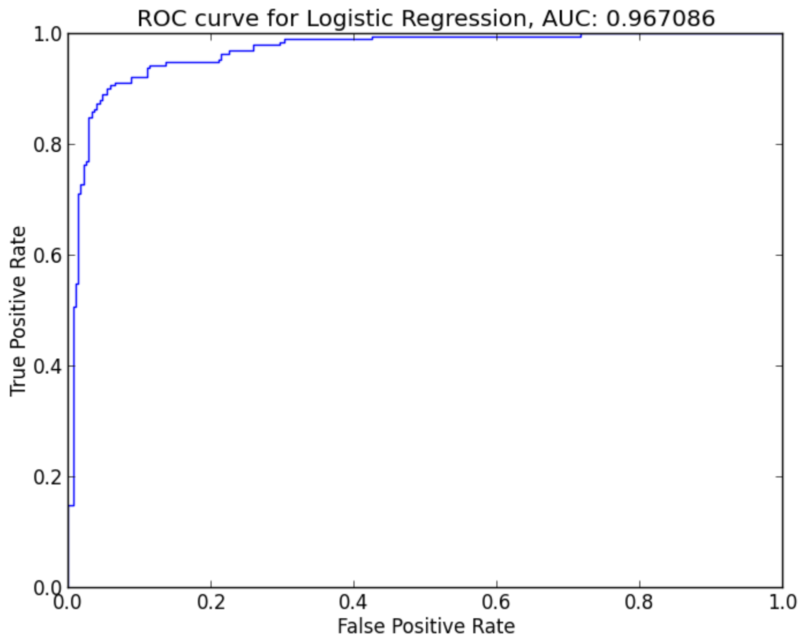
		Prediction 1	Prediction 0
Label	1	154	27
	0	10	268

D)

ROC curve for Linear Regression:



ROC curve for Logistic Regression:



PROBLEM 2

Iteration 1, total_mistake 136
Iteration 2, total_mistake 68
Iteration 3, total_mistake 50
Iteration 4, total_mistake 22
Iteration 5, total_mistake 21
Iteration 6, total_mistake 34
Iteration 7, total_mistake 25
Iteration 8, total_mistake 0
Classifier weights: -14. 2.52873259 5.70717051 8.52231457 11.32560723
Normalized with threshold: 0.18062376 0.40765504 0.60873676 0.80897195

PROBLEM 3

b) After training this way, we get a encoder-decoder. With the trained weights, we can encode our data to hidden variables which reduced the size of data. Then the exact data will be kept secret without the weights in our encoder-decoder mechanism. Then, once we need the data, we can decode it with the weights. The weights here are just like keys to encode and decode data.

c) This encoder-decoder scheme cannot work with one or two hidden variables. Because, the values we get from the sigmoid function is more like binary values. They are distributed near two value. So with n hidden variables, we are more like using n binary values. With that, we can only encode 2^n values. So with N input/output values, we need at least $\lceil log_2 N \rceil$ hidden variables.

PROBLEM 4

With our sigmoid function $f(x)$, for a 3-layer neural network, the outputs can be written as

$$g_k(\mathbf{x}) = z_k = f(\sum_j w_{kj} f(\sum_i w_{ij} x_i + w_{j0}) + w_{k0})$$

The inner f is for hidden variables. If we change it to a linear function, say $lf(x) = ax + b$, then:

$$\begin{aligned}
g_k(\mathbf{x}) = z_k &= f(\sum_j w_{kj} lf(\sum_i w_{ij} x_i + w_{j0}) + w_{k0}) \\
&= f(\sum_j w_{kj}(a_j(\sum_i w_{ij} x_i + w_{j0}) + b_j) + w_{k0}) \\
&= f(\sum_j w_{kj}(a_j \sum_i w_{ij} x_i + a_j w_{j0} + b_j) + w_{k0}) \\
&= f(\sum_j w_{kj} a_j \sum_i w_{ij} x_i + \sum_j w_{kj}(a_j w_{j0} + b_j) + w_{k0}) \\
&= f(\sum_i (\sum_j w_{kj} a_j w_{ij}) x_i + \sum_j w_{kj}(a_j w_{j0} + b_j) + w_{k0})
\end{aligned}$$

Let $w'_{ki} = \sum_j w_{kj} a_j w_{ij}$, $w'_{k0} = \sum_j w_{kj}(a_j w_{j0} + b_j) + w_{k0}$, we have

$$g_k(\mathbf{x}) = f(\sum_i w'_{ki} x_i + w'_{k0})$$

Which is equivalent to a two-layer network output.
Therefore, a three-layer network with linear hidden units can be reshaped to a two-layer network.
With a two-layer network, we cannot solve non-linearly separable problem XOR because: Say, we have weights w_1 w_2 w_3, basis b, and a threshold t for the network. For the XOR problem, we need:
(1) $w_0 b + w_1 0 + w_2 1 \geq t$ (2) $w_0 b + w_1 1 + w_2 0 \geq t$
(3) $w_0 b + w_1 0 + w_2 0 < t$ (4) $w_0 b + w_1 1 + w_2 1 < t$
From (1)(2), we have $2w_0 + w_1 + w_2 \geq 2t$, from (3)(4), we have $2w_0 + w_1 + w_2 < 2t$ which is a contradiction due to the XOR problem itself is not linearly separable. We can not solve this problem with such network. Similary for other non-linearly separable problems since the function is linear.

PROBLEM 5