Additional Notes on Lab 2

- Linear Regression Vs. Logistic Regression

	Activation Function	Error function (one training example)	Gradient Descent Algorithm
Linear Regression	**Linear** function	**half of the squared error** $\frac{1}{2}(\hat{y}-y)^2$	$\mathbf{w} \leftarrow \mathbf{w} + \eta\,(y-\hat{y})\mathbf{x}$ where $\hat{y} = f(n) = n$ $= w_0 x_0 + w_1 x_1 + \ldots + w_M x_M$
Logistic Regression	**Logistic** function	**cross-entropy error** $-(y \ln\hat{y} + (1-y)\ln(1-\hat{y}))$	$\mathbf{w} \leftarrow \mathbf{w} + \eta\,(y-\hat{y})\mathbf{x}$ where $\hat{y} = \sigma(n) = \frac{1}{1+e^{-n}}$ $= \frac{1}{1+e^{-(w_0x_0+w_1x_1+\cdots+w_Mx_M)}}$

- You can follow the Linear Regression procedure to implement the Logistic Regression. However, unlike the Linear Regression, the Logistic Regression estimates the output value y as $\sigma(n)$. Thus, $\hat{y} = \sigma(n)$ and $0 < \hat{y} < 1$.
- You can implement either the Stochastic version or the Batch version.
- Notice that you need to convert a probability to a class label when you predict the class label for a testing example. Thus, the prediction is 1 if $\hat{y} \geq 0.5$; otherwise, it is 0.
- The following gives an example of plotting the final decision line and training/testing examples in one figure.