ECE523: Engineering Applications of Machine Learning and Data Analytics
Due 03/03/2017 @ 11:59PM (D2L)

Name:	____________________________
Signature:	____________________________
Date:	____________________________

Instructions: There are seven problems. Partial credit is given for answers that are partially correct. No credit is given for answers that are wrong or illegible. All work must be supported and code must be submitted for credit.

Theory:	__________
Practice:	__________
Total:	__________

Part A: Theory (10pts)

(10pts) Support Vector Machines

In class, we discussed that if our data is not linearly separable, then we need to modify our optimization problem to include slack variables. The formulation that was used is known as the ℓ_1-norm soft marging SVM. Now consider the formulation of the ℓ_2-norm soft margin SVM, which squares the slack variables within the sum. Notice that non-negativity of the slack variables has been removed.

$$\arg\min_{\mathbf{w},b,\xi} \frac{1}{2}\|\mathbf{w}\|^2 + \frac{C}{2}\sum_{i=1}^{n}\xi_i^2$$
$$\text{s.t. } y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \xi_i \qquad \forall i \in [n]$$

Derive the dual form expression along with any constraints. Work must be shown. *Hints*: Refer to the methodology that was used in class to derive the dual form. The solution is given by:

$$\arg\max_{\alpha} \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j \mathbf{x}_i^T\mathbf{x}_j - \frac{1}{2C}\sum_{i=1}^{n}\alpha_i^2$$
$$\text{s.t. } \alpha_i \geq 0 \quad \forall i \in [n] \quad \text{and} \quad \sum_{i=1}^{n}\alpha_i y_i = 0$$

Part A: Practice (40pts)

You are free to use functions already implemented in Matlab, Python or R with the exception of problem 1. I recommend using Python's Scikit-learn (`http://scikit-learn.org/stable/`) as is implements most of the methods we will be discussing in this course…as well as problems in this homework!

(20pts) Multi-Layer Perceptron

In class we discussed the derivation of the backpropagation algorithm for neural networks. In this problem, you will train a neural network on the classical MNIST data set. Train a Multi-Layer Perceptron (MLP) neural network on the classical MNIST data set. This is an opened implementation problem, but I expect that you implement the MLP with at least two different hidden layer sizes and use regularization.

- Report the classification error on the training and testing data each configuration of the neural network. For example, you should report the results in the form of a table

	Classification Error	
	training	**testing**
50HLN+no regularization	0.234	0.253
50HLN+L_2 regularization	0.192	0.203
250HLN+no regularization	0.134	0.153
250HLN+L_2 regularization	0.092	0.013

List all the parameters that you are using (i.e., number of learning rounds, regularization parameters, learning rate, etc.)

- I would suggest using Google's TensorFlow library to implement the MLP; however, you are free to use whatever library you'd like. If that is the case, here is a link to the data
  `http://yann.lecun.com/exdb/mnist/`
  `https://www.tensorflow.org/get_started/mnist/beginners`

(20pts) Support Vector Machines

Generate 2D Gaussian data with *at least* two components (i.e., you need at least one for the positive class and one for the negative class). Train and test a classifier of two disjoint data sets generated from the Gaussian components you described above and your selection of kernel parameters. Plot the training data and indicate Compute the classifier error, plot the training data, and plot the test data with the predicted class labels. Use at least two different kernels, and submit your results & code.

(5pts) Deep Learning (Bonus)

Re-do the problem on the MNIST data set, but with a deep neural network. Compare the accuracy of the deep network to the shallow networks.