Laboratory 4

Multivariable Logistic Regression Classification

Due Date: Beginning of Week 6 Lab

Concepts:

- Multivariable Logistic Regression Classification
- Regularization

Total Points: 100 points

Objectives:

Logistic regression is one of the most popular supervised classification algorithm. The one-vs-all multiclass logistic regression classifier is an extended version of logistic regression to solve multiclass classification problems. Students will implement the one-vs-all logistic regression classifier to distinguish ten digits in the MNIST hand written digit dataset and evaluate performance of the developed model on the test data that is not used during the training.

Files Needed:

- Lab4.mlx
- sigmoid.m
- displayData.m
- MNIST.mat

Assignment: Multiclass classification using regularized logistic regression

- **Multiclass classification** means a classification task with more than two classes. Logistic regression algorithm can also use to solve **multiclass classification problems.** In multiclass classification problems, a training set consisting of data points belonging to N different classes is given, and the goal is to construct a classifier which, given a new data sample, will correctly predict the class to which the new data belongs.

- The one-vs-all logistic regression classifier is implemented by training **multiple logistic regression classifiers,** one for each of the *C* classes in the training dataset. The algorithm turns the multiclass classification problem into **C binary classification problems** (i.e. where we predict only $y \in \{0, 1\}$) by taking values of one class and turn them into positive examples, and the rest of classes - into negatives**.**
- In Lab4, you will implement one-vs-all logistic regression to recognize hand-written digits on the MNIST dataset. Automated handwritten digit recognition is widely used today from

1. Data source & displayData function:modified from Andrew Ng

recognizing postal codes on mail envelopes to recognizing amounts written on bank checks. The MNIST dataset is a set of handwritten digits categorized 0-9 and is available at http://yann.lecun.com/exdb/mnist/.

1. Load the data from a file[1]

- The file "MNIST.mat " contains a training set of 4,000 handwritten digits and a test set of 1,000 handwritten digits. The images have a size of 20 by 20 grayscale images of digits. The 20 by 20 digit image is flattened into a 1 by 400 row vector. Each of these digit images becomes a single row in the data. The data is already normalized in the range of [0, 1] and separated into the training set and the test data.

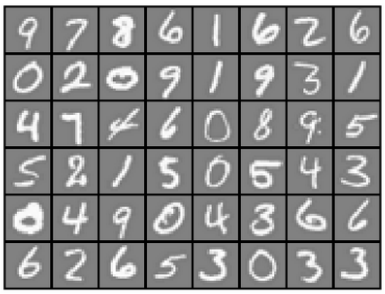
- Using the ***load( )*** function in Matlab, load the "MNIST.mat' file. The .mat format means that the data has been saved from the Matlab workspace. The matrix for the training data set is already named as 'xTrain' and the corresponding labels for the training set is named as 'yTrain', so you do not need to assign names to them. Similarly, the matrices for the test data set and the corresponding labels are names as 'xTest' and ' yTest', respectively.
- Note: There is no zero indexing in the Matlab. So the digits "1" to "9" are labeled as "1" to "9" and the digit "0 " is labeled as "10".

- Complete the lines of code in Section 1.

2. Visualize the data[1]

- Before starting on any prediction model, it is often useful to understand the data by plotting it.
- Display the randomly selected image samples from the training dataset. The **randperm**() function in Matlab returns a row vector containing a random permutation of the integers from 1 to n.
- Generate 48 numbers in the range of [1, 4000] using the **randperm**() function. Then, extract the image samples from the training data using the indexes generated by the **randperm**() function.

- Display the selected image samples by calling the provided function 'dispalyData()'. The plot is similar with the below figure. However the selected digit images will be different because the indexes are selected by the random number generator.



- Complete the lines of code in Section 2.

1. Data source & displayData function:modified from Andrew Ng

3. Add Bias term

- The "**bias term**" or "intercept parameter" allows us to move the linear model along the y-axis.
- Complete the lines of code in Section 3.

4. Training the one-vs-all classifier to find parameters using gradient descent

- To implement the one-vs-all classifier, multiple regularized logistic regression classifiers are trained one for each of $\boldsymbol{C}$ classes. Each classifier is trained independently using a " for"- loop from t=1 to $\boldsymbol{C}$. The training of the one-vs-all logistic regression with $\boldsymbol{C}$ classes is described as below.

  For each class t ,where t=1,..,C
  - Train a regularized logistic regression classifier $h_w^{(t)}(x)$ to predict the probability, p(y = $t|x$). The procedures for the regularized logistic regression are explained in the subsection 4.1.
  - To do this, all class t labels as positive samples (y=1) and all other classes as negative samples (y=0).

4.1 procedures for the logistic regression binary classifier

a) Sigmoid function

- The logistic regression hypothesis is defined as a sigmoid function of an input Z where Z is a liner model:

$$h_w^{(t)}(x) = \frac{1}{1+e^{-z}} \text{ , where } Z = \sum_{i=0}^{k} \boldsymbol{w_i x_i}$$

b) Log loss cost function

- The regularized cost function in logistic regression is the log loss function with the regularized term :

$$J^{(t)}(w) = -[\frac{1}{m}\sum_{i=1}^{m} \boldsymbol{y}^{(i)} \cdot \log\left(\boldsymbol{h_w}(\boldsymbol{x}^{(i)})\right) + (\boldsymbol{1} - \boldsymbol{y}^{(i)}) \cdot \log\left(\boldsymbol{1} - \boldsymbol{h_w}(\boldsymbol{x}^{(i)})\right)] + \frac{\lambda}{2m}\sum_{j=1}^{k} \boldsymbol{w_j^2}$$

- You will fit the classifier parameters, w with the training data using the gradient descent method. The objective of the training is to find the parameters, w that minimize the cost function:

$$\underset{w}{\text{Minimize}} \quad J^{(t)}(w)$$

- The gradient decent method will find the optimal weight , w using a " for"- loop from i=1 to iteration. The equations for the weight update is given:

$$\textbf{1) } \Delta \boldsymbol{w_0} = \frac{\boldsymbol{1}}{\boldsymbol{m}}\sum_{\boldsymbol{i=1}}^{\boldsymbol{m}}(\boldsymbol{h_w}(\boldsymbol{x^i}) - \boldsymbol{y^i})$$

1. Data source & displayData function:modified from Andrew Ng

$$w_0 = w_0 - \partial * \Delta w_0$$

2) for each w_j (j=1,…,k)

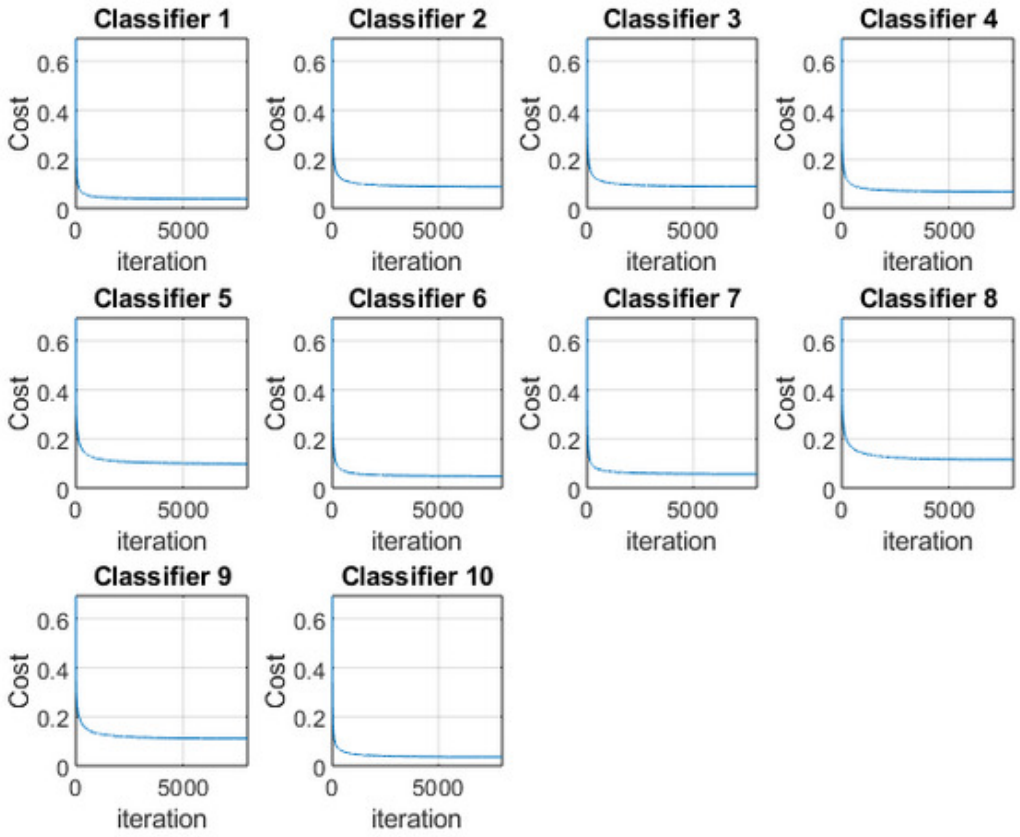
$$\Delta w_j = \frac{1}{m}\sum_{i=1}^{m}(h_w(x^i) - y^i)\, x_j^i + \frac{\lambda}{m} w_j$$

$$w_j = w_j - \partial * \Delta w_j$$

- To implement the regularization term, simply add the term, $\frac{\lambda}{m} w_j$ to the partial derivative of the cost function w.r.t w_j **except** w_0**.**

- At the end of the for loop iteration, you have the final log loss cost and the final parameters, w for the binary classification model for each class t.

5. **Plot the Log Loss cost convergence graph for each classifier.** Generate plots that shows how the cost changes over the iterations for each classifier t. You can use the ***subplot()*** function in the Matlab to display the multiple plots. The plots of the cost versus the iteration look like below:



1. Data source & displayData function:modified from Andrew Ng

- Complete the lines of code to implement **the Log Loss cost convergence graph** in Section 5.

6. The training accuracy of the one-vs-all classifier

- After the training is completed, the one-vs-all classifier can predict the digit contained in a given image. For the given input image, the probability that the input image belongs to each class is computed using the trained logistic regression classifiers. The one-vs-all classifier will pick the class for which the corresponding classifier outputs the highest probability.

- To predict the digits in given images,

  - Calculate the linear model $Z = \sum_{i=0}^{k} w_i x_i$
  - Calculate the sigmoid output of the input Z
  - Pick the class for which the corresponding classifier outputs the highest probability

- Write the lines of code for the calculation of the training accuracy in Section7.

Question 1: what is the accuracy of the developed model on the training data?

7. The one-vs-all classifier evaluation on the test data

- Add the intercept term (Bias term) in the input vector. A new input matrix is generated by concatenating the bias term vector and the new normalized input feature vector.
- Make the predictions on the new testing data using the one-vs-all classifier;

- **Calculate the accuracy of the developed classifier on the test data**

  - Calculate the linear model $Z = \sum_{i=0}^{k} w_i x_i$
  - Calculate the sigmoid output of the input Z
  - Pick the class for which the corresponding classifier outputs the highest probability

- Complete the lines of code to evaluate the multiclass classifier on the test data set
- **Question 2:** what is the accuracy of the multiclass classifier on the test data set?

8. Regularization

- Complex model (has lots of parameters) often prone to **overfitting.** Overfitting can be reduced by imposing a constraint on the overall magnitude of the parameters (weights),w. It introduces a penalty cost term for bringing in more features with the objective function.

- The regularized cost function in logistic regression is the log loss function with the regularized term :

$$J^{(t)}(w) = -[\frac{1}{m}\sum_{i=1}^{m} \boldsymbol{y}^{(i)} \cdot \log\left(\boldsymbol{h_w}(\boldsymbol{x}^{(i)})\right) + (\mathbf{1} - \boldsymbol{y}^{(i)}) \cdot \log\left(\mathbf{1} - \boldsymbol{h_w}(\boldsymbol{x}^{(i)})\right)] + \frac{\lambda}{2m}\sum_{j=1}^{k} \boldsymbol{w_j^2} \ .$$

- **The parameter $\boldsymbol{\lambda}$** controls the trade-off between training well and keeping the parameters small. When the value of λ is large, values of the weights will be small. When the value of λ is small, values of the weights will become large.
- **Train the one-vs-all classifier with different $\boldsymbol{\lambda}$ values**. Repeat Section 4, Section 6, and Section 7 with different λ values when λ =1, λ=5, and λ =10.
- **Question 3:** Find the accuracies of the classifier on the training data and the testing data with the different λ values when λ =1, λ=5, and λ =10.

You can work in groups of up to two people

What to Submit to Blackboard:

- **one** ZIP file that includes :
  1) **A report (15pts): contains answers for Question1 –Question3.**
  2) (85 pts): The live script file, "Lab4.mlx", in that all outputs are generated either inline or on the right side.

- **Pay attention to the zip file name convention:**
  Lab4_Student1 Lastname_Student2 Lastname.zip
  Ex) Lab4_Green_Smith.zip

1. Data source & displayData function:modified from Andrew Ng

Laboratory 4

Multivariable Logistic Regression Classification

Due Date: Beginning of Week 6 Lab

Concepts:

- Multivariable Logistic Regression Classification
- Regularization

Total Points: 100 points

Objectives:

Logistic regression is one of the most popular supervised classification algorithm. The one-vs-all multiclass logistic regression classifier is an extended version of logistic regression to solve multiclass classification problems. Students will implement the one-vs-all logistic regression classifier to distinguish ten digits in the MNIST hand written digit dataset and evaluate performance of the developed model on the test data that is not used during the training.

Files Needed:

- Lab4.mlx
- sigmoid.m
- displayData.m
- MNIST.mat

Assignment: Multiclass classification using regularized logistic regression

- **Multiclass classification** means a classification task with more than two classes. Logistic regression algorithm can also use to solve **multiclass classification problems.** In multiclass classification problems, a training set consisting of data points belonging to N different classes is given, and the goal is to construct a classifier which, given a new data sample, will correctly predict the class to which the new data belongs.

- The one-vs-all logistic regression classifier is implemented by training **multiple logistic regression classifiers,** one for each of the *C* classes in the training dataset. The algorithm turns the multiclass classification problem into **C binary classification problems** (i.e. where we predict only $y \in \{0, 1\}$) by taking values of one class and turn them into positive examples, and the rest of classes - into negatives**.**
- In Lab4, you will implement one-vs-all logistic regression to recognize hand-written digits on the MNIST dataset. Automated handwritten digit recognition is widely used today from

1. Data source & displayData function:modified from Andrew Ng

recognizing postal codes on mail envelopes to recognizing amounts written on bank checks. The MNIST dataset is a set of handwritten digits categorized 0-9 and is available at http://yann.lecun.com/exdb/mnist/.

1. Load the data from a file[1]

- The file "MNIST.mat " contains a training set of 4,000 handwritten digits and a test set of 1,000 handwritten digits. The images have a size of 20 by 20 grayscale images of digits. The 20 by 20 digit image is flattened into a 1 by 400 row vector. Each of these digit images becomes a single row in the data. The data is already normalized in the range of [0, 1] and separated into the training set and the test data.

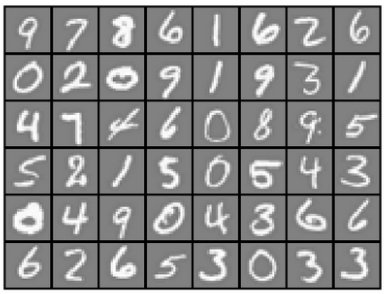
- Using the ***load( )*** function in Matlab, load the "MNIST.mat' file. The .mat format means that the data has been saved from the Matlab workspace. The matrix for the training data set is already named as 'xTrain' and the corresponding labels for the training set is named as 'yTrain', so you do not need to assign names to them. Similarly, the matrices for the test data set and the corresponding labels are names as 'xTest' and ' yTest', respectively.
- Note: There is no zero indexing in the Matlab. So the digits "1" to "9" are labeled as "1" to "9" and the digit "0 " is labeled as "10".

- Complete the lines of code in Section 1.

2. Visualize the data[1]

- Before starting on any prediction model, it is often useful to understand the data by plotting it.
- Display the randomly selected image samples from the training dataset. The **randperm**() function in Matlab returns a row vector containing a random permutation of the integers from 1 to n.
- Generate 48 numbers in the range of [1, 4000] using the **randperm**() function. Then, extract the image samples from the training data using the indexes generated by the **randperm**() function.

- Display the selected image samples by calling the provided function 'dispalyData()'. The plot is similar with the below figure. However the selected digit images will be different because the indexes are selected by the random number generator.



- Complete the lines of code in Section 2.

1. Data source & displayData function:modified from Andrew Ng

3. Add Bias term

- The "**bias term**" or "intercept parameter" allows us to move the linear model along the y-axis.
- Complete the lines of code in Section 3.

4. Training the one-vs-all classifier to find parameters using gradient descent

- To implement the one-vs-all classifier, multiple regularized logistic regression classifiers are trained one for each of $\boldsymbol{C}$ classes. Each classifier is trained independently using a " for"- loop from t=1 to $\boldsymbol{C}$. The training of the one-vs-all logistic regression with $\boldsymbol{C}$ classes is described as below.

  For each class t ,where t=1,..,C
  - Train a regularized logistic regression classifier $h_w^{(t)}(x)$ to predict the probability, p(y = $t|x$). The procedures for the regularized logistic regression are explained in the subsection 4.1.
  - To do this, all class t labels as positive samples (y=1) and all other classes as negative samples (y=0).

4.1 procedures for the logistic regression binary classifier

a) Sigmoid function

- The logistic regression hypothesis is defined as a sigmoid function of an input Z where Z is a liner model:

$$h_w^{(t)}(x) = \frac{1}{1+e^{-z}} \text{ , where } Z = \sum_{i=0}^{k} \boldsymbol{w_i x_i}$$

b) Log loss cost function

- The regularized cost function in logistic regression is the log loss function with the regularized term :

$$J^{(t)}(w) = -[\frac{1}{m}\sum_{i=1}^{m} \boldsymbol{y}^{(i)} \cdot \log\left(\boldsymbol{h_w}(\boldsymbol{x}^{(i)})\right) + (\boldsymbol{1} - \boldsymbol{y}^{(i)}) \cdot \log\left(\boldsymbol{1} - \boldsymbol{h_w}(\boldsymbol{x}^{(i)})\right)] + \frac{\lambda}{2m}\sum_{j=1}^{k} \boldsymbol{w_j^2}$$

- You will fit the classifier parameters, w with the training data using the gradient descent method. The objective of the training is to find the parameters, w that minimize the cost function:

$$\underset{w}{\text{Minimize}} \quad J^{(t)}(w)$$

- The gradient decent method will find the optimal weight , w using a " for"- loop from i=1 to iteration. The equations for the weight update is given:

$$\textbf{1) } \Delta \boldsymbol{w_0} = \frac{\boldsymbol{1}}{\boldsymbol{m}}\sum_{\boldsymbol{i=1}}^{\boldsymbol{m}}(\boldsymbol{h_w}(\boldsymbol{x^i}) - \boldsymbol{y^i})$$

1. Data source & displayData function:modified from Andrew Ng

$$w_0 = w_0 - \partial * \Delta w_0$$

2) for each w_j (j=1,…,k)

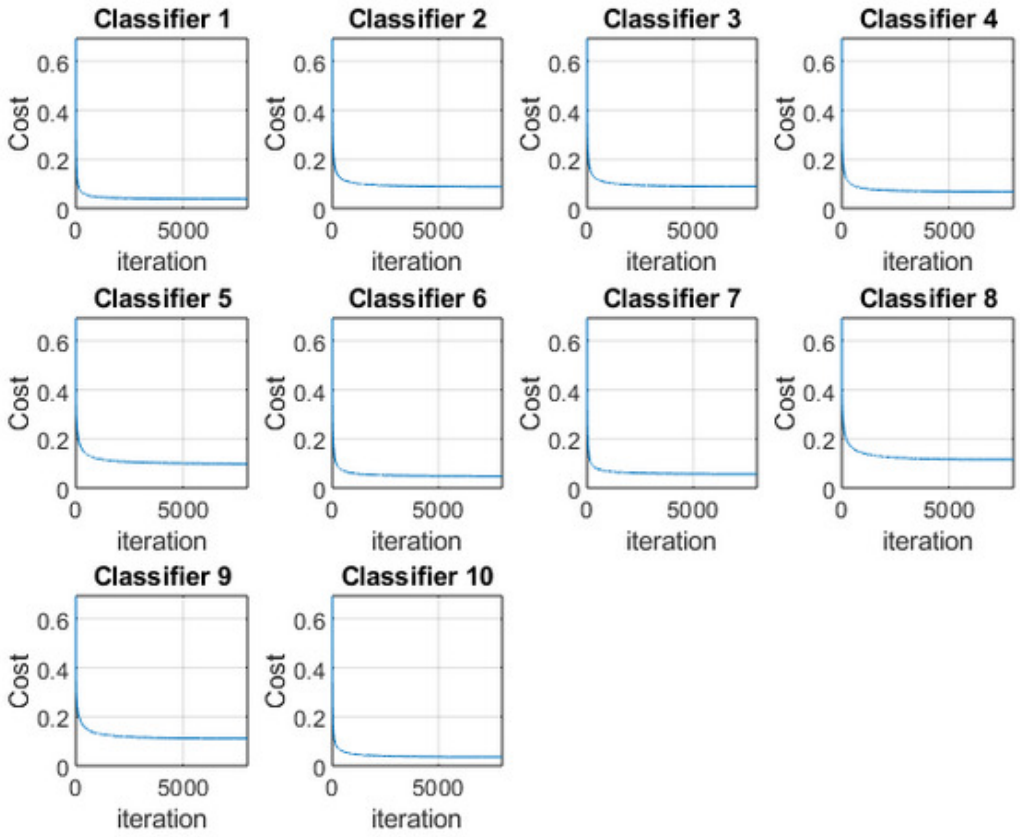
$$\Delta w_j = \frac{1}{m}\sum_{i=1}^{m}(h_w(x^i) - y^i)\, x_j^i + \frac{\lambda}{m} w_j$$

$$w_j = w_j - \partial * \Delta w_j$$

- To implement the regularization term, simply add the term, $\frac{\lambda}{m} w_j$ to the partial derivative of the cost function w.r.t w_j **except** w_0**.**

- At the end of the for loop iteration, you have the final log loss cost and the final parameters, w for the binary classification model for each class t.

5. **Plot the Log Loss cost convergence graph for each classifier.** Generate plots that shows how the cost changes over the iterations for each classifier t. You can use the ***subplot()*** function in the Matlab to display the multiple plots. The plots of the cost versus the iteration look like below:



1. Data source & displayData function:modified from Andrew Ng

- Complete the lines of code to implement **the Log Loss cost convergence graph** in Section 5.

6. The training accuracy of the one-vs-all classifier

- After the training is completed, the one-vs-all classifier can predict the digit contained in a given image. For the given input image, the probability that the input image belongs to each class is computed using the trained logistic regression classifiers. The one-vs-all classifier will pick the class for which the corresponding classifier outputs the highest probability.

- To predict the digits in given images,

  - Calculate the linear model $Z = \sum_{i=0}^{k} w_i x_i$
  - Calculate the sigmoid output of the input Z
  - Pick the class for which the corresponding classifier outputs the highest probability

- Write the lines of code for the calculation of the training accuracy in Section7.

Question 1: what is the accuracy of the developed model on the training data?

7. The one-vs-all classifier evaluation on the test data

- Add the intercept term (Bias term) in the input vector. A new input matrix is generated by concatenating the bias term vector and the new normalized input feature vector.
- Make the predictions on the new testing data using the one-vs-all classifier;

- **Calculate the accuracy of the developed classifier on the test data**

  - Calculate the linear model $Z = \sum_{i=0}^{k} w_i x_i$
  - Calculate the sigmoid output of the input Z
  - Pick the class for which the corresponding classifier outputs the highest probability

- Complete the lines of code to evaluate the multiclass classifier on the test data set
- **Question 2:** what is the accuracy of the multiclass classifier on the test data set?

8. Regularization

- Complex model (has lots of parameters) often prone to **overfitting.** Overfitting can be reduced by imposing a constraint on the overall magnitude of the parameters (weights),w. It introduces a penalty cost term for bringing in more features with the objective function.

- The regularized cost function in logistic regression is the log loss function with the regularized term :

$$J^{(t)}(w) = -[\frac{1}{m}\sum_{i=1}^{m} \boldsymbol{y}^{(i)} \cdot \log\left(\boldsymbol{h_w}(\boldsymbol{x}^{(i)})\right) + (\mathbf{1} - \boldsymbol{y}^{(i)}) \cdot \log\left(\mathbf{1} - \boldsymbol{h_w}(\boldsymbol{x}^{(i)})\right)] + \frac{\lambda}{2m}\sum_{j=1}^{k} \boldsymbol{w_j^2} \ .$$

- **The parameter $\boldsymbol{\lambda}$** controls the trade-off between training well and keeping the parameters small. When the value of λ is large, values of the weights will be small. When the value of λ is small, values of the weights will become large.
- **Train the one-vs-all classifier with different $\boldsymbol{\lambda}$ values**. Repeat Section 4, Section 6, and Section 7 with different λ values when λ =1, λ=5, and λ =10.
- **Question 3:** Find the accuracies of the classifier on the training data and the testing data with the different λ values when λ =1, λ=5, and λ =10.

You can work in groups of up to two people

What to Submit to Blackboard:

- **one** ZIP file that includes :
  1) **A report (15pts): contains answers for Question1 –Question3.**
  2) (85 pts): The live script file, "Lab4.mlx", in that all outputs are generated either inline or on the right side.

- **Pay attention to the zip file name convention:**
  Lab4_Student1 Lastname_Student2 Lastname.zip
  Ex) Lab4_Green_Smith.zip

1. Data source & displayData function:modified from Andrew Ng

Laboratory 4

Multivariable Logistic Regression Classification

Due Date: Beginning of Week 6 Lab

Concepts:

- Multivariable Logistic Regression Classification
- Regularization

Total Points: 100 points

Objectives:

Logistic regression is one of the most popular supervised classification algorithm. The one-vs-all multiclass logistic regression classifier is an extended version of logistic regression to solve multiclass classification problems. Students will implement the one-vs-all logistic regression classifier to distinguish ten digits in the MNIST hand written digit dataset and evaluate performance of the developed model on the test data that is not used during the training.

Files Needed:

- Lab4.mlx
- sigmoid.m
- displayData.m
- MNIST.mat

Assignment: Multiclass classification using regularized logistic regression

- **Multiclass classification** means a classification task with more than two classes. Logistic regression algorithm can also use to solve **multiclass classification problems.** In multiclass classification problems, a training set consisting of data points belonging to N different classes is given, and the goal is to construct a classifier which, given a new data sample, will correctly predict the class to which the new data belongs.

- The one-vs-all logistic regression classifier is implemented by training **multiple logistic regression classifiers,** one for each of the *C* classes in the training dataset. The algorithm turns the multiclass classification problem into **C binary classification problems** (i.e. where we predict only $y \in \{0, 1\}$) by taking values of one class and turn them into positive examples, and the rest of classes - into negatives**.**
- In Lab4, you will implement one-vs-all logistic regression to recognize hand-written digits on the MNIST dataset. Automated handwritten digit recognition is widely used today from

1. Data source & displayData function:modified from Andrew Ng

recognizing postal codes on mail envelopes to recognizing amounts written on bank checks. The MNIST dataset is a set of handwritten digits categorized 0-9 and is available at http://yann.lecun.com/exdb/mnist/.

1. Load the data from a file[1]

- The file "MNIST.mat " contains a training set of 4,000 handwritten digits and a test set of 1,000 handwritten digits. The images have a size of 20 by 20 grayscale images of digits. The 20 by 20 digit image is flattened into a 1 by 400 row vector. Each of these digit images becomes a single row in the data. The data is already normalized in the range of [0, 1] and separated into the training set and the test data.

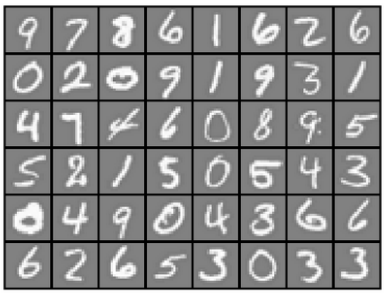
- Using the ***load( )*** function in Matlab, load the "MNIST.mat' file. The .mat format means that the data has been saved from the Matlab workspace. The matrix for the training data set is already named as 'xTrain' and the corresponding labels for the training set is named as 'yTrain', so you do not need to assign names to them. Similarly, the matrices for the test data set and the corresponding labels are names as 'xTest' and ' yTest', respectively.
- Note: There is no zero indexing in the Matlab. So the digits "1" to "9" are labeled as "1" to "9" and the digit "0 " is labeled as "10".

- Complete the lines of code in Section 1.

2. Visualize the data[1]

- Before starting on any prediction model, it is often useful to understand the data by plotting it.
- Display the randomly selected image samples from the training dataset. The **randperm**() function in Matlab returns a row vector containing a random permutation of the integers from 1 to n.
- Generate 48 numbers in the range of [1, 4000] using the **randperm**() function. Then, extract the image samples from the training data using the indexes generated by the **randperm**() function.

- Display the selected image samples by calling the provided function 'dispalyData()'. The plot is similar with the below figure. However the selected digit images will be different because the indexes are selected by the random number generator.



- Complete the lines of code in Section 2.

1. Data source & displayData function:modified from Andrew Ng

3. Add Bias term

- The "**bias term**" or "intercept parameter" allows us to move the linear model along the y-axis.
- Complete the lines of code in Section 3.

4. Training the one-vs-all classifier to find parameters using gradient descent

- To implement the one-vs-all classifier, multiple regularized logistic regression classifiers are trained one for each of $\boldsymbol{C}$ classes. Each classifier is trained independently using a " for"- loop from t=1 to $\boldsymbol{C}$. The training of the one-vs-all logistic regression with $\boldsymbol{C}$ classes is described as below.

  For each class t ,where t=1,..,C
  - Train a regularized logistic regression classifier $h_w^{(t)}(x)$ to predict the probability, p(y = $t|x$). The procedures for the regularized logistic regression are explained in the subsection 4.1.
  - To do this, all class t labels as positive samples (y=1) and all other classes as negative samples (y=0).

4.1 procedures for the logistic regression binary classifier

a) Sigmoid function

- The logistic regression hypothesis is defined as a sigmoid function of an input Z where Z is a liner model:

$$h_w^{(t)}(x) = \frac{1}{1+e^{-z}} \text{ , where } Z = \sum_{i=0}^{k} \boldsymbol{w_i x_i}$$

b) Log loss cost function

- The regularized cost function in logistic regression is the log loss function with the regularized term :

$$J^{(t)}(w) = -[\frac{1}{m}\sum_{i=1}^{m} \boldsymbol{y}^{(i)} \cdot \log\left(\boldsymbol{h_w}(\boldsymbol{x}^{(i)})\right) + (\boldsymbol{1} - \boldsymbol{y}^{(i)}) \cdot \log\left(\boldsymbol{1} - \boldsymbol{h_w}(\boldsymbol{x}^{(i)})\right)] + \frac{\lambda}{2m}\sum_{j=1}^{k} \boldsymbol{w_j^2}$$

- You will fit the classifier parameters, w with the training data using the gradient descent method. The objective of the training is to find the parameters, w that minimize the cost function:

$$\underset{w}{\text{Minimize}} \quad J^{(t)}(w)$$

- The gradient decent method will find the optimal weight , w using a " for"- loop from i=1 to iteration. The equations for the weight update is given:

$$\textbf{1) } \Delta \boldsymbol{w_0} = \frac{\boldsymbol{1}}{\boldsymbol{m}}\sum_{\boldsymbol{i=1}}^{\boldsymbol{m}}(\boldsymbol{h_w}(\boldsymbol{x^i}) - \boldsymbol{y^i})$$

1. Data source & displayData function:modified from Andrew Ng

$$w_0 = w_0 - \partial * \Delta w_0$$

2) for each w_j (j=1,…,k)

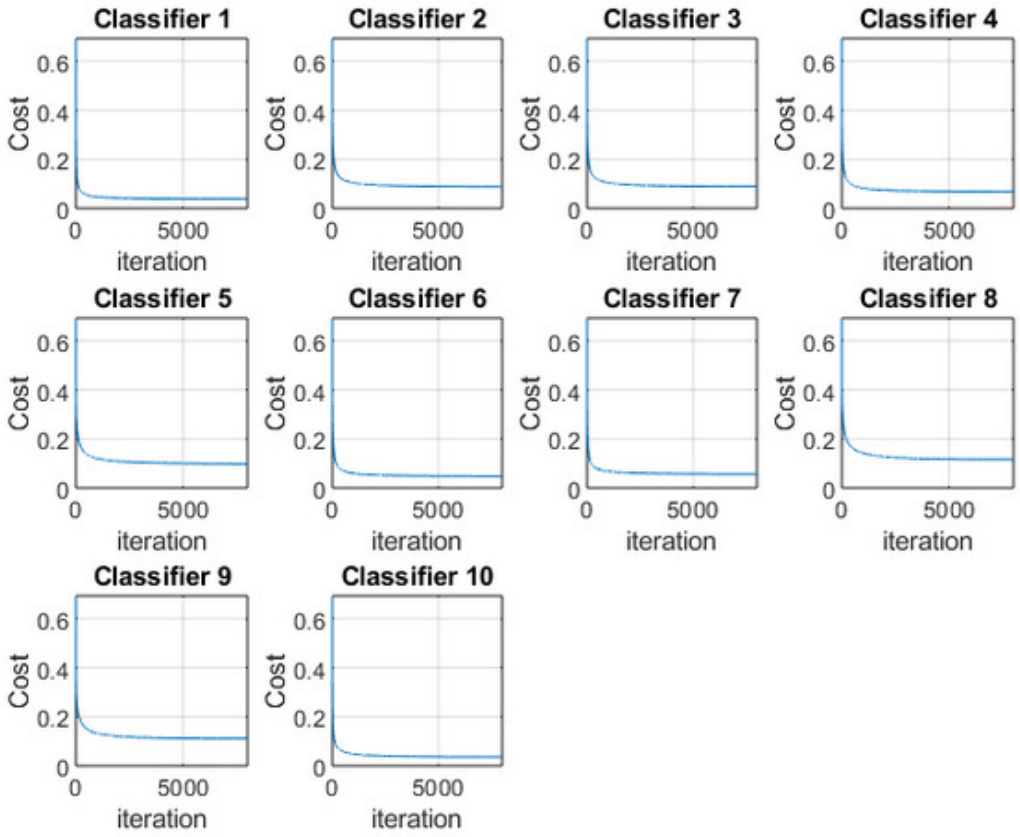
$$\Delta w_j = \frac{1}{m}\sum_{i=1}^{m}(h_w(x^i) - y^i)\, x_j^i + \frac{\lambda}{m} w_j$$

$$w_j = w_j - \partial * \Delta w_j$$

- To implement the regularization term, simply add the term, $\frac{\lambda}{m} w_j$ to the partial derivative of the cost function w.r.t w_j **except** w_0**.**

- At the end of the for loop iteration, you have the final log loss cost and the final parameters, w for the binary classification model for each class t.

5. **Plot the Log Loss cost convergence graph for each classifier.** Generate plots that shows how the cost changes over the iterations for each classifier t. You can use the ***subplot()*** function in the Matlab to display the multiple plots. The plots of the cost versus the iteration look like below:



1. Data source & displayData function:modified from Andrew Ng

- Complete the lines of code to implement **the Log Loss cost convergence graph** in Section 5.

6. The training accuracy of the one-vs-all classifier

- After the training is completed, the one-vs-all classifier can predict the digit contained in a given image. For the given input image, the probability that the input image belongs to each class is computed using the trained logistic regression classifiers. The one-vs-all classifier will pick the class for which the corresponding classifier outputs the highest probability.

- To predict the digits in given images,

  - Calculate the linear model $Z = \sum_{i=0}^{k} w_i x_i$
  - Calculate the sigmoid output of the input Z
  - Pick the class for which the corresponding classifier outputs the highest probability

- Write the lines of code for the calculation of the training accuracy in Section7.

Question 1: what is the accuracy of the developed model on the training data?

7. The one-vs-all classifier evaluation on the test data

- Add the intercept term (Bias term) in the input vector. A new input matrix is generated by concatenating the bias term vector and the new normalized input feature vector.
- Make the predictions on the new testing data using the one-vs-all classifier;

- **Calculate the accuracy of the developed classifier on the test data**

  - Calculate the linear model $Z = \sum_{i=0}^{k} w_i x_i$
  - Calculate the sigmoid output of the input Z
  - Pick the class for which the corresponding classifier outputs the highest probability

- Complete the lines of code to evaluate the multiclass classifier on the test data set
- **Question 2:** what is the accuracy of the multiclass classifier on the test data set?

8. Regularization

- Complex model (has lots of parameters) often prone to **overfitting.** Overfitting can be reduced by imposing a constraint on the overall magnitude of the parameters (weights),w. It introduces a penalty cost term for bringing in more features with the objective function.

- The regularized cost function in logistic regression is the log loss function with the regularized term :

$$J^{(t)}(w) = -[\frac{1}{m}\sum_{i=1}^{m} \boldsymbol{y}^{(i)} \cdot \log\left(\boldsymbol{h_w}(\boldsymbol{x}^{(i)})\right) + (\mathbf{1} - \boldsymbol{y}^{(i)}) \cdot \log\left(\mathbf{1} - \boldsymbol{h_w}(\boldsymbol{x}^{(i)})\right)] + \frac{\lambda}{2m}\sum_{j=1}^{k} \boldsymbol{w_j^2} \ .$$

- **The parameter $\boldsymbol{\lambda}$** controls the trade-off between training well and keeping the parameters small. When the value of λ is large, values of the weights will be small. When the value of λ is small, values of the weights will become large.
- **Train the one-vs-all classifier with different $\boldsymbol{\lambda}$ values**. Repeat Section 4, Section 6, and Section 7 with different λ values when λ =1, λ=5, and λ =10.
- **Question 3:** Find the accuracies of the classifier on the training data and the testing data with the different λ values when λ =1, λ=5, and λ =10.

You can work in groups of up to two people

What to Submit to Blackboard:

- **one** ZIP file that includes :
  1) **A report (15pts): contains answers for Question1 –Question3.**
  2) (85 pts): The live script file, "Lab4.mlx", in that all outputs are generated either inline or on the right side.

- **Pay attention to the zip file name convention:**
  Lab4_Student1 Lastname_Student2 Lastname.zip
  Ex) Lab4_Green_Smith.zip

1. Data source & displayData function:modified from Andrew Ng

Laboratory 4

Multivariable Logistic Regression Classification

Due Date: Beginning of Week 6 Lab

Concepts:

- Multivariable Logistic Regression Classification
- Regularization

Total Points: 100 points

Objectives:

Logistic regression is one of the most popular supervised classification algorithm. The one-vs-all multiclass logistic regression classifier is an extended version of logistic regression to solve multiclass classification problems. Students will implement the one-vs-all logistic regression classifier to distinguish ten digits in the MNIST hand written digit dataset and evaluate performance of the developed model on the test data that is not used during the training.

Files Needed:

- Lab4.mlx
- sigmoid.m
- displayData.m
- MNIST.mat

Assignment: Multiclass classification using regularized logistic regression

- **Multiclass classification** means a classification task with more than two classes. Logistic regression algorithm can also use to solve **multiclass classification problems.** In multiclass classification problems, a training set consisting of data points belonging to N different classes is given, and the goal is to construct a classifier which, given a new data sample, will correctly predict the class to which the new data belongs.

- The one-vs-all logistic regression classifier is implemented by training **multiple logistic regression classifiers,** one for each of the *C* classes in the training dataset. The algorithm turns the multiclass classification problem into **C binary classification problems** (i.e. where we predict only $y \in \{0, 1\}$) by taking values of one class and turn them into positive examples, and the rest of classes - into negatives**.**
- In Lab4, you will implement one-vs-all logistic regression to recognize hand-written digits on the MNIST dataset. Automated handwritten digit recognition is widely used today from

1. Data source & displayData function:modified from Andrew Ng

recognizing postal codes on mail envelopes to recognizing amounts written on bank checks. The MNIST dataset is a set of handwritten digits categorized 0-9 and is available at http://yann.lecun.com/exdb/mnist/.

1. Load the data from a file[1]

- The file "MNIST.mat " contains a training set of 4,000 handwritten digits and a test set of 1,000 handwritten digits. The images have a size of 20 by 20 grayscale images of digits. The 20 by 20 digit image is flattened into a 1 by 400 row vector. Each of these digit images becomes a single row in the data. The data is already normalized in the range of [0, 1] and separated into the training set and the test data.

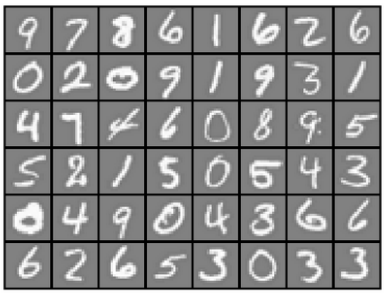
- Using the ***load( )*** function in Matlab, load the "MNIST.mat' file. The .mat format means that the data has been saved from the Matlab workspace. The matrix for the training data set is already named as 'xTrain' and the corresponding labels for the training set is named as 'yTrain', so you do not need to assign names to them. Similarly, the matrices for the test data set and the corresponding labels are names as 'xTest' and ' yTest', respectively.
- Note: There is no zero indexing in the Matlab. So the digits "1" to "9" are labeled as "1" to "9" and the digit "0 " is labeled as "10".

- Complete the lines of code in Section 1.

2. Visualize the data[1]

- Before starting on any prediction model, it is often useful to understand the data by plotting it.
- Display the randomly selected image samples from the training dataset. The **randperm**() function in Matlab returns a row vector containing a random permutation of the integers from 1 to n.
- Generate 48 numbers in the range of [1, 4000] using the **randperm**() function. Then, extract the image samples from the training data using the indexes generated by the **randperm**() function.

- Display the selected image samples by calling the provided function 'dispalyData()'. The plot is similar with the below figure. However the selected digit images will be different because the indexes are selected by the random number generator.



- Complete the lines of code in Section 2.

1. Data source & displayData function:modified from Andrew Ng

3. Add Bias term

- The "**bias term**" or "intercept parameter" allows us to move the linear model along the y-axis.
- Complete the lines of code in Section 3.

4. Training the one-vs-all classifier to find parameters using gradient descent

- To implement the one-vs-all classifier, multiple regularized logistic regression classifiers are trained one for each of $\boldsymbol{C}$ classes. Each classifier is trained independently using a " for"- loop from t=1 to $\boldsymbol{C}$. The training of the one-vs-all logistic regression with $\boldsymbol{C}$ classes is described as below.

  For each class t ,where t=1,..,C
  - Train a regularized logistic regression classifier $h_w^{(t)}(x)$ to predict the probability, p(y = $t|x$). The procedures for the regularized logistic regression are explained in the subsection 4.1.
  - To do this, all class t labels as positive samples (y=1) and all other classes as negative samples (y=0).

4.1 procedures for the logistic regression binary classifier

a) Sigmoid function

- The logistic regression hypothesis is defined as a sigmoid function of an input Z where Z is a liner model:

$$h_w^{(t)}(x) = \frac{1}{1+e^{-z}} \text{ , where } Z = \sum_{i=0}^{k} \boldsymbol{w_i x_i}$$

b) Log loss cost function

- The regularized cost function in logistic regression is the log loss function with the regularized term :

$$J^{(t)}(w) = -[\frac{1}{m}\sum_{i=1}^{m} \boldsymbol{y}^{(i)} \cdot \log\left(\boldsymbol{h_w}(\boldsymbol{x}^{(i)})\right) + (\boldsymbol{1} - \boldsymbol{y}^{(i)}) \cdot \log\left(\boldsymbol{1} - \boldsymbol{h_w}(\boldsymbol{x}^{(i)})\right)] + \frac{\lambda}{2m}\sum_{j=1}^{k} \boldsymbol{w_j^2}$$

- You will fit the classifier parameters, w with the training data using the gradient descent method. The objective of the training is to find the parameters, w that minimize the cost function:

$$\underset{w}{\text{Minimize}} \quad J^{(t)}(w)$$

- The gradient decent method will find the optimal weight , w using a " for"- loop from i=1 to iteration. The equations for the weight update is given:

$$\textbf{1) } \Delta \boldsymbol{w_0} = \frac{\boldsymbol{1}}{\boldsymbol{m}}\sum_{\boldsymbol{i=1}}^{\boldsymbol{m}}(\boldsymbol{h_w}(\boldsymbol{x^i}) - \boldsymbol{y^i})$$

1. Data source & displayData function:modified from Andrew Ng

$$w_0 = w_0 - \partial * \Delta w_0$$

2) for each w_j (j=1,…,k)

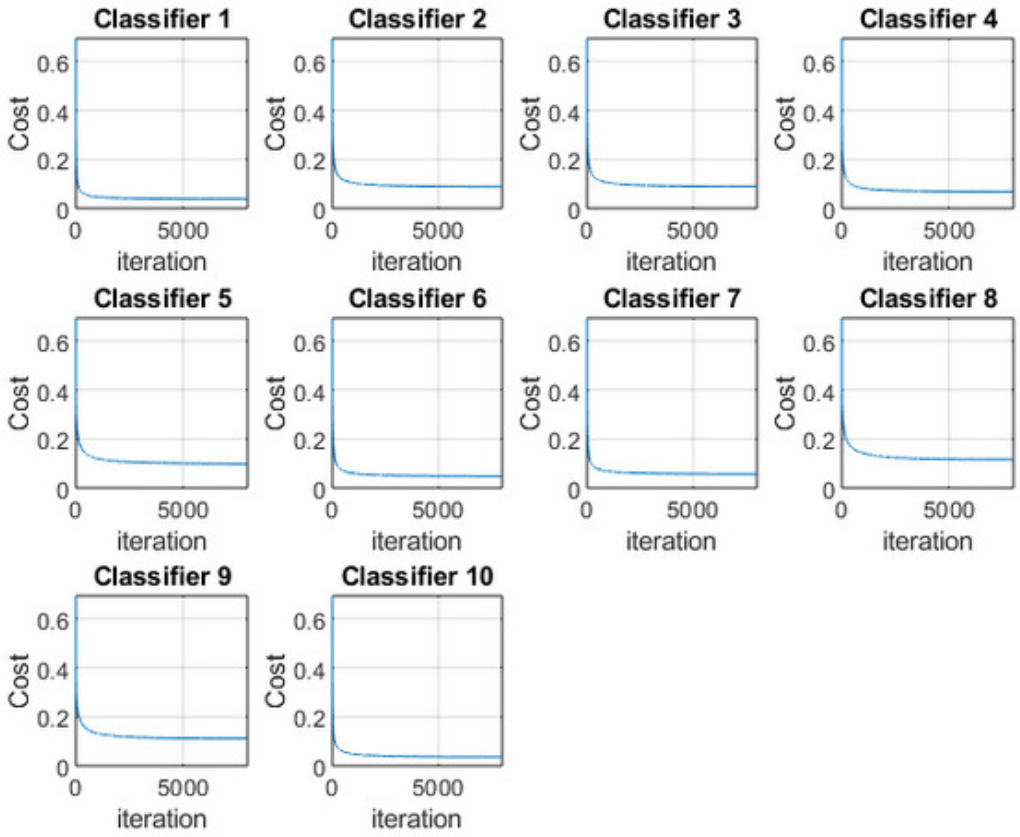
$$\Delta w_j = \frac{1}{m}\sum_{i=1}^{m}(h_w(x^i) - y^i)\, x_j^i + \frac{\lambda}{m} w_j$$

$$w_j = w_j - \partial * \Delta w_j$$

- To implement the regularization term, simply add the term, $\frac{\lambda}{m} w_j$ to the partial derivative of the cost function w.r.t w_j **except** w_0**.**

- At the end of the for loop iteration, you have the final log loss cost and the final parameters, w for the binary classification model for each class t.

5. **Plot the Log Loss cost convergence graph for each classifier.** Generate plots that shows how the cost changes over the iterations for each classifier t. You can use the ***subplot()*** function in the Matlab to display the multiple plots. The plots of the cost versus the iteration look like below:



1. Data source & displayData function:modified from Andrew Ng

- Complete the lines of code to implement **the Log Loss cost convergence graph** in Section 5.

6. The training accuracy of the one-vs-all classifier

- After the training is completed, the one-vs-all classifier can predict the digit contained in a given image. For the given input image, the probability that the input image belongs to each class is computed using the trained logistic regression classifiers. The one-vs-all classifier will pick the class for which the corresponding classifier outputs the highest probability.

- To predict the digits in given images,

  - Calculate the linear model $Z = \sum_{i=0}^{k} w_i x_i$
  - Calculate the sigmoid output of the input Z
  - Pick the class for which the corresponding classifier outputs the highest probability

- Write the lines of code for the calculation of the training accuracy in Section7.

Question 1: what is the accuracy of the developed model on the training data?

7. The one-vs-all classifier evaluation on the test data

- Add the intercept term (Bias term) in the input vector. A new input matrix is generated by concatenating the bias term vector and the new normalized input feature vector.
- Make the predictions on the new testing data using the one-vs-all classifier;

- **Calculate the accuracy of the developed classifier on the test data**

  - Calculate the linear model $Z = \sum_{i=0}^{k} w_i x_i$
  - Calculate the sigmoid output of the input Z
  - Pick the class for which the corresponding classifier outputs the highest probability

- Complete the lines of code to evaluate the multiclass classifier on the test data set
- **Question 2:** what is the accuracy of the multiclass classifier on the test data set?

8. Regularization

- Complex model (has lots of parameters) often prone to **overfitting.** Overfitting can be reduced by imposing a constraint on the overall magnitude of the parameters (weights),w. It introduces a penalty cost term for bringing in more features with the objective function.

- The regularized cost function in logistic regression is the log loss function with the regularized term :

$$J^{(t)}(w) = -[\frac{1}{m}\sum_{i=1}^{m} \boldsymbol{y}^{(i)} \cdot \log\left(\boldsymbol{h_w}(\boldsymbol{x}^{(i)})\right) + (\mathbf{1} - \boldsymbol{y}^{(i)}) \cdot \log\left(\mathbf{1} - \boldsymbol{h_w}(\boldsymbol{x}^{(i)})\right)] + \frac{\lambda}{2m}\sum_{j=1}^{k} \boldsymbol{w_j^2} \ .$$

- **The parameter $\boldsymbol{\lambda}$** controls the trade-off between training well and keeping the parameters small. When the value of λ is large, values of the weights will be small. When the value of λ is small, values of the weights will become large.
- **Train the one-vs-all classifier with different $\boldsymbol{\lambda}$ values**. Repeat Section 4, Section 6, and Section 7 with different λ values when λ =1, λ=5, and λ =10.
- **Question 3:** Find the accuracies of the classifier on the training data and the testing data with the different λ values when λ =1, λ=5, and λ =10.

You can work in groups of up to two people

What to Submit to Blackboard:

- **one** ZIP file that includes :
  1) **A report (15pts): contains answers for Question1 –Question3.**
  2) (85 pts): The live script file, "Lab4.mlx", in that all outputs are generated either inline or on the right side.

- **Pay attention to the zip file name convention:**
  Lab4_Student1 Lastname_Student2 Lastname.zip
  Ex) Lab4_Green_Smith.zip

1. Data source & displayData function:modified from Andrew Ng

Laboratory 4

Multivariable Logistic Regression Classification

Due Date: Beginning of Week 6 Lab

Concepts:

- Multivariable Logistic Regression Classification
- Regularization

Total Points: 100 points

Objectives:

Logistic regression is one of the most popular supervised classification algorithm. The one-vs-all multiclass logistic regression classifier is an extended version of logistic regression to solve multiclass classification problems. Students will implement the one-vs-all logistic regression classifier to distinguish ten digits in the MNIST hand written digit dataset and evaluate performance of the developed model on the test data that is not used during the training.

Files Needed:

- Lab4.mlx
- sigmoid.m
- displayData.m
- MNIST.mat

Assignment: Multiclass classification using regularized logistic regression

- **Multiclass classification** means a classification task with more than two classes. Logistic regression algorithm can also use to solve **multiclass classification problems.** In multiclass classification problems, a training set consisting of data points belonging to N different classes is given, and the goal is to construct a classifier which, given a new data sample, will correctly predict the class to which the new data belongs.

- The one-vs-all logistic regression classifier is implemented by training **multiple logistic regression classifiers,** one for each of the *C* classes in the training dataset. The algorithm turns the multiclass classification problem into **C binary classification problems** (i.e. where we predict only $y \in \{0, 1\}$) by taking values of one class and turn them into positive examples, and the rest of classes - into negatives**.**
- In Lab4, you will implement one-vs-all logistic regression to recognize hand-written digits on the MNIST dataset. Automated handwritten digit recognition is widely used today from

1. Data source & displayData function:modified from Andrew Ng

recognizing postal codes on mail envelopes to recognizing amounts written on bank checks. The MNIST dataset is a set of handwritten digits categorized 0-9 and is available at http://yann.lecun.com/exdb/mnist/.

1. Load the data from a file[1]

- The file "MNIST.mat " contains a training set of 4,000 handwritten digits and a test set of 1,000 handwritten digits. The images have a size of 20 by 20 grayscale images of digits. The 20 by 20 digit image is flattened into a 1 by 400 row vector. Each of these digit images becomes a single row in the data. The data is already normalized in the range of [0, 1] and separated into the training set and the test data.

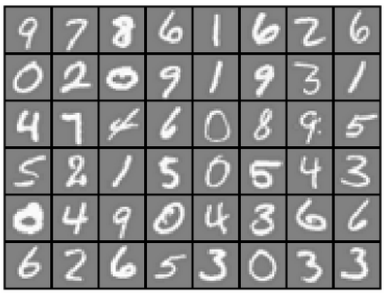
- Using the ***load( )*** function in Matlab, load the "MNIST.mat' file. The .mat format means that the data has been saved from the Matlab workspace. The matrix for the training data set is already named as 'xTrain' and the corresponding labels for the training set is named as 'yTrain', so you do not need to assign names to them. Similarly, the matrices for the test data set and the corresponding labels are names as 'xTest' and ' yTest', respectively.
- Note: There is no zero indexing in the Matlab. So the digits "1" to "9" are labeled as "1" to "9" and the digit "0 " is labeled as "10".

- Complete the lines of code in Section 1.

2. Visualize the data[1]

- Before starting on any prediction model, it is often useful to understand the data by plotting it.
- Display the randomly selected image samples from the training dataset. The **randperm**() function in Matlab returns a row vector containing a random permutation of the integers from 1 to n.
- Generate 48 numbers in the range of [1, 4000] using the **randperm**() function. Then, extract the image samples from the training data using the indexes generated by the **randperm**() function.

- Display the selected image samples by calling the provided function 'dispalyData()'. The plot is similar with the below figure. However the selected digit images will be different because the indexes are selected by the random number generator.



- Complete the lines of code in Section 2.

1. Data source & displayData function:modified from Andrew Ng

3. Add Bias term

- The "**bias term**" or "intercept parameter" allows us to move the linear model along the y-axis.
- Complete the lines of code in Section 3.

4. Training the one-vs-all classifier to find parameters using gradient descent

- To implement the one-vs-all classifier, multiple regularized logistic regression classifiers are trained one for each of $\boldsymbol{C}$ classes. Each classifier is trained independently using a " for"- loop from t=1 to $\boldsymbol{C}$. The training of the one-vs-all logistic regression with $\boldsymbol{C}$ classes is described as below.

  For each class t ,where t=1,..,C
  - Train a regularized logistic regression classifier $h_w^{(t)}(x)$ to predict the probability, p(y = $t|x$). The procedures for the regularized logistic regression are explained in the subsection 4.1.
  - To do this, all class t labels as positive samples (y=1) and all other classes as negative samples (y=0).

4.1 procedures for the logistic regression binary classifier

a) Sigmoid function

- The logistic regression hypothesis is defined as a sigmoid function of an input Z where Z is a liner model:

$$h_w^{(t)}(x) = \frac{1}{1+e^{-z}} \text{ , where } Z = \sum_{i=0}^{k} \boldsymbol{w_i x_i}$$

b) Log loss cost function

- The regularized cost function in logistic regression is the log loss function with the regularized term :

$$J^{(t)}(w) = -[\frac{1}{m}\sum_{i=1}^{m} \boldsymbol{y}^{(i)} \cdot \log\left(\boldsymbol{h_w}(\boldsymbol{x}^{(i)})\right) + (\boldsymbol{1} - \boldsymbol{y}^{(i)}) \cdot \log\left(\boldsymbol{1} - \boldsymbol{h_w}(\boldsymbol{x}^{(i)})\right)] + \frac{\lambda}{2m}\sum_{j=1}^{k} \boldsymbol{w_j^2}$$

- You will fit the classifier parameters, w with the training data using the gradient descent method. The objective of the training is to find the parameters, w that minimize the cost function:

$$\underset{w}{\text{Minimize}} \quad J^{(t)}(w)$$

- The gradient decent method will find the optimal weight , w using a " for"- loop from i=1 to iteration. The equations for the weight update is given:

$$\textbf{1) } \Delta \boldsymbol{w_0} = \frac{\boldsymbol{1}}{\boldsymbol{m}}\sum_{\boldsymbol{i=1}}^{\boldsymbol{m}}(\boldsymbol{h_w}(\boldsymbol{x^i}) - \boldsymbol{y^i})$$

1. Data source & displayData function:modified from Andrew Ng

$$w_0 = w_0 - \partial * \Delta w_0$$

2) for each w_j (j=1,…,k)

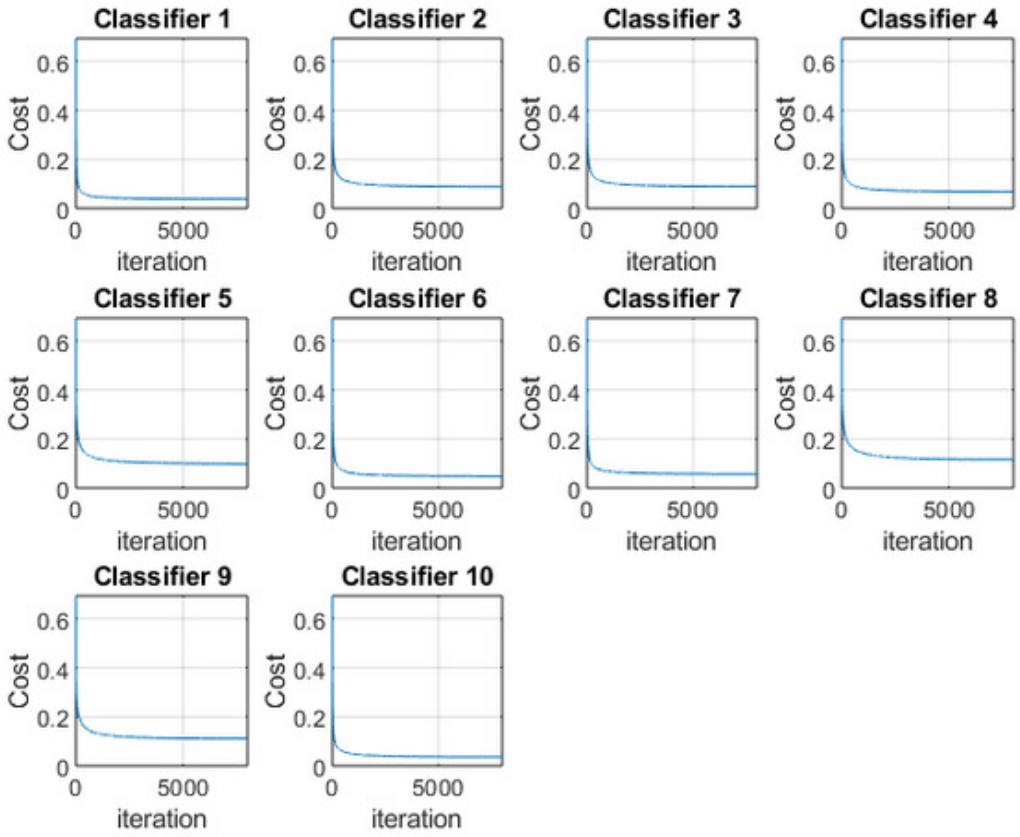
$$\Delta w_j = \frac{1}{m}\sum_{i=1}^{m}(h_w(x^i) - y^i)\, x_j^i + \frac{\lambda}{m} w_j$$

$$w_j = w_j - \partial * \Delta w_j$$

- To implement the regularization term, simply add the term, $\frac{\lambda}{m} w_j$ to the partial derivative of the cost function w.r.t w_j **except** w_0**.**

- At the end of the for loop iteration, you have the final log loss cost and the final parameters, w for the binary classification model for each class t.

5. **Plot the Log Loss cost convergence graph for each classifier.** Generate plots that shows how the cost changes over the iterations for each classifier t. You can use the ***subplot()*** function in the Matlab to display the multiple plots. The plots of the cost versus the iteration look like below:



1. Data source & displayData function:modified from Andrew Ng

- Complete the lines of code to implement **the Log Loss cost convergence graph** in Section 5.

6. The training accuracy of the one-vs-all classifier

- After the training is completed, the one-vs-all classifier can predict the digit contained in a given image. For the given input image, the probability that the input image belongs to each class is computed using the trained logistic regression classifiers. The one-vs-all classifier will pick the class for which the corresponding classifier outputs the highest probability.

- To predict the digits in given images,

  - Calculate the linear model $Z = \sum_{i=0}^{k} w_i x_i$
  - Calculate the sigmoid output of the input Z
  - Pick the class for which the corresponding classifier outputs the highest probability

- Write the lines of code for the calculation of the training accuracy in Section7.

Question 1: what is the accuracy of the developed model on the training data?

7. The one-vs-all classifier evaluation on the test data

- Add the intercept term (Bias term) in the input vector. A new input matrix is generated by concatenating the bias term vector and the new normalized input feature vector.
- Make the predictions on the new testing data using the one-vs-all classifier;

- **Calculate the accuracy of the developed classifier on the test data**

  - Calculate the linear model $Z = \sum_{i=0}^{k} w_i x_i$
  - Calculate the sigmoid output of the input Z
  - Pick the class for which the corresponding classifier outputs the highest probability

- Complete the lines of code to evaluate the multiclass classifier on the test data set
- **Question 2:** what is the accuracy of the multiclass classifier on the test data set?

8. Regularization

- Complex model (has lots of parameters) often prone to **overfitting.** Overfitting can be reduced by imposing a constraint on the overall magnitude of the parameters (weights),w. It introduces a penalty cost term for bringing in more features with the objective function.

- The regularized cost function in logistic regression is the log loss function with the regularized term :

$$J^{(t)}(w) = -[\frac{1}{m}\sum_{i=1}^{m} \boldsymbol{y}^{(i)} \cdot \log\left(\boldsymbol{h_w}(\boldsymbol{x}^{(i)})\right) + (\mathbf{1} - \boldsymbol{y}^{(i)}) \cdot \log\left(\mathbf{1} - \boldsymbol{h_w}(\boldsymbol{x}^{(i)})\right)] + \frac{\lambda}{2m}\sum_{j=1}^{k} \boldsymbol{w_j^2} \ .$$

- **The parameter $\boldsymbol{\lambda}$** controls the trade-off between training well and keeping the parameters small. When the value of λ is large, values of the weights will be small. When the value of λ is small, values of the weights will become large.
- **Train the one-vs-all classifier with different $\boldsymbol{\lambda}$ values**. Repeat Section 4, Section 6, and Section 7 with different λ values when λ =1, λ=5, and λ =10.
- **Question 3:** Find the accuracies of the classifier on the training data and the testing data with the different λ values when λ =1, λ=5, and λ =10.

You can work in groups of up to two people

What to Submit to Blackboard:

- **one** ZIP file that includes :
  1) **A report (15pts): contains answers for Question1 –Question3.**
  2) (85 pts): The live script file, "Lab4.mlx", in that all outputs are generated either inline or on the right side.

- **Pay attention to the zip file name convention:**
  Lab4_Student1 Lastname_Student2 Lastname.zip
  Ex) Lab4_Green_Smith.zip

1. Data source & displayData function:modified from Andrew Ng

Laboratory 4

Multivariable Logistic Regression Classification

Due Date: Beginning of Week 6 Lab

Concepts:

- Multivariable Logistic Regression Classification
- Regularization

Total Points: 100 points

Objectives:

Logistic regression is one of the most popular supervised classification algorithm. The one-vs-all multiclass logistic regression classifier is an extended version of logistic regression to solve multiclass classification problems. Students will implement the one-vs-all logistic regression classifier to distinguish ten digits in the MNIST hand written digit dataset and evaluate performance of the developed model on the test data that is not used during the training.

Files Needed:

- Lab4.mlx
- sigmoid.m
- displayData.m
- MNIST.mat

Assignment: Multiclass classification using regularized logistic regression

- **Multiclass classification** means a classification task with more than two classes. Logistic regression algorithm can also use to solve **multiclass classification problems.** In multiclass classification problems, a training set consisting of data points belonging to N different classes is given, and the goal is to construct a classifier which, given a new data sample, will correctly predict the class to which the new data belongs.

- The one-vs-all logistic regression classifier is implemented by training **multiple logistic regression classifiers,** one for each of the *C* classes in the training dataset. The algorithm turns the multiclass classification problem into **C binary classification problems** (i.e. where we predict only $y \in \{0, 1\}$) by taking values of one class and turn them into positive examples, and the rest of classes - into negatives**.**
- In Lab4, you will implement one-vs-all logistic regression to recognize hand-written digits on the MNIST dataset. Automated handwritten digit recognition is widely used today from

1. Data source & displayData function:modified from Andrew Ng

recognizing postal codes on mail envelopes to recognizing amounts written on bank checks. The MNIST dataset is a set of handwritten digits categorized 0-9 and is available at http://yann.lecun.com/exdb/mnist/.

1. Load the data from a file[1]

- The file "MNIST.mat " contains a training set of 4,000 handwritten digits and a test set of 1,000 handwritten digits. The images have a size of 20 by 20 grayscale images of digits. The 20 by 20 digit image is flattened into a 1 by 400 row vector. Each of these digit images becomes a single row in the data. The data is already normalized in the range of [0, 1] and separated into the training set and the test data.

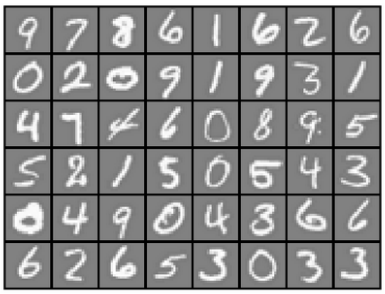
- Using the ***load( )*** function in Matlab, load the "MNIST.mat' file. The .mat format means that the data has been saved from the Matlab workspace. The matrix for the training data set is already named as 'xTrain' and the corresponding labels for the training set is named as 'yTrain', so you do not need to assign names to them. Similarly, the matrices for the test data set and the corresponding labels are names as 'xTest' and ' yTest', respectively.
- Note: There is no zero indexing in the Matlab. So the digits "1" to "9" are labeled as "1" to "9" and the digit "0 " is labeled as "10".

- Complete the lines of code in Section 1.

2. Visualize the data[1]

- Before starting on any prediction model, it is often useful to understand the data by plotting it.
- Display the randomly selected image samples from the training dataset. The **randperm**() function in Matlab returns a row vector containing a random permutation of the integers from 1 to n.
- Generate 48 numbers in the range of [1, 4000] using the **randperm**() function. Then, extract the image samples from the training data using the indexes generated by the **randperm**() function.

- Display the selected image samples by calling the provided function 'dispalyData()'. The plot is similar with the below figure. However the selected digit images will be different because the indexes are selected by the random number generator.



- Complete the lines of code in Section 2.

1. Data source & displayData function:modified from Andrew Ng

3. Add Bias term

- The "**bias term**" or "intercept parameter" allows us to move the linear model along the y-axis.
- Complete the lines of code in Section 3.

4. Training the one-vs-all classifier to find parameters using gradient descent

- To implement the one-vs-all classifier, multiple regularized logistic regression classifiers are trained one for each of $\boldsymbol{C}$ classes. Each classifier is trained independently using a " for"- loop from t=1 to $\boldsymbol{C}$. The training of the one-vs-all logistic regression with $\boldsymbol{C}$ classes is described as below.

  For each class t ,where t=1,..,C
  - Train a regularized logistic regression classifier $h_w^{(t)}(x)$ to predict the probability, p(y = $t|x$). The procedures for the regularized logistic regression are explained in the subsection 4.1.
  - To do this, all class t labels as positive samples (y=1) and all other classes as negative samples (y=0).

4.1 procedures for the logistic regression binary classifier

a) Sigmoid function

- The logistic regression hypothesis is defined as a sigmoid function of an input Z where Z is a liner model:

$$h_w^{(t)}(x) = \frac{1}{1+e^{-z}} \text{ , where } Z = \sum_{i=0}^{k} \boldsymbol{w_i x_i}$$

b) Log loss cost function

- The regularized cost function in logistic regression is the log loss function with the regularized term :

$$J^{(t)}(w) = -[\frac{1}{m}\sum_{i=1}^{m} \boldsymbol{y}^{(i)} \cdot \log\left(\boldsymbol{h_w}(\boldsymbol{x}^{(i)})\right) + (\boldsymbol{1} - \boldsymbol{y}^{(i)}) \cdot \log\left(\boldsymbol{1} - \boldsymbol{h_w}(\boldsymbol{x}^{(i)})\right)] + \frac{\lambda}{2m}\sum_{j=1}^{k} \boldsymbol{w_j^2}$$

- You will fit the classifier parameters, w with the training data using the gradient descent method. The objective of the training is to find the parameters, w that minimize the cost function:

$$\underset{w}{\text{Minimize}} \quad J^{(t)}(w)$$

- The gradient decent method will find the optimal weight , w using a " for"- loop from i=1 to iteration. The equations for the weight update is given:

$$\textbf{1) } \Delta \boldsymbol{w_0} = \frac{\boldsymbol{1}}{\boldsymbol{m}}\sum_{\boldsymbol{i=1}}^{\boldsymbol{m}}(\boldsymbol{h_w}(\boldsymbol{x^i}) - \boldsymbol{y^i})$$

1. Data source & displayData function:modified from Andrew Ng

$$w_0 = w_0 - \partial * \Delta w_0$$

2) for each w_j (j=1,…,k)

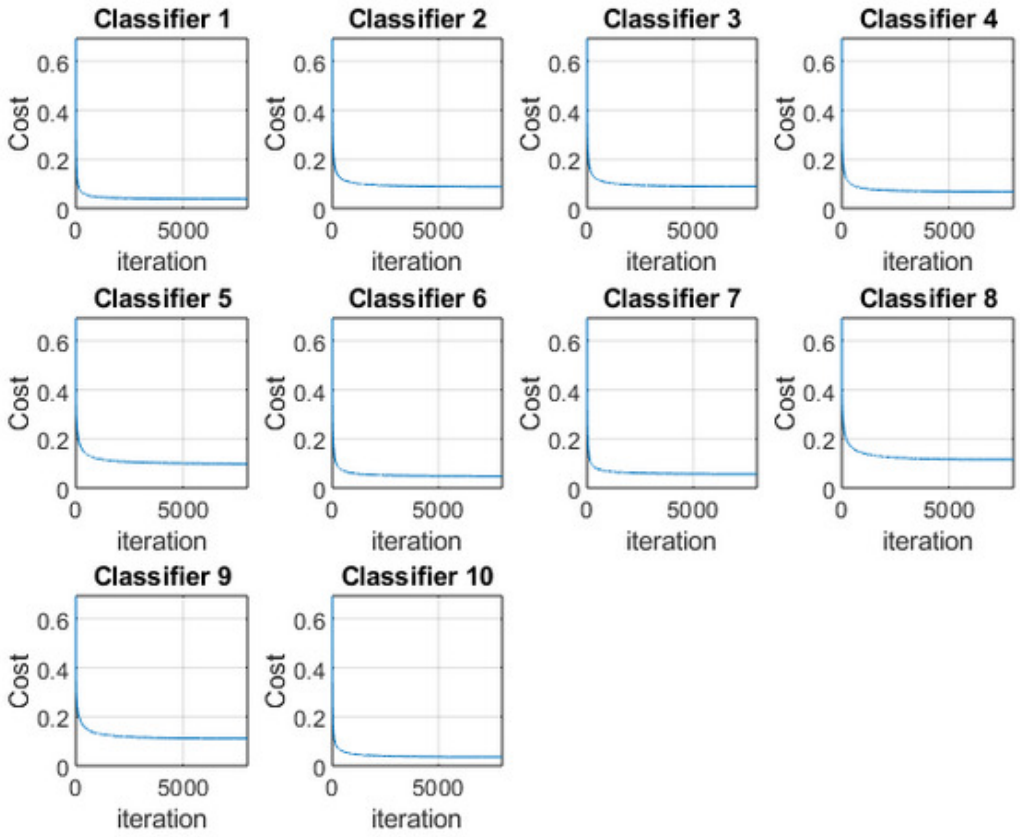
$$\Delta w_j = \frac{1}{m}\sum_{i=1}^{m}(h_w(x^i) - y^i)\, x_j^i + \frac{\lambda}{m} w_j$$

$$w_j = w_j - \partial * \Delta w_j$$

- To implement the regularization term, simply add the term, $\frac{\lambda}{m} w_j$ to the partial derivative of the cost function w.r.t w_j **except** w_0**.**

- At the end of the for loop iteration, you have the final log loss cost and the final parameters, w for the binary classification model for each class t.

5. **Plot the Log Loss cost convergence graph for each classifier.** Generate plots that shows how the cost changes over the iterations for each classifier t. You can use the ***subplot()*** function in the Matlab to display the multiple plots. The plots of the cost versus the iteration look like below:



1. Data source & displayData function:modified from Andrew Ng

- Complete the lines of code to implement **the Log Loss cost convergence graph** in Section 5.

6. The training accuracy of the one-vs-all classifier

- After the training is completed, the one-vs-all classifier can predict the digit contained in a given image. For the given input image, the probability that the input image belongs to each class is computed using the trained logistic regression classifiers. The one-vs-all classifier will pick the class for which the corresponding classifier outputs the highest probability.

- To predict the digits in given images,

  - Calculate the linear model $Z = \sum_{i=0}^{k} w_i x_i$
  - Calculate the sigmoid output of the input Z
  - Pick the class for which the corresponding classifier outputs the highest probability

- Write the lines of code for the calculation of the training accuracy in Section7.

Question 1: what is the accuracy of the developed model on the training data?

7. The one-vs-all classifier evaluation on the test data

- Add the intercept term (Bias term) in the input vector. A new input matrix is generated by concatenating the bias term vector and the new normalized input feature vector.
- Make the predictions on the new testing data using the one-vs-all classifier;

- **Calculate the accuracy of the developed classifier on the test data**

  - Calculate the linear model $Z = \sum_{i=0}^{k} w_i x_i$
  - Calculate the sigmoid output of the input Z
  - Pick the class for which the corresponding classifier outputs the highest probability

- Complete the lines of code to evaluate the multiclass classifier on the test data set
- **Question 2:** what is the accuracy of the multiclass classifier on the test data set?

8. Regularization

- Complex model (has lots of parameters) often prone to **overfitting.** Overfitting can be reduced by imposing a constraint on the overall magnitude of the parameters (weights),w. It introduces a penalty cost term for bringing in more features with the objective function.

- The regularized cost function in logistic regression is the log loss function with the regularized term :

$$J^{(t)}(w) = -[\frac{1}{m}\sum_{i=1}^{m} \boldsymbol{y}^{(i)} \cdot \log\left(\boldsymbol{h_w}(\boldsymbol{x}^{(i)})\right) + (\mathbf{1} - \boldsymbol{y}^{(i)}) \cdot \log\left(\mathbf{1} - \boldsymbol{h_w}(\boldsymbol{x}^{(i)})\right)] + \frac{\lambda}{2m}\sum_{j=1}^{k} \boldsymbol{w_j^2} \ .$$

- **The parameter $\boldsymbol{\lambda}$** controls the trade-off between training well and keeping the parameters small. When the value of λ is large, values of the weights will be small. When the value of λ is small, values of the weights will become large.
- **Train the one-vs-all classifier with different $\boldsymbol{\lambda}$ values**. Repeat Section 4, Section 6, and Section 7 with different λ values when λ =1, λ=5, and λ =10.
- **Question 3:** Find the accuracies of the classifier on the training data and the testing data with the different λ values when λ =1, λ=5, and λ =10.

You can work in groups of up to two people

What to Submit to Blackboard:

- **one** ZIP file that includes :
  1) **A report (15pts): contains answers for Question1 –Question3.**
  2) (85 pts): The live script file, "Lab4.mlx", in that all outputs are generated either inline or on the right side.

- **Pay attention to the zip file name convention:**
  Lab4_Student1 Lastname_Student2 Lastname.zip
  Ex) Lab4_Green_Smith.zip

1. Data source & displayData function:modified from Andrew Ng

Laboratory 4

Multivariable Logistic Regression Classification

Due Date: Beginning of Week 6 Lab

Concepts:

- Multivariable Logistic Regression Classification
- Regularization

Total Points: 100 points

Objectives:

Logistic regression is one of the most popular supervised classification algorithm. The one-vs-all multiclass logistic regression classifier is an extended version of logistic regression to solve multiclass classification problems. Students will implement the one-vs-all logistic regression classifier to distinguish ten digits in the MNIST hand written digit dataset and evaluate performance of the developed model on the test data that is not used during the training.

Files Needed:

- Lab4.mlx
- sigmoid.m
- displayData.m
- MNIST.mat

Assignment: Multiclass classification using regularized logistic regression

- **Multiclass classification** means a classification task with more than two classes. Logistic regression algorithm can also use to solve **multiclass classification problems.** In multiclass classification problems, a training set consisting of data points belonging to N different classes is given, and the goal is to construct a classifier which, given a new data sample, will correctly predict the class to which the new data belongs.

- The one-vs-all logistic regression classifier is implemented by training **multiple logistic regression classifiers,** one for each of the *C* classes in the training dataset. The algorithm turns the multiclass classification problem into **C binary classification problems** (i.e. where we predict only $y \in \{0, 1\}$) by taking values of one class and turn them into positive examples, and the rest of classes - into negatives**.**
- In Lab4, you will implement one-vs-all logistic regression to recognize hand-written digits on the MNIST dataset. Automated handwritten digit recognition is widely used today from

1. Data source & displayData function:modified from Andrew Ng

recognizing postal codes on mail envelopes to recognizing amounts written on bank checks. The MNIST dataset is a set of handwritten digits categorized 0-9 and is available at http://yann.lecun.com/exdb/mnist/.

1. Load the data from a file[1]

- The file "MNIST.mat " contains a training set of 4,000 handwritten digits and a test set of 1,000 handwritten digits. The images have a size of 20 by 20 grayscale images of digits. The 20 by 20 digit image is flattened into a 1 by 400 row vector. Each of these digit images becomes a single row in the data. The data is already normalized in the range of [0, 1] and separated into the training set and the test data.

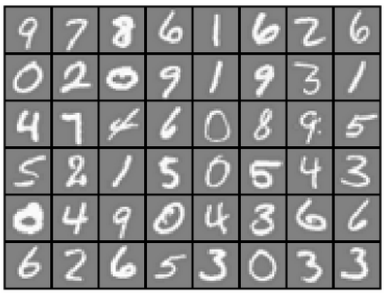
- Using the ***load( )*** function in Matlab, load the "MNIST.mat' file. The .mat format means that the data has been saved from the Matlab workspace. The matrix for the training data set is already named as 'xTrain' and the corresponding labels for the training set is named as 'yTrain', so you do not need to assign names to them. Similarly, the matrices for the test data set and the corresponding labels are names as 'xTest' and ' yTest', respectively.
- Note: There is no zero indexing in the Matlab. So the digits "1" to "9" are labeled as "1" to "9" and the digit "0 " is labeled as "10".

- Complete the lines of code in Section 1.

2. Visualize the data[1]

- Before starting on any prediction model, it is often useful to understand the data by plotting it.
- Display the randomly selected image samples from the training dataset. The **randperm**() function in Matlab returns a row vector containing a random permutation of the integers from 1 to n.
- Generate 48 numbers in the range of [1, 4000] using the **randperm**() function. Then, extract the image samples from the training data using the indexes generated by the **randperm**() function.

- Display the selected image samples by calling the provided function 'dispalyData()'. The plot is similar with the below figure. However the selected digit images will be different because the indexes are selected by the random number generator.



- Complete the lines of code in Section 2.

1. Data source & displayData function:modified from Andrew Ng

3. Add Bias term

- The "**bias term**" or "intercept parameter" allows us to move the linear model along the y-axis.
- Complete the lines of code in Section 3.

4. Training the one-vs-all classifier to find parameters using gradient descent

- To implement the one-vs-all classifier, multiple regularized logistic regression classifiers are trained one for each of $\boldsymbol{C}$ classes. Each classifier is trained independently using a " for"- loop from t=1 to $\boldsymbol{C}$. The training of the one-vs-all logistic regression with $\boldsymbol{C}$ classes is described as below.

  For each class t ,where t=1,..,C
  - Train a regularized logistic regression classifier $h_w^{(t)}(x)$ to predict the probability, p(y = $t|x$). The procedures for the regularized logistic regression are explained in the subsection 4.1.
  - To do this, all class t labels as positive samples (y=1) and all other classes as negative samples (y=0).

4.1 procedures for the logistic regression binary classifier

a) Sigmoid function

- The logistic regression hypothesis is defined as a sigmoid function of an input Z where Z is a liner model:

$$h_w^{(t)}(x) = \frac{1}{1+e^{-z}} \text{ , where } Z = \sum_{i=0}^{k} \boldsymbol{w_i x_i}$$

b) Log loss cost function

- The regularized cost function in logistic regression is the log loss function with the regularized term :

$$J^{(t)}(w) = -[\frac{1}{m}\sum_{i=1}^{m} \boldsymbol{y}^{(i)} \cdot \log\left(\boldsymbol{h_w}(\boldsymbol{x}^{(i)})\right) + (\boldsymbol{1} - \boldsymbol{y}^{(i)}) \cdot \log\left(\boldsymbol{1} - \boldsymbol{h_w}(\boldsymbol{x}^{(i)})\right)] + \frac{\lambda}{2m}\sum_{j=1}^{k} \boldsymbol{w_j^2}$$

- You will fit the classifier parameters, w with the training data using the gradient descent method. The objective of the training is to find the parameters, w that minimize the cost function:

$$\underset{w}{\text{Minimize}} \quad J^{(t)}(w)$$

- The gradient decent method will find the optimal weight , w using a " for"- loop from i=1 to iteration. The equations for the weight update is given:

$$\textbf{1) } \Delta \boldsymbol{w_0} = \frac{\boldsymbol{1}}{\boldsymbol{m}}\sum_{\boldsymbol{i=1}}^{\boldsymbol{m}}(\boldsymbol{h_w}(\boldsymbol{x^i}) - \boldsymbol{y^i})$$

1. Data source & displayData function:modified from Andrew Ng

$$w_0 = w_0 - \partial * \Delta w_0$$

2) for each w_j (j=1,…,k)

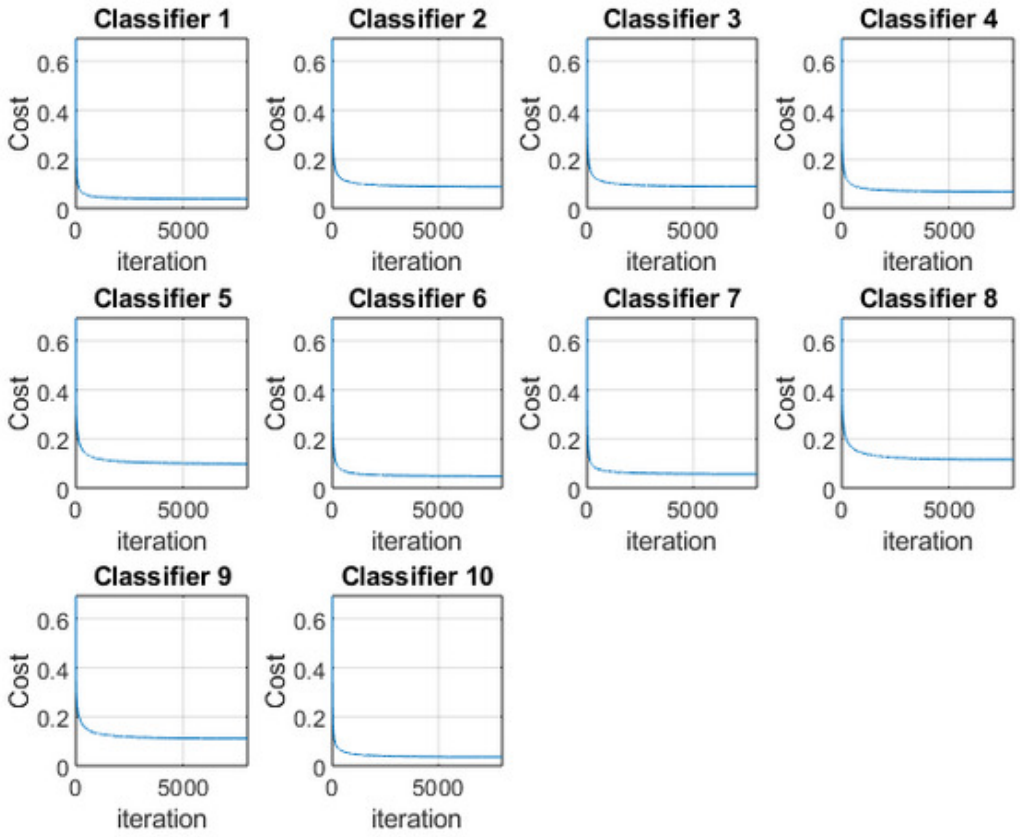
$$\Delta w_j = \frac{1}{m}\sum_{i=1}^{m}(h_w(x^i) - y^i)\, x_j^i + \frac{\lambda}{m} w_j$$

$$w_j = w_j - \partial * \Delta w_j$$

- To implement the regularization term, simply add the term, $\frac{\lambda}{m} w_j$ to the partial derivative of the cost function w.r.t w_j **except** w_0**.**

- At the end of the for loop iteration, you have the final log loss cost and the final parameters, w for the binary classification model for each class t.

5. **Plot the Log Loss cost convergence graph for each classifier.** Generate plots that shows how the cost changes over the iterations for each classifier t. You can use the ***subplot()*** function in the Matlab to display the multiple plots. The plots of the cost versus the iteration look like below:



1. Data source & displayData function:modified from Andrew Ng

- Complete the lines of code to implement **the Log Loss cost convergence graph** in Section 5.

6. The training accuracy of the one-vs-all classifier

- After the training is completed, the one-vs-all classifier can predict the digit contained in a given image. For the given input image, the probability that the input image belongs to each class is computed using the trained logistic regression classifiers. The one-vs-all classifier will pick the class for which the corresponding classifier outputs the highest probability.

- To predict the digits in given images,

  - Calculate the linear model $Z = \sum_{i=0}^{k} w_i x_i$
  - Calculate the sigmoid output of the input Z
  - Pick the class for which the corresponding classifier outputs the highest probability

- Write the lines of code for the calculation of the training accuracy in Section7.

Question 1: what is the accuracy of the developed model on the training data?

7. The one-vs-all classifier evaluation on the test data

- Add the intercept term (Bias term) in the input vector. A new input matrix is generated by concatenating the bias term vector and the new normalized input feature vector.
- Make the predictions on the new testing data using the one-vs-all classifier;

- **Calculate the accuracy of the developed classifier on the test data**

  - Calculate the linear model $Z = \sum_{i=0}^{k} w_i x_i$
  - Calculate the sigmoid output of the input Z
  - Pick the class for which the corresponding classifier outputs the highest probability

- Complete the lines of code to evaluate the multiclass classifier on the test data set
- **Question 2:** what is the accuracy of the multiclass classifier on the test data set?

8. Regularization

- Complex model (has lots of parameters) often prone to **overfitting.** Overfitting can be reduced by imposing a constraint on the overall magnitude of the parameters (weights),w. It introduces a penalty cost term for bringing in more features with the objective function.

- The regularized cost function in logistic regression is the log loss function with the regularized term :

$$J^{(t)}(w) = -[\frac{1}{m}\sum_{i=1}^{m} \boldsymbol{y}^{(i)} \cdot \log\left(\boldsymbol{h_w}(\boldsymbol{x}^{(i)})\right) + (\mathbf{1} - \boldsymbol{y}^{(i)}) \cdot \log\left(\mathbf{1} - \boldsymbol{h_w}(\boldsymbol{x}^{(i)})\right)] + \frac{\lambda}{2m}\sum_{j=1}^{k} \boldsymbol{w_j^2} \ .$$

- **The parameter $\boldsymbol{\lambda}$** controls the trade-off between training well and keeping the parameters small. When the value of λ is large, values of the weights will be small. When the value of λ is small, values of the weights will become large.
- **Train the one-vs-all classifier with different $\boldsymbol{\lambda}$ values**. Repeat Section 4, Section 6, and Section 7 with different λ values when λ =1, λ=5, and λ =10.
- **Question 3:** Find the accuracies of the classifier on the training data and the testing data with the different λ values when λ =1, λ=5, and λ =10.

You can work in groups of up to two people

What to Submit to Blackboard:

- **one** ZIP file that includes :
  1) **A report (15pts): contains answers for Question1 –Question3.**
  2) (85 pts): The live script file, "Lab4.mlx", in that all outputs are generated either inline or on the right side.

- **Pay attention to the zip file name convention:**
  Lab4_Student1 Lastname_Student2 Lastname.zip
  Ex) Lab4_Green_Smith.zip

1. Data source & displayData function:modified from Andrew Ng

Laboratory 4

Multivariable Logistic Regression Classification

Due Date: Beginning of Week 6 Lab

Concepts:

- Multivariable Logistic Regression Classification
- Regularization

Total Points: 100 points

Objectives:

Logistic regression is one of the most popular supervised classification algorithm. The one-vs-all multiclass logistic regression classifier is an extended version of logistic regression to solve multiclass classification problems. Students will implement the one-vs-all logistic regression classifier to distinguish ten digits in the MNIST hand written digit dataset and evaluate performance of the developed model on the test data that is not used during the training.

Files Needed:

- Lab4.mlx
- sigmoid.m
- displayData.m
- MNIST.mat

Assignment: Multiclass classification using regularized logistic regression

- **Multiclass classification** means a classification task with more than two classes. Logistic regression algorithm can also use to solve **multiclass classification problems.** In multiclass classification problems, a training set consisting of data points belonging to N different classes is given, and the goal is to construct a classifier which, given a new data sample, will correctly predict the class to which the new data belongs.

- The one-vs-all logistic regression classifier is implemented by training **multiple logistic regression classifiers,** one for each of the *C* classes in the training dataset. The algorithm turns the multiclass classification problem into **C binary classification problems** (i.e. where we predict only $y \in \{0, 1\}$) by taking values of one class and turn them into positive examples, and the rest of classes - into negatives**.**
- In Lab4, you will implement one-vs-all logistic regression to recognize hand-written digits on the MNIST dataset. Automated handwritten digit recognition is widely used today from

1. Data source & displayData function:modified from Andrew Ng

recognizing postal codes on mail envelopes to recognizing amounts written on bank checks. The MNIST dataset is a set of handwritten digits categorized 0-9 and is available at http://yann.lecun.com/exdb/mnist/.

1. Load the data from a file[1]

- The file "MNIST.mat " contains a training set of 4,000 handwritten digits and a test set of 1,000 handwritten digits. The images have a size of 20 by 20 grayscale images of digits. The 20 by 20 digit image is flattened into a 1 by 400 row vector. Each of these digit images becomes a single row in the data. The data is already normalized in the range of [0, 1] and separated into the training set and the test data.

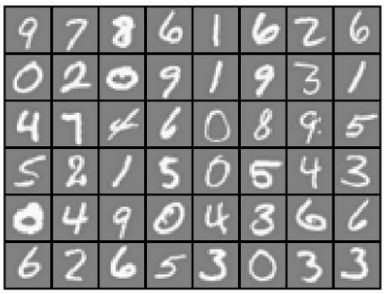
- Using the ***load( )*** function in Matlab, load the "MNIST.mat' file. The .mat format means that the data has been saved from the Matlab workspace. The matrix for the training data set is already named as 'xTrain' and the corresponding labels for the training set is named as 'yTrain', so you do not need to assign names to them. Similarly, the matrices for the test data set and the corresponding labels are names as 'xTest' and ' yTest', respectively.
- Note: There is no zero indexing in the Matlab. So the digits "1" to "9" are labeled as "1" to "9" and the digit "0 " is labeled as "10".

- Complete the lines of code in Section 1.

2. Visualize the data[1]

- Before starting on any prediction model, it is often useful to understand the data by plotting it.
- Display the randomly selected image samples from the training dataset. The **randperm**() function in Matlab returns a row vector containing a random permutation of the integers from 1 to n.
- Generate 48 numbers in the range of [1, 4000] using the **randperm**() function. Then, extract the image samples from the training data using the indexes generated by the **randperm**() function.

- Display the selected image samples by calling the provided function 'dispalyData()'. The plot is similar with the below figure. However the selected digit images will be different because the indexes are selected by the random number generator.



- Complete the lines of code in Section 2.

1. Data source & displayData function:modified from Andrew Ng

3. Add Bias term

- The "**bias term**" or "intercept parameter" allows us to move the linear model along the y-axis.
- Complete the lines of code in Section 3.

4. Training the one-vs-all classifier to find parameters using gradient descent

- To implement the one-vs-all classifier, multiple regularized logistic regression classifiers are trained one for each of $\boldsymbol{C}$ classes. Each classifier is trained independently using a " for"- loop from t=1 to $\boldsymbol{C}$. The training of the one-vs-all logistic regression with $\boldsymbol{C}$ classes is described as below.

  For each class t ,where t=1,..,C
  - Train a regularized logistic regression classifier $h_w^{(t)}(x)$ to predict the probability, p(y = $t|x$). The procedures for the regularized logistic regression are explained in the subsection 4.1.
  - To do this, all class t labels as positive samples (y=1) and all other classes as negative samples (y=0).

4.1 procedures for the logistic regression binary classifier

a) Sigmoid function

- The logistic regression hypothesis is defined as a sigmoid function of an input Z where Z is a liner model:

$$h_w^{(t)}(x) = \frac{1}{1+e^{-z}} \text{ , where } Z = \sum_{i=0}^{k} \boldsymbol{w_i x_i}$$

b) Log loss cost function

- The regularized cost function in logistic regression is the log loss function with the regularized term :

$$J^{(t)}(w) = -[\frac{1}{m}\sum_{i=1}^{m} \boldsymbol{y}^{(i)} \cdot \log\left(\boldsymbol{h_w}(\boldsymbol{x}^{(i)})\right) + (\boldsymbol{1} - \boldsymbol{y}^{(i)}) \cdot \log\left(\boldsymbol{1} - \boldsymbol{h_w}(\boldsymbol{x}^{(i)})\right)] + \frac{\lambda}{2m}\sum_{j=1}^{k} \boldsymbol{w_j^2}$$

- You will fit the classifier parameters, w with the training data using the gradient descent method. The objective of the training is to find the parameters, w that minimize the cost function:

$$\underset{w}{\text{Minimize}} \quad J^{(t)}(w)$$

- The gradient decent method will find the optimal weight , w using a " for"- loop from i=1 to iteration. The equations for the weight update is given:

$$\textbf{1) } \Delta \boldsymbol{w_0} = \frac{\boldsymbol{1}}{\boldsymbol{m}}\sum_{\boldsymbol{i=1}}^{\boldsymbol{m}}(\boldsymbol{h_w}(\boldsymbol{x^i}) - \boldsymbol{y^i})$$

1. Data source & displayData function:modified from Andrew Ng

$$w_0 = w_0 - \partial * \Delta w_0$$

2) for each w_j (j=1,…,k)

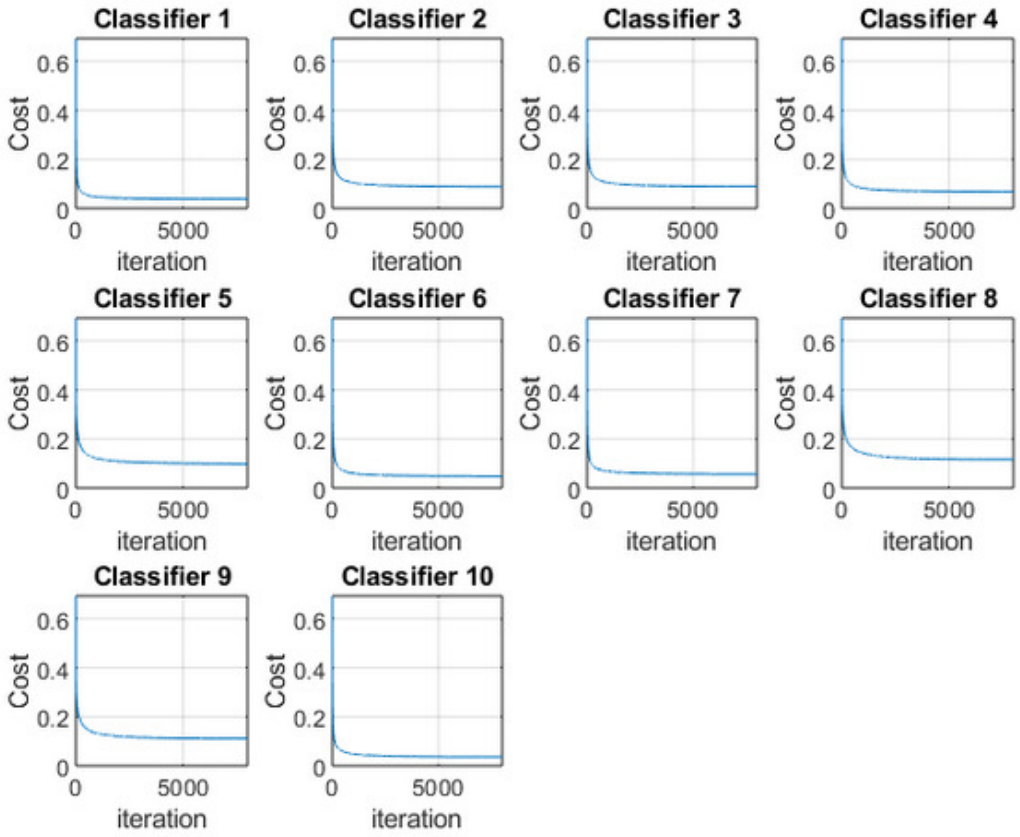
$$\Delta w_j = \frac{1}{m}\sum_{i=1}^{m}(h_w(x^i) - y^i)\, x_j^i + \frac{\lambda}{m} w_j$$

$$w_j = w_j - \partial * \Delta w_j$$

- To implement the regularization term, simply add the term, $\frac{\lambda}{m} w_j$ to the partial derivative of the cost function w.r.t w_j **except** w_0**.**

- At the end of the for loop iteration, you have the final log loss cost and the final parameters, w for the binary classification model for each class t.

5. **Plot the Log Loss cost convergence graph for each classifier.** Generate plots that shows how the cost changes over the iterations for each classifier t. You can use the ***subplot()*** function in the Matlab to display the multiple plots. The plots of the cost versus the iteration look like below:



1. Data source & displayData function:modified from Andrew Ng

- Complete the lines of code to implement **the Log Loss cost convergence graph** in Section 5.

6. The training accuracy of the one-vs-all classifier

- After the training is completed, the one-vs-all classifier can predict the digit contained in a given image. For the given input image, the probability that the input image belongs to each class is computed using the trained logistic regression classifiers. The one-vs-all classifier will pick the class for which the corresponding classifier outputs the highest probability.

- To predict the digits in given images,

  - Calculate the linear model $Z = \sum_{i=0}^{k} w_i x_i$
  - Calculate the sigmoid output of the input Z
  - Pick the class for which the corresponding classifier outputs the highest probability

- Write the lines of code for the calculation of the training accuracy in Section7.

Question 1: what is the accuracy of the developed model on the training data?

7. The one-vs-all classifier evaluation on the test data

- Add the intercept term (Bias term) in the input vector. A new input matrix is generated by concatenating the bias term vector and the new normalized input feature vector.
- Make the predictions on the new testing data using the one-vs-all classifier;

- **Calculate the accuracy of the developed classifier on the test data**

  - Calculate the linear model $Z = \sum_{i=0}^{k} w_i x_i$
  - Calculate the sigmoid output of the input Z
  - Pick the class for which the corresponding classifier outputs the highest probability

- Complete the lines of code to evaluate the multiclass classifier on the test data set
- **Question 2:** what is the accuracy of the multiclass classifier on the test data set?

8. Regularization

- Complex model (has lots of parameters) often prone to **overfitting.** Overfitting can be reduced by imposing a constraint on the overall magnitude of the parameters (weights),w. It introduces a penalty cost term for bringing in more features with the objective function.

- The regularized cost function in logistic regression is the log loss function with the regularized term :

$$J^{(t)}(w) = -[\frac{1}{m}\sum_{i=1}^{m} \boldsymbol{y}^{(i)} \cdot \log\left(\boldsymbol{h_w}(\boldsymbol{x}^{(i)})\right) + (\mathbf{1} - \boldsymbol{y}^{(i)}) \cdot \log\left(\mathbf{1} - \boldsymbol{h_w}(\boldsymbol{x}^{(i)})\right)] + \frac{\lambda}{2m}\sum_{j=1}^{k} \boldsymbol{w_j^2} \ .$$

- **The parameter $\boldsymbol{\lambda}$** controls the trade-off between training well and keeping the parameters small. When the value of λ is large, values of the weights will be small. When the value of λ is small, values of the weights will become large.
- **Train the one-vs-all classifier with different $\boldsymbol{\lambda}$ values**. Repeat Section 4, Section 6, and Section 7 with different λ values when λ =1, λ=5, and λ =10.
- **Question 3:** Find the accuracies of the classifier on the training data and the testing data with the different λ values when λ =1, λ=5, and λ =10.

You can work in groups of up to two people

What to Submit to Blackboard:

- **one** ZIP file that includes :
  1) **A report (15pts): contains answers for Question1 –Question3.**
  2) (85 pts): The live script file, "Lab4.mlx", in that all outputs are generated either inline or on the right side.

- **Pay attention to the zip file name convention:**
  Lab4_Student1 Lastname_Student2 Lastname.zip
  Ex) Lab4_Green_Smith.zip

1. Data source & displayData function:modified from Andrew Ng

Laboratory 4

Multivariable Logistic Regression Classification

Due Date: Beginning of Week 6 Lab

Concepts:

- Multivariable Logistic Regression Classification
- Regularization

Total Points: 100 points

Objectives:

Logistic regression is one of the most popular supervised classification algorithm. The one-vs-all multiclass logistic regression classifier is an extended version of logistic regression to solve multiclass classification problems. Students will implement the one-vs-all logistic regression classifier to distinguish ten digits in the MNIST hand written digit dataset and evaluate performance of the developed model on the test data that is not used during the training.

Files Needed:

- Lab4.mlx
- sigmoid.m
- displayData.m
- MNIST.mat

Assignment: Multiclass classification using regularized logistic regression

- **Multiclass classification** means a classification task with more than two classes. Logistic regression algorithm can also use to solve **multiclass classification problems.** In multiclass classification problems, a training set consisting of data points belonging to N different classes is given, and the goal is to construct a classifier which, given a new data sample, will correctly predict the class to which the new data belongs.

- The one-vs-all logistic regression classifier is implemented by training **multiple logistic regression classifiers,** one for each of the *C* classes in the training dataset. The algorithm turns the multiclass classification problem into **C binary classification problems** (i.e. where we predict only $y \in \{0, 1\}$) by taking values of one class and turn them into positive examples, and the rest of classes - into negatives**.**
- In Lab4, you will implement one-vs-all logistic regression to recognize hand-written digits on the MNIST dataset. Automated handwritten digit recognition is widely used today from

1. Data source & displayData function:modified from Andrew Ng

recognizing postal codes on mail envelopes to recognizing amounts written on bank checks. The MNIST dataset is a set of handwritten digits categorized 0-9 and is available at http://yann.lecun.com/exdb/mnist/.

1. Load the data from a file[1]

- The file "MNIST.mat " contains a training set of 4,000 handwritten digits and a test set of 1,000 handwritten digits. The images have a size of 20 by 20 grayscale images of digits. The 20 by 20 digit image is flattened into a 1 by 400 row vector. Each of these digit images becomes a single row in the data. The data is already normalized in the range of [0, 1] and separated into the training set and the test data.

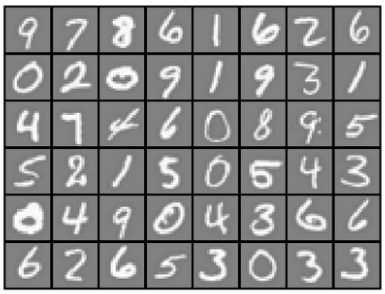
- Using the ***load( )*** function in Matlab, load the "MNIST.mat' file. The .mat format means that the data has been saved from the Matlab workspace. The matrix for the training data set is already named as 'xTrain' and the corresponding labels for the training set is named as 'yTrain', so you do not need to assign names to them. Similarly, the matrices for the test data set and the corresponding labels are names as 'xTest' and ' yTest', respectively.
- Note: There is no zero indexing in the Matlab. So the digits "1" to "9" are labeled as "1" to "9" and the digit "0 " is labeled as "10".

- Complete the lines of code in Section 1.

2. Visualize the data[1]

- Before starting on any prediction model, it is often useful to understand the data by plotting it.
- Display the randomly selected image samples from the training dataset. The **randperm**() function in Matlab returns a row vector containing a random permutation of the integers from 1 to n.
- Generate 48 numbers in the range of [1, 4000] using the **randperm**() function. Then, extract the image samples from the training data using the indexes generated by the **randperm**() function.

- Display the selected image samples by calling the provided function 'dispalyData()'. The plot is similar with the below figure. However the selected digit images will be different because the indexes are selected by the random number generator.



- Complete the lines of code in Section 2.

1. Data source & displayData function:modified from Andrew Ng

3. Add Bias term

- The "**bias term**" or "intercept parameter" allows us to move the linear model along the y-axis.
- Complete the lines of code in Section 3.

4. Training the one-vs-all classifier to find parameters using gradient descent

- To implement the one-vs-all classifier, multiple regularized logistic regression classifiers are trained one for each of $\boldsymbol{C}$ classes. Each classifier is trained independently using a " for"- loop from t=1 to $\boldsymbol{C}$. The training of the one-vs-all logistic regression with $\boldsymbol{C}$ classes is described as below.

  For each class t ,where t=1,..,C
  - Train a regularized logistic regression classifier $h_w^{(t)}(x)$ to predict the probability, p(y = $t|x$). The procedures for the regularized logistic regression are explained in the subsection 4.1.
  - To do this, all class t labels as positive samples (y=1) and all other classes as negative samples (y=0).

4.1 procedures for the logistic regression binary classifier

a) Sigmoid function

- The logistic regression hypothesis is defined as a sigmoid function of an input Z where Z is a liner model:

$$h_w^{(t)}(x) = \frac{1}{1+e^{-z}} \text{ , where } Z = \sum_{i=0}^{k} \boldsymbol{w_i x_i}$$

b) Log loss cost function

- The regularized cost function in logistic regression is the log loss function with the regularized term :

$$J^{(t)}(w) = -[\frac{1}{m}\sum_{i=1}^{m} \boldsymbol{y}^{(i)} \cdot \log\left(\boldsymbol{h_w}(\boldsymbol{x}^{(i)})\right) + (\boldsymbol{1} - \boldsymbol{y}^{(i)}) \cdot \log\left(\boldsymbol{1} - \boldsymbol{h_w}(\boldsymbol{x}^{(i)})\right)] + \frac{\lambda}{2m}\sum_{j=1}^{k} \boldsymbol{w_j^2}$$

- You will fit the classifier parameters, w with the training data using the gradient descent method. The objective of the training is to find the parameters, w that minimize the cost function:

$$\underset{w}{\text{Minimize}} \quad J^{(t)}(w)$$

- The gradient decent method will find the optimal weight , w using a " for"- loop from i=1 to iteration. The equations for the weight update is given:

$$\textbf{1) } \Delta \boldsymbol{w_0} = \frac{\boldsymbol{1}}{\boldsymbol{m}}\sum_{\boldsymbol{i=1}}^{\boldsymbol{m}}(\boldsymbol{h_w}(\boldsymbol{x^i}) - \boldsymbol{y^i})$$

1. Data source & displayData function:modified from Andrew Ng

$$w_0 = w_0 - \partial * \Delta w_0$$

2) for each w_j (j=1,…,k)

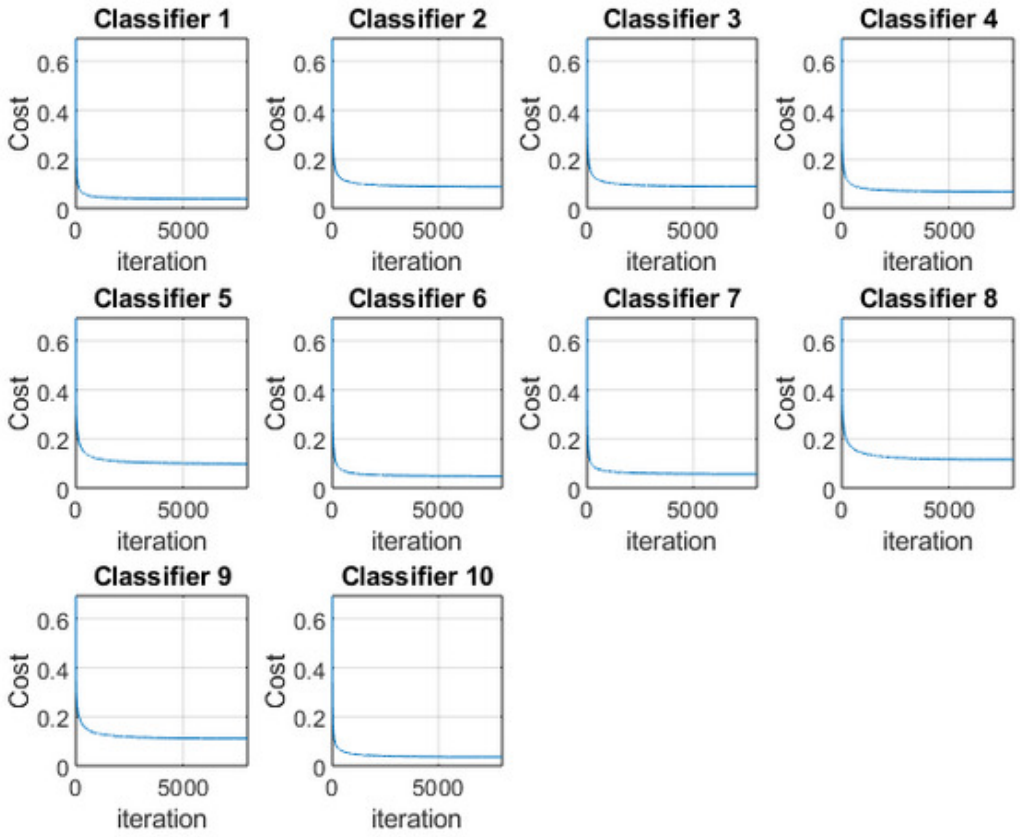
$$\Delta w_j = \frac{1}{m}\sum_{i=1}^{m}(h_w(x^i) - y^i)\, x_j^i + \frac{\lambda}{m} w_j$$

$$w_j = w_j - \partial * \Delta w_j$$

- To implement the regularization term, simply add the term, $\frac{\lambda}{m} w_j$ to the partial derivative of the cost function w.r.t w_j **except** w_0**.**

- At the end of the for loop iteration, you have the final log loss cost and the final parameters, w for the binary classification model for each class t.

5. **Plot the Log Loss cost convergence graph for each classifier.** Generate plots that shows how the cost changes over the iterations for each classifier t. You can use the ***subplot()*** function in the Matlab to display the multiple plots. The plots of the cost versus the iteration look like below:



1. Data source & displayData function:modified from Andrew Ng

- Complete the lines of code to implement **the Log Loss cost convergence graph** in Section 5.

6. The training accuracy of the one-vs-all classifier

- After the training is completed, the one-vs-all classifier can predict the digit contained in a given image. For the given input image, the probability that the input image belongs to each class is computed using the trained logistic regression classifiers. The one-vs-all classifier will pick the class for which the corresponding classifier outputs the highest probability.

- To predict the digits in given images,

  - Calculate the linear model $Z = \sum_{i=0}^{k} w_i x_i$
  - Calculate the sigmoid output of the input Z
  - Pick the class for which the corresponding classifier outputs the highest probability

- Write the lines of code for the calculation of the training accuracy in Section7.

Question 1: what is the accuracy of the developed model on the training data?

7. The one-vs-all classifier evaluation on the test data

- Add the intercept term (Bias term) in the input vector. A new input matrix is generated by concatenating the bias term vector and the new normalized input feature vector.
- Make the predictions on the new testing data using the one-vs-all classifier;

- **Calculate the accuracy of the developed classifier on the test data**

  - Calculate the linear model $Z = \sum_{i=0}^{k} w_i x_i$
  - Calculate the sigmoid output of the input Z
  - Pick the class for which the corresponding classifier outputs the highest probability

- Complete the lines of code to evaluate the multiclass classifier on the test data set
- **Question 2:** what is the accuracy of the multiclass classifier on the test data set?

8. Regularization

- Complex model (has lots of parameters) often prone to **overfitting.** Overfitting can be reduced by imposing a constraint on the overall magnitude of the parameters (weights),w. It introduces a penalty cost term for bringing in more features with the objective function.

- The regularized cost function in logistic regression is the log loss function with the regularized term :

$$J^{(t)}(w) = -[\frac{1}{m}\sum_{i=1}^{m} \boldsymbol{y}^{(i)} \cdot \log\left(\boldsymbol{h_w}(\boldsymbol{x}^{(i)})\right) + (\mathbf{1} - \boldsymbol{y}^{(i)}) \cdot \log\left(\mathbf{1} - \boldsymbol{h_w}(\boldsymbol{x}^{(i)})\right)] + \frac{\lambda}{2m}\sum_{j=1}^{k} \boldsymbol{w_j^2} \ .$$

- **The parameter $\boldsymbol{\lambda}$** controls the trade-off between training well and keeping the parameters small. When the value of λ is large, values of the weights will be small. When the value of λ is small, values of the weights will become large.
- **Train the one-vs-all classifier with different $\boldsymbol{\lambda}$ values**. Repeat Section 4, Section 6, and Section 7 with different λ values when λ =1, λ=5, and λ =10.
- **Question 3:** Find the accuracies of the classifier on the training data and the testing data with the different λ values when λ =1, λ=5, and λ =10.

You can work in groups of up to two people

What to Submit to Blackboard:

- **one** ZIP file that includes :
  1) **A report (15pts): contains answers for Question1 –Question3.**
  2) (85 pts): The live script file, "Lab4.mlx", in that all outputs are generated either inline or on the right side.

- **Pay attention to the zip file name convention:**
  Lab4_Student1 Lastname_Student2 Lastname.zip
  Ex) Lab4_Green_Smith.zip

1. Data source & displayData function:modified from Andrew Ng

Laboratory 4

Multivariable Logistic Regression Classification

Due Date: Beginning of Week 6 Lab

Concepts:

- Multivariable Logistic Regression Classification
- Regularization

Total Points: 100 points

Objectives:

Logistic regression is one of the most popular supervised classification algorithm. The one-vs-all multiclass logistic regression classifier is an extended version of logistic regression to solve multiclass classification problems. Students will implement the one-vs-all logistic regression classifier to distinguish ten digits in the MNIST hand written digit dataset and evaluate performance of the developed model on the test data that is not used during the training.

Files Needed:

- Lab4.mlx
- sigmoid.m
- displayData.m
- MNIST.mat

Assignment: Multiclass classification using regularized logistic regression

- **Multiclass classification** means a classification task with more than two classes. Logistic regression algorithm can also use to solve **multiclass classification problems.** In multiclass classification problems, a training set consisting of data points belonging to N different classes is given, and the goal is to construct a classifier which, given a new data sample, will correctly predict the class to which the new data belongs.

- The one-vs-all logistic regression classifier is implemented by training **multiple logistic regression classifiers,** one for each of the *C* classes in the training dataset. The algorithm turns the multiclass classification problem into **C binary classification problems** (i.e. where we predict only $y \in \{0, 1\}$) by taking values of one class and turn them into positive examples, and the rest of classes - into negatives**.**
- In Lab4, you will implement one-vs-all logistic regression to recognize hand-written digits on the MNIST dataset. Automated handwritten digit recognition is widely used today from

1. Data source & displayData function:modified from Andrew Ng

recognizing postal codes on mail envelopes to recognizing amounts written on bank checks. The MNIST dataset is a set of handwritten digits categorized 0-9 and is available at http://yann.lecun.com/exdb/mnist/.

1. Load the data from a file[1]

- The file "MNIST.mat " contains a training set of 4,000 handwritten digits and a test set of 1,000 handwritten digits. The images have a size of 20 by 20 grayscale images of digits. The 20 by 20 digit image is flattened into a 1 by 400 row vector. Each of these digit images becomes a single row in the data. The data is already normalized in the range of [0, 1] and separated into the training set and the test data.

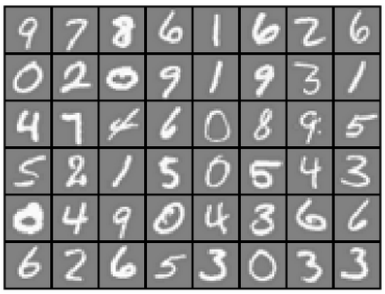
- Using the ***load( )*** function in Matlab, load the "MNIST.mat' file. The .mat format means that the data has been saved from the Matlab workspace. The matrix for the training data set is already named as 'xTrain' and the corresponding labels for the training set is named as 'yTrain', so you do not need to assign names to them. Similarly, the matrices for the test data set and the corresponding labels are names as 'xTest' and ' yTest', respectively.
- Note: There is no zero indexing in the Matlab. So the digits "1" to "9" are labeled as "1" to "9" and the digit "0 " is labeled as "10".

- Complete the lines of code in Section 1.

2. Visualize the data[1]

- Before starting on any prediction model, it is often useful to understand the data by plotting it.
- Display the randomly selected image samples from the training dataset. The **randperm**() function in Matlab returns a row vector containing a random permutation of the integers from 1 to n.
- Generate 48 numbers in the range of [1, 4000] using the **randperm**() function. Then, extract the image samples from the training data using the indexes generated by the **randperm**() function.

- Display the selected image samples by calling the provided function 'dispalyData()'. The plot is similar with the below figure. However the selected digit images will be different because the indexes are selected by the random number generator.



- Complete the lines of code in Section 2.

1. Data source & displayData function:modified from Andrew Ng

3. Add Bias term

- The "**bias term**" or "intercept parameter" allows us to move the linear model along the y-axis.
- Complete the lines of code in Section 3.

4. Training the one-vs-all classifier to find parameters using gradient descent

- To implement the one-vs-all classifier, multiple regularized logistic regression classifiers are trained one for each of $\boldsymbol{C}$ classes. Each classifier is trained independently using a " for"- loop from t=1 to $\boldsymbol{C}$. The training of the one-vs-all logistic regression with $\boldsymbol{C}$ classes is described as below.

  For each class t ,where t=1,..,C
  - Train a regularized logistic regression classifier $h_w^{(t)}(x)$ to predict the probability, p(y = $t|x$). The procedures for the regularized logistic regression are explained in the subsection 4.1.
  - To do this, all class t labels as positive samples (y=1) and all other classes as negative samples (y=0).

4.1 procedures for the logistic regression binary classifier

a) Sigmoid function

- The logistic regression hypothesis is defined as a sigmoid function of an input Z where Z is a liner model:

$$h_w^{(t)}(x) = \frac{1}{1+e^{-z}} \text{ , where } Z = \sum_{i=0}^{k} \boldsymbol{w_i x_i}$$

b) Log loss cost function

- The regularized cost function in logistic regression is the log loss function with the regularized term :

$$J^{(t)}(w) = -[\frac{1}{m}\sum_{i=1}^{m} \boldsymbol{y}^{(i)} \cdot \log\left(\boldsymbol{h_w}(\boldsymbol{x}^{(i)})\right) + (\boldsymbol{1} - \boldsymbol{y}^{(i)}) \cdot \log\left(\boldsymbol{1} - \boldsymbol{h_w}(\boldsymbol{x}^{(i)})\right)] + \frac{\lambda}{2m}\sum_{j=1}^{k} \boldsymbol{w_j^2}$$

- You will fit the classifier parameters, w with the training data using the gradient descent method. The objective of the training is to find the parameters, w that minimize the cost function:

$$\underset{w}{\text{Minimize}} \quad J^{(t)}(w)$$

- The gradient decent method will find the optimal weight , w using a " for"- loop from i=1 to iteration. The equations for the weight update is given:

$$\textbf{1) } \Delta \boldsymbol{w_0} = \frac{\boldsymbol{1}}{\boldsymbol{m}}\sum_{\boldsymbol{i=1}}^{\boldsymbol{m}}(\boldsymbol{h_w}(\boldsymbol{x^i}) - \boldsymbol{y^i})$$

1. Data source & displayData function:modified from Andrew Ng

$$w_0 = w_0 - \partial * \Delta w_0$$

2) for each w_j (j=1,…,k)

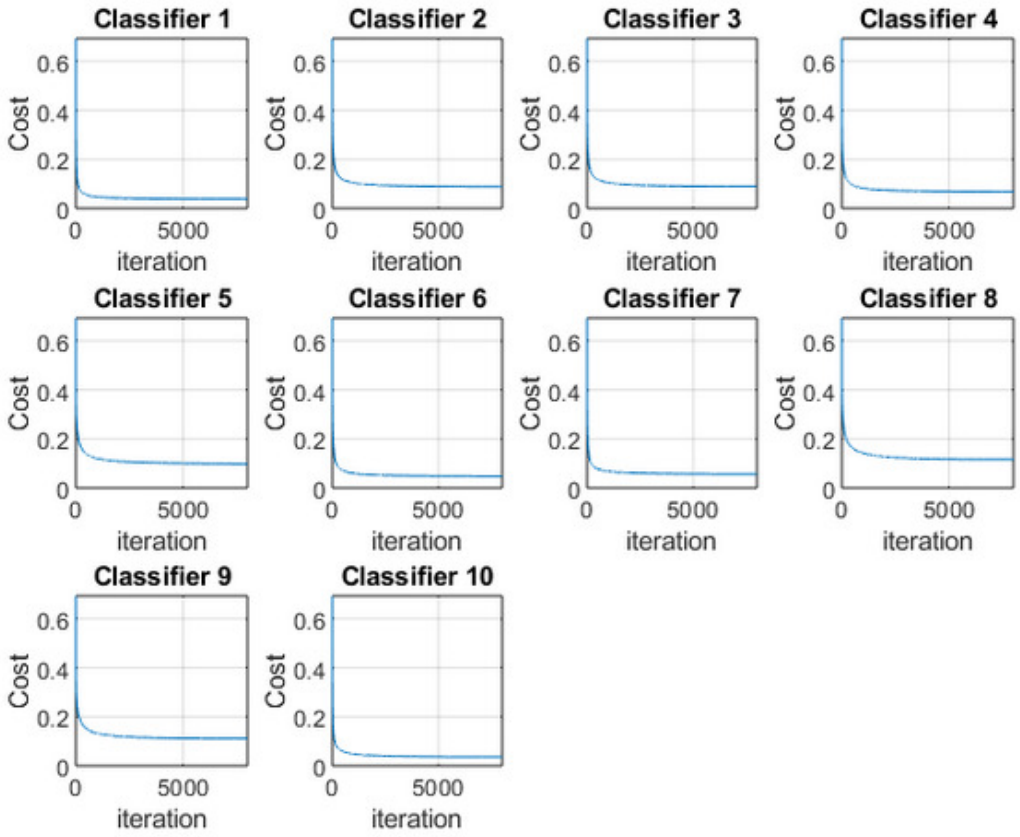
$$\Delta w_j = \frac{1}{m}\sum_{i=1}^{m}(h_w(x^i) - y^i)\, x_j^i + \frac{\lambda}{m} w_j$$

$$w_j = w_j - \partial * \Delta w_j$$

- To implement the regularization term, simply add the term, $\frac{\lambda}{m} w_j$ to the partial derivative of the cost function w.r.t w_j **except** w_0**.**

- At the end of the for loop iteration, you have the final log loss cost and the final parameters, w for the binary classification model for each class t.

5. **Plot the Log Loss cost convergence graph for each classifier.** Generate plots that shows how the cost changes over the iterations for each classifier t. You can use the ***subplot()*** function in the Matlab to display the multiple plots. The plots of the cost versus the iteration look like below:



1. Data source & displayData function:modified from Andrew Ng

- Complete the lines of code to implement **the Log Loss cost convergence graph** in Section 5.

6. The training accuracy of the one-vs-all classifier

- After the training is completed, the one-vs-all classifier can predict the digit contained in a given image. For the given input image, the probability that the input image belongs to each class is computed using the trained logistic regression classifiers. The one-vs-all classifier will pick the class for which the corresponding classifier outputs the highest probability.

- To predict the digits in given images,

  - Calculate the linear model $Z = \sum_{i=0}^{k} w_i x_i$
  - Calculate the sigmoid output of the input Z
  - Pick the class for which the corresponding classifier outputs the highest probability

- Write the lines of code for the calculation of the training accuracy in Section7.

Question 1: what is the accuracy of the developed model on the training data?

7. The one-vs-all classifier evaluation on the test data

- Add the intercept term (Bias term) in the input vector. A new input matrix is generated by concatenating the bias term vector and the new normalized input feature vector.
- Make the predictions on the new testing data using the one-vs-all classifier;

- **Calculate the accuracy of the developed classifier on the test data**

  - Calculate the linear model $Z = \sum_{i=0}^{k} w_i x_i$
  - Calculate the sigmoid output of the input Z
  - Pick the class for which the corresponding classifier outputs the highest probability

- Complete the lines of code to evaluate the multiclass classifier on the test data set
- **Question 2:** what is the accuracy of the multiclass classifier on the test data set?

8. Regularization

- Complex model (has lots of parameters) often prone to **overfitting.** Overfitting can be reduced by imposing a constraint on the overall magnitude of the parameters (weights),w. It introduces a penalty cost term for bringing in more features with the objective function.

- The regularized cost function in logistic regression is the log loss function with the regularized term :

$$J^{(t)}(w) = -[\frac{1}{m}\sum_{i=1}^{m} \boldsymbol{y}^{(i)} \cdot \log\left(\boldsymbol{h_w}(\boldsymbol{x}^{(i)})\right) + (\mathbf{1} - \boldsymbol{y}^{(i)}) \cdot \log\left(\mathbf{1} - \boldsymbol{h_w}(\boldsymbol{x}^{(i)})\right)] + \frac{\lambda}{2m}\sum_{j=1}^{k} \boldsymbol{w_j^2} \ .$$

- **The parameter $\boldsymbol{\lambda}$** controls the trade-off between training well and keeping the parameters small. When the value of λ is large, values of the weights will be small. When the value of λ is small, values of the weights will become large.
- **Train the one-vs-all classifier with different $\boldsymbol{\lambda}$ values**. Repeat Section 4, Section 6, and Section 7 with different λ values when λ =1, λ=5, and λ =10.
- **Question 3:** Find the accuracies of the classifier on the training data and the testing data with the different λ values when λ =1, λ=5, and λ =10.

You can work in groups of up to two people

What to Submit to Blackboard:

- **one** ZIP file that includes :
  1) **A report (15pts): contains answers for Question1 –Question3.**
  2) (85 pts): The live script file, "Lab4.mlx", in that all outputs are generated either inline or on the right side.

- **Pay attention to the zip file name convention:**
  Lab4_Student1 Lastname_Student2 Lastname.zip
  Ex) Lab4_Green_Smith.zip

1. Data source & displayData function:modified from Andrew Ng

Laboratory 4

Multivariable Logistic Regression Classification

Due Date: Beginning of Week 6 Lab

Concepts:

- Multivariable Logistic Regression Classification
- Regularization

Total Points: 100 points

Objectives:

Logistic regression is one of the most popular supervised classification algorithm. The one-vs-all multiclass logistic regression classifier is an extended version of logistic regression to solve multiclass classification problems. Students will implement the one-vs-all logistic regression classifier to distinguish ten digits in the MNIST hand written digit dataset and evaluate performance of the developed model on the test data that is not used during the training.

Files Needed:

- Lab4.mlx
- sigmoid.m
- displayData.m
- MNIST.mat

Assignment: Multiclass classification using regularized logistic regression

- **Multiclass classification** means a classification task with more than two classes. Logistic regression algorithm can also use to solve **multiclass classification problems.** In multiclass classification problems, a training set consisting of data points belonging to N different classes is given, and the goal is to construct a classifier which, given a new data sample, will correctly predict the class to which the new data belongs.

- The one-vs-all logistic regression classifier is implemented by training **multiple logistic regression classifiers,** one for each of the *C* classes in the training dataset. The algorithm turns the multiclass classification problem into **C binary classification problems** (i.e. where we predict only $y \in \{0, 1\}$) by taking values of one class and turn them into positive examples, and the rest of classes - into negatives**.**
- In Lab4, you will implement one-vs-all logistic regression to recognize hand-written digits on the MNIST dataset. Automated handwritten digit recognition is widely used today from

1. Data source & displayData function:modified from Andrew Ng

recognizing postal codes on mail envelopes to recognizing amounts written on bank checks. The MNIST dataset is a set of handwritten digits categorized 0-9 and is available at http://yann.lecun.com/exdb/mnist/.

1. Load the data from a file[1]

- The file "MNIST.mat " contains a training set of 4,000 handwritten digits and a test set of 1,000 handwritten digits. The images have a size of 20 by 20 grayscale images of digits. The 20 by 20 digit image is flattened into a 1 by 400 row vector. Each of these digit images becomes a single row in the data. The data is already normalized in the range of [0, 1] and separated into the training set and the test data.

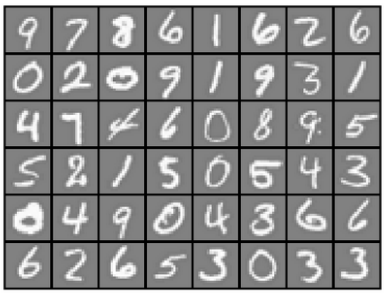
- Using the ***load( )*** function in Matlab, load the "MNIST.mat' file. The .mat format means that the data has been saved from the Matlab workspace. The matrix for the training data set is already named as 'xTrain' and the corresponding labels for the training set is named as 'yTrain', so you do not need to assign names to them. Similarly, the matrices for the test data set and the corresponding labels are names as 'xTest' and ' yTest', respectively.
- Note: There is no zero indexing in the Matlab. So the digits "1" to "9" are labeled as "1" to "9" and the digit "0 " is labeled as "10".

- Complete the lines of code in Section 1.

2. Visualize the data[1]

- Before starting on any prediction model, it is often useful to understand the data by plotting it.
- Display the randomly selected image samples from the training dataset. The **randperm**() function in Matlab returns a row vector containing a random permutation of the integers from 1 to n.
- Generate 48 numbers in the range of [1, 4000] using the **randperm**() function. Then, extract the image samples from the training data using the indexes generated by the **randperm**() function.

- Display the selected image samples by calling the provided function 'dispalyData()'. The plot is similar with the below figure. However the selected digit images will be different because the indexes are selected by the random number generator.



- Complete the lines of code in Section 2.

1. Data source & displayData function:modified from Andrew Ng

3. Add Bias term

- The "**bias term**" or "intercept parameter" allows us to move the linear model along the y-axis.
- Complete the lines of code in Section 3.

4. Training the one-vs-all classifier to find parameters using gradient descent

- To implement the one-vs-all classifier, multiple regularized logistic regression classifiers are trained one for each of $\boldsymbol{C}$ classes. Each classifier is trained independently using a " for"- loop from t=1 to $\boldsymbol{C}$. The training of the one-vs-all logistic regression with $\boldsymbol{C}$ classes is described as below.

  For each class t ,where t=1,..,C
  - Train a regularized logistic regression classifier $h_w^{(t)}(x)$ to predict the probability, p(y = $t|x$). The procedures for the regularized logistic regression are explained in the subsection 4.1.
  - To do this, all class t labels as positive samples (y=1) and all other classes as negative samples (y=0).

4.1 procedures for the logistic regression binary classifier

a) Sigmoid function

- The logistic regression hypothesis is defined as a sigmoid function of an input Z where Z is a liner model:

$$h_w^{(t)}(x) = \frac{1}{1+e^{-z}} \text{ , where } Z = \sum_{i=0}^{k} \boldsymbol{w_i x_i}$$

b) Log loss cost function

- The regularized cost function in logistic regression is the log loss function with the regularized term :

$$J^{(t)}(w) = -[\frac{1}{m}\sum_{i=1}^{m} \boldsymbol{y}^{(i)} \cdot \log\left(\boldsymbol{h_w}(\boldsymbol{x}^{(i)})\right) + (\boldsymbol{1} - \boldsymbol{y}^{(i)}) \cdot \log\left(\boldsymbol{1} - \boldsymbol{h_w}(\boldsymbol{x}^{(i)})\right)] + \frac{\lambda}{2m}\sum_{j=1}^{k} \boldsymbol{w_j^2}$$

- You will fit the classifier parameters, w with the training data using the gradient descent method. The objective of the training is to find the parameters, w that minimize the cost function:

$$\underset{w}{\text{Minimize}} \quad J^{(t)}(w)$$

- The gradient decent method will find the optimal weight , w using a " for"- loop from i=1 to iteration. The equations for the weight update is given:

$$\textbf{1) } \Delta \boldsymbol{w_0} = \frac{\boldsymbol{1}}{\boldsymbol{m}}\sum_{\boldsymbol{i=1}}^{\boldsymbol{m}}(\boldsymbol{h_w}(\boldsymbol{x^i}) - \boldsymbol{y^i})$$

1. Data source & displayData function:modified from Andrew Ng

$$w_0 = w_0 - \partial * \Delta w_0$$

2) for each w_j (j=1,…,k)

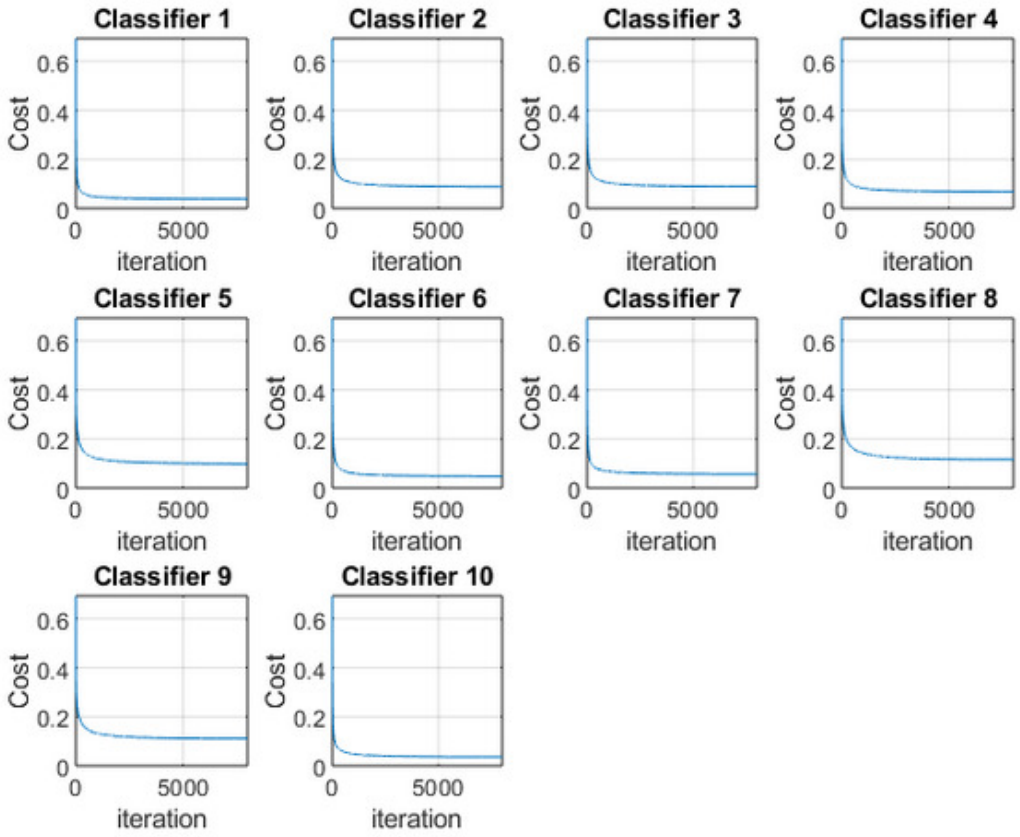
$$\Delta w_j = \frac{1}{m}\sum_{i=1}^{m}(h_w(x^i) - y^i)\, x_j^i + \frac{\lambda}{m} w_j$$

$$w_j = w_j - \partial * \Delta w_j$$

- To implement the regularization term, simply add the term, $\frac{\lambda}{m} w_j$ to the partial derivative of the cost function w.r.t w_j **except** w_0**.**

- At the end of the for loop iteration, you have the final log loss cost and the final parameters, w for the binary classification model for each class t.

5. **Plot the Log Loss cost convergence graph for each classifier.** Generate plots that shows how the cost changes over the iterations for each classifier t. You can use the ***subplot()*** function in the Matlab to display the multiple plots. The plots of the cost versus the iteration look like below:



1. Data source & displayData function:modified from Andrew Ng

- Complete the lines of code to implement **the Log Loss cost convergence graph** in Section 5.

6. The training accuracy of the one-vs-all classifier

- After the training is completed, the one-vs-all classifier can predict the digit contained in a given image. For the given input image, the probability that the input image belongs to each class is computed using the trained logistic regression classifiers. The one-vs-all classifier will pick the class for which the corresponding classifier outputs the highest probability.

- To predict the digits in given images,

  - Calculate the linear model $Z = \sum_{i=0}^{k} w_i x_i$
  - Calculate the sigmoid output of the input Z
  - Pick the class for which the corresponding classifier outputs the highest probability

- Write the lines of code for the calculation of the training accuracy in Section7.

Question 1: what is the accuracy of the developed model on the training data?

7. The one-vs-all classifier evaluation on the test data

- Add the intercept term (Bias term) in the input vector. A new input matrix is generated by concatenating the bias term vector and the new normalized input feature vector.
- Make the predictions on the new testing data using the one-vs-all classifier;

- **Calculate the accuracy of the developed classifier on the test data**

  - Calculate the linear model $Z = \sum_{i=0}^{k} w_i x_i$
  - Calculate the sigmoid output of the input Z
  - Pick the class for which the corresponding classifier outputs the highest probability

- Complete the lines of code to evaluate the multiclass classifier on the test data set
- **Question 2:** what is the accuracy of the multiclass classifier on the test data set?

8. Regularization

- Complex model (has lots of parameters) often prone to **overfitting.** Overfitting can be reduced by imposing a constraint on the overall magnitude of the parameters (weights),w. It introduces a penalty cost term for bringing in more features with the objective function.

- The regularized cost function in logistic regression is the log loss function with the regularized term :

$$J^{(t)}(w) = -[\frac{1}{m}\sum_{i=1}^{m} \boldsymbol{y}^{(i)} \cdot \log\left(\boldsymbol{h_w}(\boldsymbol{x}^{(i)})\right) + (\mathbf{1} - \boldsymbol{y}^{(i)}) \cdot \log\left(\mathbf{1} - \boldsymbol{h_w}(\boldsymbol{x}^{(i)})\right)] + \frac{\lambda}{2m}\sum_{j=1}^{k} \boldsymbol{w_j^2} \ .$$

- **The parameter $\boldsymbol{\lambda}$** controls the trade-off between training well and keeping the parameters small. When the value of λ is large, values of the weights will be small. When the value of λ is small, values of the weights will become large.
- **Train the one-vs-all classifier with different $\boldsymbol{\lambda}$ values**. Repeat Section 4, Section 6, and Section 7 with different λ values when λ =1, λ=5, and λ =10.
- **Question 3:** Find the accuracies of the classifier on the training data and the testing data with the different λ values when λ =1, λ=5, and λ =10.

You can work in groups of up to two people

What to Submit to Blackboard:

- **one** ZIP file that includes :
  1) **A report (15pts): contains answers for Question1 –Question3.**
  2) (85 pts): The live script file, "Lab4.mlx", in that all outputs are generated either inline or on the right side.

- **Pay attention to the zip file name convention:**
  Lab4_Student1 Lastname_Student2 Lastname.zip
  Ex) Lab4_Green_Smith.zip

1. Data source & displayData function:modified from Andrew Ng

Laboratory 4

Multivariable Logistic Regression Classification

Due Date: Beginning of Week 6 Lab

Concepts:

- Multivariable Logistic Regression Classification
- Regularization

Total Points: 100 points

Objectives:

Logistic regression is one of the most popular supervised classification algorithm. The one-vs-all multiclass logistic regression classifier is an extended version of logistic regression to solve multiclass classification problems. Students will implement the one-vs-all logistic regression classifier to distinguish ten digits in the MNIST hand written digit dataset and evaluate performance of the developed model on the test data that is not used during the training.

Files Needed:

- Lab4.mlx
- sigmoid.m
- displayData.m
- MNIST.mat

Assignment: Multiclass classification using regularized logistic regression

- **Multiclass classification** means a classification task with more than two classes. Logistic regression algorithm can also use to solve **multiclass classification problems.** In multiclass classification problems, a training set consisting of data points belonging to N different classes is given, and the goal is to construct a classifier which, given a new data sample, will correctly predict the class to which the new data belongs.

- The one-vs-all logistic regression classifier is implemented by training **multiple logistic regression classifiers,** one for each of the *C* classes in the training dataset. The algorithm turns the multiclass classification problem into **C binary classification problems** (i.e. where we predict only $y \in \{0, 1\}$) by taking values of one class and turn them into positive examples, and the rest of classes - into negatives**.**
- In Lab4, you will implement one-vs-all logistic regression to recognize hand-written digits on the MNIST dataset. Automated handwritten digit recognition is widely used today from

1. Data source & displayData function:modified from Andrew Ng

recognizing postal codes on mail envelopes to recognizing amounts written on bank checks. The MNIST dataset is a set of handwritten digits categorized 0-9 and is available at http://yann.lecun.com/exdb/mnist/.

1. Load the data from a file[1]

- The file "MNIST.mat " contains a training set of 4,000 handwritten digits and a test set of 1,000 handwritten digits. The images have a size of 20 by 20 grayscale images of digits. The 20 by 20 digit image is flattened into a 1 by 400 row vector. Each of these digit images becomes a single row in the data. The data is already normalized in the range of [0, 1] and separated into the training set and the test data.

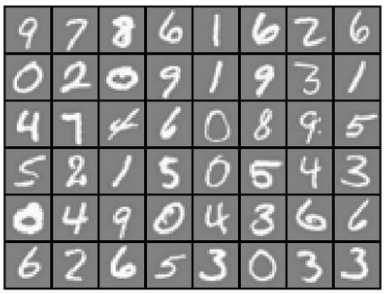
- Using the ***load( )*** function in Matlab, load the "MNIST.mat' file. The .mat format means that the data has been saved from the Matlab workspace. The matrix for the training data set is already named as 'xTrain' and the corresponding labels for the training set is named as 'yTrain', so you do not need to assign names to them. Similarly, the matrices for the test data set and the corresponding labels are names as 'xTest' and ' yTest', respectively.
- Note: There is no zero indexing in the Matlab. So the digits "1" to "9" are labeled as "1" to "9" and the digit "0 " is labeled as "10".

- Complete the lines of code in Section 1.

2. Visualize the data[1]

- Before starting on any prediction model, it is often useful to understand the data by plotting it.
- Display the randomly selected image samples from the training dataset. The **randperm**() function in Matlab returns a row vector containing a random permutation of the integers from 1 to n.
- Generate 48 numbers in the range of [1, 4000] using the **randperm**() function. Then, extract the image samples from the training data using the indexes generated by the **randperm**() function.

- Display the selected image samples by calling the provided function 'dispalyData()'. The plot is similar with the below figure. However the selected digit images will be different because the indexes are selected by the random number generator.



- Complete the lines of code in Section 2.

1. Data source & displayData function:modified from Andrew Ng

3. Add Bias term

- The "**bias term**" or "intercept parameter" allows us to move the linear model along the y-axis.
- Complete the lines of code in Section 3.

4. Training the one-vs-all classifier to find parameters using gradient descent

- To implement the one-vs-all classifier, multiple regularized logistic regression classifiers are trained one for each of $\boldsymbol{C}$ classes. Each classifier is trained independently using a " for"- loop from t=1 to $\boldsymbol{C}$. The training of the one-vs-all logistic regression with $\boldsymbol{C}$ classes is described as below.

  For each class t ,where t=1,..,C
  - Train a regularized logistic regression classifier $h_w^{(t)}(x)$ to predict the probability, p(y = $t|x$). The procedures for the regularized logistic regression are explained in the subsection 4.1.
  - To do this, all class t labels as positive samples (y=1) and all other classes as negative samples (y=0).

4.1 procedures for the logistic regression binary classifier

a) Sigmoid function

- The logistic regression hypothesis is defined as a sigmoid function of an input Z where Z is a liner model:

$$h_w^{(t)}(x) = \frac{1}{1+e^{-z}} \text{ , where } Z = \sum_{i=0}^{k} \boldsymbol{w_i x_i}$$

b) Log loss cost function

- The regularized cost function in logistic regression is the log loss function with the regularized term :

$$J^{(t)}(w) = -[\frac{1}{m}\sum_{i=1}^{m} \boldsymbol{y}^{(i)} \cdot \log\left(\boldsymbol{h_w}(\boldsymbol{x}^{(i)})\right) + (\boldsymbol{1} - \boldsymbol{y}^{(i)}) \cdot \log\left(\boldsymbol{1} - \boldsymbol{h_w}(\boldsymbol{x}^{(i)})\right)] + \frac{\lambda}{2m}\sum_{j=1}^{k} \boldsymbol{w_j^2}$$

- You will fit the classifier parameters, w with the training data using the gradient descent method. The objective of the training is to find the parameters, w that minimize the cost function:

$$\underset{w}{\text{Minimize}} \quad J^{(t)}(w)$$

- The gradient decent method will find the optimal weight , w using a " for"- loop from i=1 to iteration. The equations for the weight update is given:

$$\textbf{1) } \Delta \boldsymbol{w_0} = \frac{\boldsymbol{1}}{\boldsymbol{m}}\sum_{\boldsymbol{i=1}}^{\boldsymbol{m}}(\boldsymbol{h_w}(\boldsymbol{x^i}) - \boldsymbol{y^i})$$

1. Data source & displayData function:modified from Andrew Ng

$$w_0 = w_0 - \partial * \Delta w_0$$

2) for each w_j (j=1,…,k)

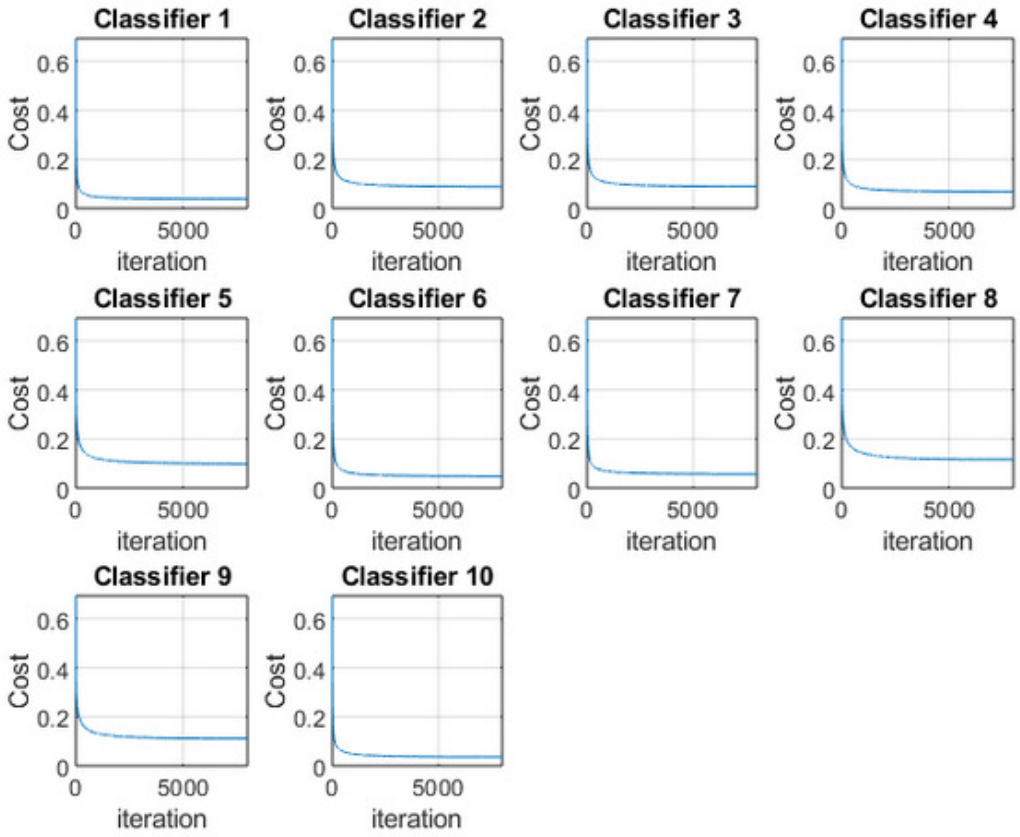
$$\Delta w_j = \frac{1}{m}\sum_{i=1}^{m}(h_w(x^i) - y^i)\, x_j^i + \frac{\lambda}{m} w_j$$

$$w_j = w_j - \partial * \Delta w_j$$

- To implement the regularization term, simply add the term, $\frac{\lambda}{m} w_j$ to the partial derivative of the cost function w.r.t w_j **except** w_0**.**

- At the end of the for loop iteration, you have the final log loss cost and the final parameters, w for the binary classification model for each class t.

5. **Plot the Log Loss cost convergence graph for each classifier.** Generate plots that shows how the cost changes over the iterations for each classifier t. You can use the ***subplot()*** function in the Matlab to display the multiple plots. The plots of the cost versus the iteration look like below:



1. Data source & displayData function:modified from Andrew Ng

- Complete the lines of code to implement **the Log Loss cost convergence graph** in Section 5.

6. The training accuracy of the one-vs-all classifier

- After the training is completed, the one-vs-all classifier can predict the digit contained in a given image. For the given input image, the probability that the input image belongs to each class is computed using the trained logistic regression classifiers. The one-vs-all classifier will pick the class for which the corresponding classifier outputs the highest probability.

- To predict the digits in given images,

  - Calculate the linear model $Z = \sum_{i=0}^{k} w_i x_i$
  - Calculate the sigmoid output of the input Z
  - Pick the class for which the corresponding classifier outputs the highest probability

- Write the lines of code for the calculation of the training accuracy in Section7.

Question 1: what is the accuracy of the developed model on the training data?

7. The one-vs-all classifier evaluation on the test data

- Add the intercept term (Bias term) in the input vector. A new input matrix is generated by concatenating the bias term vector and the new normalized input feature vector.
- Make the predictions on the new testing data using the one-vs-all classifier;

- **Calculate the accuracy of the developed classifier on the test data**

  - Calculate the linear model $Z = \sum_{i=0}^{k} w_i x_i$
  - Calculate the sigmoid output of the input Z
  - Pick the class for which the corresponding classifier outputs the highest probability

- Complete the lines of code to evaluate the multiclass classifier on the test data set
- **Question 2:** what is the accuracy of the multiclass classifier on the test data set?

8. Regularization

- Complex model (has lots of parameters) often prone to **overfitting.** Overfitting can be reduced by imposing a constraint on the overall magnitude of the parameters (weights),w. It introduces a penalty cost term for bringing in more features with the objective function.

- The regularized cost function in logistic regression is the log loss function with the regularized term :

$$J^{(t)}(w) = -[\frac{1}{m}\sum_{i=1}^{m} \boldsymbol{y}^{(i)} \cdot \log\left(\boldsymbol{h_w}(\boldsymbol{x}^{(i)})\right) + (\mathbf{1} - \boldsymbol{y}^{(i)}) \cdot \log\left(\mathbf{1} - \boldsymbol{h_w}(\boldsymbol{x}^{(i)})\right)] + \frac{\lambda}{2m}\sum_{j=1}^{k} \boldsymbol{w_j^2} \ .$$

- **The parameter $\boldsymbol{\lambda}$** controls the trade-off between training well and keeping the parameters small. When the value of λ is large, values of the weights will be small. When the value of λ is small, values of the weights will become large.
- **Train the one-vs-all classifier with different $\boldsymbol{\lambda}$ values**. Repeat Section 4, Section 6, and Section 7 with different λ values when λ =1, λ=5, and λ =10.
- **Question 3:** Find the accuracies of the classifier on the training data and the testing data with the different λ values when λ =1, λ=5, and λ =10.

You can work in groups of up to two people

What to Submit to Blackboard:

- **one** ZIP file that includes :
  1) **A report (15pts): contains answers for Question1 –Question3.**
  2) (85 pts): The live script file, "Lab4.mlx", in that all outputs are generated either inline or on the right side.

- **Pay attention to the zip file name convention:**
  Lab4_Student1 Lastname_Student2 Lastname.zip
  Ex) Lab4_Green_Smith.zip

1. Data source & displayData function:modified from Andrew Ng

Laboratory 4

Multivariable Logistic Regression Classification

Due Date: Beginning of Week 6 Lab

Concepts:

- Multivariable Logistic Regression Classification
- Regularization

Total Points: 100 points

Objectives:

Logistic regression is one of the most popular supervised classification algorithm. The one-vs-all multiclass logistic regression classifier is an extended version of logistic regression to solve multiclass classification problems. Students will implement the one-vs-all logistic regression classifier to distinguish ten digits in the MNIST hand written digit dataset and evaluate performance of the developed model on the test data that is not used during the training.

Files Needed:

- Lab4.mlx
- sigmoid.m
- displayData.m
- MNIST.mat

Assignment: Multiclass classification using regularized logistic regression

- **Multiclass classification** means a classification task with more than two classes. Logistic regression algorithm can also use to solve **multiclass classification problems.** In multiclass classification problems, a training set consisting of data points belonging to N different classes is given, and the goal is to construct a classifier which, given a new data sample, will correctly predict the class to which the new data belongs.

- The one-vs-all logistic regression classifier is implemented by training **multiple logistic regression classifiers,** one for each of the *C* classes in the training dataset. The algorithm turns the multiclass classification problem into **C binary classification problems** (i.e. where we predict only $y \in \{0, 1\}$) by taking values of one class and turn them into positive examples, and the rest of classes - into negatives**.**
- In Lab4, you will implement one-vs-all logistic regression to recognize hand-written digits on the MNIST dataset. Automated handwritten digit recognition is widely used today from

1. Data source & displayData function:modified from Andrew Ng

recognizing postal codes on mail envelopes to recognizing amounts written on bank checks. The MNIST dataset is a set of handwritten digits categorized 0-9 and is available at http://yann.lecun.com/exdb/mnist/.

1. Load the data from a file[1]

- The file "MNIST.mat " contains a training set of 4,000 handwritten digits and a test set of 1,000 handwritten digits. The images have a size of 20 by 20 grayscale images of digits. The 20 by 20 digit image is flattened into a 1 by 400 row vector. Each of these digit images becomes a single row in the data. The data is already normalized in the range of [0, 1] and separated into the training set and the test data.

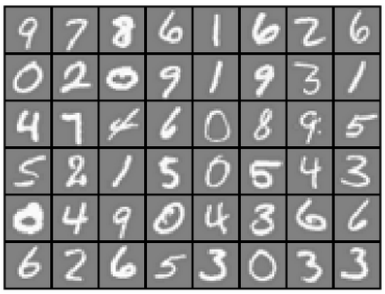
- Using the ***load( )*** function in Matlab, load the "MNIST.mat' file. The .mat format means that the data has been saved from the Matlab workspace. The matrix for the training data set is already named as 'xTrain' and the corresponding labels for the training set is named as 'yTrain', so you do not need to assign names to them. Similarly, the matrices for the test data set and the corresponding labels are names as 'xTest' and ' yTest', respectively.
- Note: There is no zero indexing in the Matlab. So the digits "1" to "9" are labeled as "1" to "9" and the digit "0 " is labeled as "10".

- Complete the lines of code in Section 1.

2. Visualize the data[1]

- Before starting on any prediction model, it is often useful to understand the data by plotting it.
- Display the randomly selected image samples from the training dataset. The **randperm**() function in Matlab returns a row vector containing a random permutation of the integers from 1 to n.
- Generate 48 numbers in the range of [1, 4000] using the **randperm**() function. Then, extract the image samples from the training data using the indexes generated by the **randperm**() function.

- Display the selected image samples by calling the provided function 'dispalyData()'. The plot is similar with the below figure. However the selected digit images will be different because the indexes are selected by the random number generator.



- Complete the lines of code in Section 2.

1. Data source & displayData function:modified from Andrew Ng

3. Add Bias term

- The "**bias term**" or "intercept parameter" allows us to move the linear model along the y-axis.
- Complete the lines of code in Section 3.

4. Training the one-vs-all classifier to find parameters using gradient descent

- To implement the one-vs-all classifier, multiple regularized logistic regression classifiers are trained one for each of $\boldsymbol{C}$ classes. Each classifier is trained independently using a " for"- loop from t=1 to $\boldsymbol{C}$. The training of the one-vs-all logistic regression with $\boldsymbol{C}$ classes is described as below.

  For each class t ,where t=1,..,C
  - Train a regularized logistic regression classifier $h_w^{(t)}(x)$ to predict the probability, p(y = $t|x$). The procedures for the regularized logistic regression are explained in the subsection 4.1.
  - To do this, all class t labels as positive samples (y=1) and all other classes as negative samples (y=0).

4.1 procedures for the logistic regression binary classifier

a) Sigmoid function

- The logistic regression hypothesis is defined as a sigmoid function of an input Z where Z is a liner model:

$$h_w^{(t)}(x) = \frac{1}{1+e^{-z}} \text{ , where } Z = \sum_{i=0}^{k} \boldsymbol{w_i x_i}$$

b) Log loss cost function

- The regularized cost function in logistic regression is the log loss function with the regularized term :

$$J^{(t)}(w) = -[\frac{1}{m}\sum_{i=1}^{m} \boldsymbol{y}^{(i)} \cdot \log\left(\boldsymbol{h_w}(\boldsymbol{x}^{(i)})\right) + (\boldsymbol{1} - \boldsymbol{y}^{(i)}) \cdot \log\left(\boldsymbol{1} - \boldsymbol{h_w}(\boldsymbol{x}^{(i)})\right)] + \frac{\lambda}{2m}\sum_{j=1}^{k} \boldsymbol{w_j^2}$$

- You will fit the classifier parameters, w with the training data using the gradient descent method. The objective of the training is to find the parameters, w that minimize the cost function:

$$\underset{w}{\text{Minimize}} \quad J^{(t)}(w)$$

- The gradient decent method will find the optimal weight , w using a " for"- loop from i=1 to iteration. The equations for the weight update is given:

$$\textbf{1) } \Delta \boldsymbol{w_0} = \frac{\boldsymbol{1}}{\boldsymbol{m}}\sum_{\boldsymbol{i=1}}^{\boldsymbol{m}}(\boldsymbol{h_w}(\boldsymbol{x^i}) - \boldsymbol{y^i})$$

1. Data source & displayData function:modified from Andrew Ng

$$w_0 = w_0 - \partial * \Delta w_0$$

2) for each w_j (j=1,…,k)

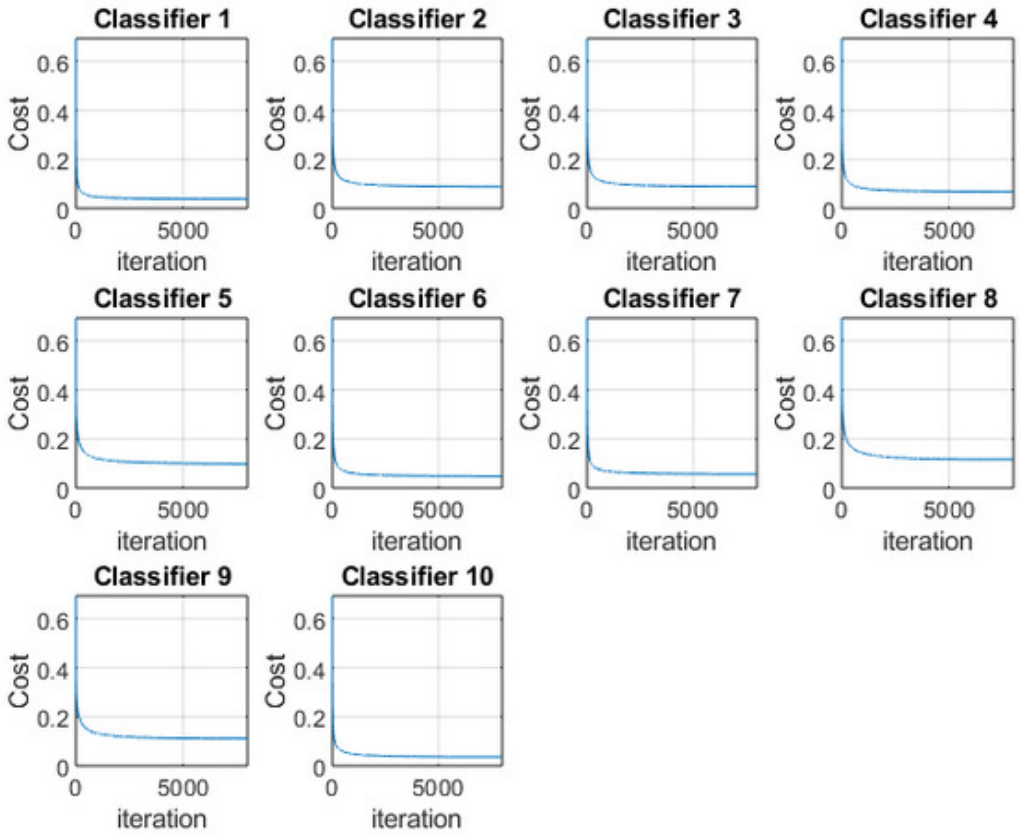
$$\Delta w_j = \frac{1}{m}\sum_{i=1}^{m}(h_w(x^i) - y^i)\, x_j^i + \frac{\lambda}{m} w_j$$

$$w_j = w_j - \partial * \Delta w_j$$

- To implement the regularization term, simply add the term, $\frac{\lambda}{m} w_j$ to the partial derivative of the cost function w.r.t w_j **except** w_0**.**

- At the end of the for loop iteration, you have the final log loss cost and the final parameters, w for the binary classification model for each class t.

5. **Plot the Log Loss cost convergence graph for each classifier.** Generate plots that shows how the cost changes over the iterations for each classifier t. You can use the ***subplot()*** function in the Matlab to display the multiple plots. The plots of the cost versus the iteration look like below:



1. Data source & displayData function:modified from Andrew Ng

- Complete the lines of code to implement **the Log Loss cost convergence graph** in Section 5.

6. The training accuracy of the one-vs-all classifier

- After the training is completed, the one-vs-all classifier can predict the digit contained in a given image. For the given input image, the probability that the input image belongs to each class is computed using the trained logistic regression classifiers. The one-vs-all classifier will pick the class for which the corresponding classifier outputs the highest probability.

- To predict the digits in given images,

  - Calculate the linear model $Z = \sum_{i=0}^{k} w_i x_i$
  - Calculate the sigmoid output of the input Z
  - Pick the class for which the corresponding classifier outputs the highest probability

- Write the lines of code for the calculation of the training accuracy in Section7.

Question 1: what is the accuracy of the developed model on the training data?

7. The one-vs-all classifier evaluation on the test data

- Add the intercept term (Bias term) in the input vector. A new input matrix is generated by concatenating the bias term vector and the new normalized input feature vector.
- Make the predictions on the new testing data using the one-vs-all classifier;

- **Calculate the accuracy of the developed classifier on the test data**

  - Calculate the linear model $Z = \sum_{i=0}^{k} w_i x_i$
  - Calculate the sigmoid output of the input Z
  - Pick the class for which the corresponding classifier outputs the highest probability

- Complete the lines of code to evaluate the multiclass classifier on the test data set
- **Question 2:** what is the accuracy of the multiclass classifier on the test data set?

8. Regularization

- Complex model (has lots of parameters) often prone to **overfitting.** Overfitting can be reduced by imposing a constraint on the overall magnitude of the parameters (weights),w. It introduces a penalty cost term for bringing in more features with the objective function.

- The regularized cost function in logistic regression is the log loss function with the regularized term :

$$J^{(t)}(w) = -[\frac{1}{m}\sum_{i=1}^{m} \boldsymbol{y}^{(i)} \cdot \log\left(\boldsymbol{h_w}(\boldsymbol{x}^{(i)})\right) + (\mathbf{1} - \boldsymbol{y}^{(i)}) \cdot \log\left(\mathbf{1} - \boldsymbol{h_w}(\boldsymbol{x}^{(i)})\right)] + \frac{\lambda}{2m}\sum_{j=1}^{k} \boldsymbol{w_j^2} \ .$$

- **The parameter $\boldsymbol{\lambda}$** controls the trade-off between training well and keeping the parameters small. When the value of λ is large, values of the weights will be small. When the value of λ is small, values of the weights will become large.
- **Train the one-vs-all classifier with different $\boldsymbol{\lambda}$ values**. Repeat Section 4, Section 6, and Section 7 with different λ values when λ =1, λ=5, and λ =10.
- **Question 3:** Find the accuracies of the classifier on the training data and the testing data with the different λ values when λ =1, λ=5, and λ =10.

You can work in groups of up to two people

What to Submit to Blackboard:

- **one** ZIP file that includes :
  1) **A report (15pts): contains answers for Question1 –Question3.**
  2) (85 pts): The live script file, "Lab4.mlx", in that all outputs are generated either inline or on the right side.

- **Pay attention to the zip file name convention:**
  Lab4_Student1 Lastname_Student2 Lastname.zip
  Ex) Lab4_Green_Smith.zip

1. Data source & displayData function:modified from Andrew Ng

Laboratory 4

Multivariable Logistic Regression Classification

Due Date: Beginning of Week 6 Lab

Concepts:

- Multivariable Logistic Regression Classification
- Regularization

Total Points: 100 points

Objectives:

Logistic regression is one of the most popular supervised classification algorithm. The one-vs-all multiclass logistic regression classifier is an extended version of logistic regression to solve multiclass classification problems. Students will implement the one-vs-all logistic regression classifier to distinguish ten digits in the MNIST hand written digit dataset and evaluate performance of the developed model on the test data that is not used during the training.

Files Needed:

- Lab4.mlx
- sigmoid.m
- displayData.m
- MNIST.mat

Assignment: Multiclass classification using regularized logistic regression

- **Multiclass classification** means a classification task with more than two classes. Logistic regression algorithm can also use to solve **multiclass classification problems.** In multiclass classification problems, a training set consisting of data points belonging to N different classes is given, and the goal is to construct a classifier which, given a new data sample, will correctly predict the class to which the new data belongs.

- The one-vs-all logistic regression classifier is implemented by training **multiple logistic regression classifiers,** one for each of the *C* classes in the training dataset. The algorithm turns the multiclass classification problem into **C binary classification problems** (i.e. where we predict only $y \in \{0, 1\}$) by taking values of one class and turn them into positive examples, and the rest of classes - into negatives**.**
- In Lab4, you will implement one-vs-all logistic regression to recognize hand-written digits on the MNIST dataset. Automated handwritten digit recognition is widely used today from

1. Data source & displayData function:modified from Andrew Ng

recognizing postal codes on mail envelopes to recognizing amounts written on bank checks. The MNIST dataset is a set of handwritten digits categorized 0-9 and is available at http://yann.lecun.com/exdb/mnist/.

1. Load the data from a file[1]

- The file "MNIST.mat " contains a training set of 4,000 handwritten digits and a test set of 1,000 handwritten digits. The images have a size of 20 by 20 grayscale images of digits. The 20 by 20 digit image is flattened into a 1 by 400 row vector. Each of these digit images becomes a single row in the data. The data is already normalized in the range of [0, 1] and separated into the training set and the test data.

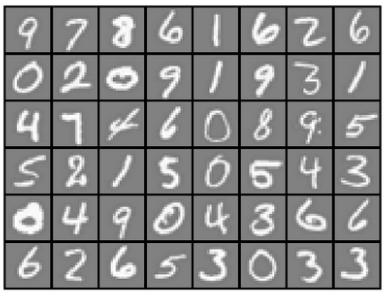
- Using the ***load( )*** function in Matlab, load the "MNIST.mat' file. The .mat format means that the data has been saved from the Matlab workspace. The matrix for the training data set is already named as 'xTrain' and the corresponding labels for the training set is named as 'yTrain', so you do not need to assign names to them. Similarly, the matrices for the test data set and the corresponding labels are names as 'xTest' and ' yTest', respectively.
- Note: There is no zero indexing in the Matlab. So the digits "1" to "9" are labeled as "1" to "9" and the digit "0 " is labeled as "10".

- Complete the lines of code in Section 1.

2. Visualize the data[1]

- Before starting on any prediction model, it is often useful to understand the data by plotting it.
- Display the randomly selected image samples from the training dataset. The **randperm**() function in Matlab returns a row vector containing a random permutation of the integers from 1 to n.
- Generate 48 numbers in the range of [1, 4000] using the **randperm**() function. Then, extract the image samples from the training data using the indexes generated by the **randperm**() function.

- Display the selected image samples by calling the provided function 'dispalyData()'. The plot is similar with the below figure. However the selected digit images will be different because the indexes are selected by the random number generator.



- Complete the lines of code in Section 2.

1. Data source & displayData function:modified from Andrew Ng

3. Add Bias term

- The "**bias term**" or "intercept parameter" allows us to move the linear model along the y-axis.
- Complete the lines of code in Section 3.

4. Training the one-vs-all classifier to find parameters using gradient descent

- To implement the one-vs-all classifier, multiple regularized logistic regression classifiers are trained one for each of $\boldsymbol{C}$ classes. Each classifier is trained independently using a " for"- loop from t=1 to $\boldsymbol{C}$. The training of the one-vs-all logistic regression with $\boldsymbol{C}$ classes is described as below.

  For each class t ,where t=1,..,C
  - Train a regularized logistic regression classifier $h_w^{(t)}(x)$ to predict the probability, p(y = $t|x$). The procedures for the regularized logistic regression are explained in the subsection 4.1.
  - To do this, all class t labels as positive samples (y=1) and all other classes as negative samples (y=0).

4.1 procedures for the logistic regression binary classifier

a) Sigmoid function

- The logistic regression hypothesis is defined as a sigmoid function of an input Z where Z is a liner model:

$$h_w^{(t)}(x) = \frac{1}{1+e^{-z}} \text{ , where } Z = \sum_{i=0}^{k} \boldsymbol{w_i x_i}$$

b) Log loss cost function

- The regularized cost function in logistic regression is the log loss function with the regularized term :

$$J^{(t)}(w) = -[\frac{1}{m}\sum_{i=1}^{m} \boldsymbol{y}^{(i)} \cdot \log\left(\boldsymbol{h_w}(\boldsymbol{x}^{(i)})\right) + (\boldsymbol{1} - \boldsymbol{y}^{(i)}) \cdot \log\left(\boldsymbol{1} - \boldsymbol{h_w}(\boldsymbol{x}^{(i)})\right)] + \frac{\lambda}{2m}\sum_{j=1}^{k} \boldsymbol{w_j^2}$$

- You will fit the classifier parameters, w with the training data using the gradient descent method. The objective of the training is to find the parameters, w that minimize the cost function:

$$\underset{w}{\text{Minimize}} \quad J^{(t)}(w)$$

- The gradient decent method will find the optimal weight , w using a " for"- loop from i=1 to iteration. The equations for the weight update is given:

$$\textbf{1) } \Delta \boldsymbol{w_0} = \frac{\boldsymbol{1}}{\boldsymbol{m}}\sum_{\boldsymbol{i=1}}^{\boldsymbol{m}}(\boldsymbol{h_w}(\boldsymbol{x^i}) - \boldsymbol{y^i})$$

1. Data source & displayData function:modified from Andrew Ng

$$w_0 = w_0 - \partial * \Delta w_0$$

2) for each w_j (j=1,…,k)

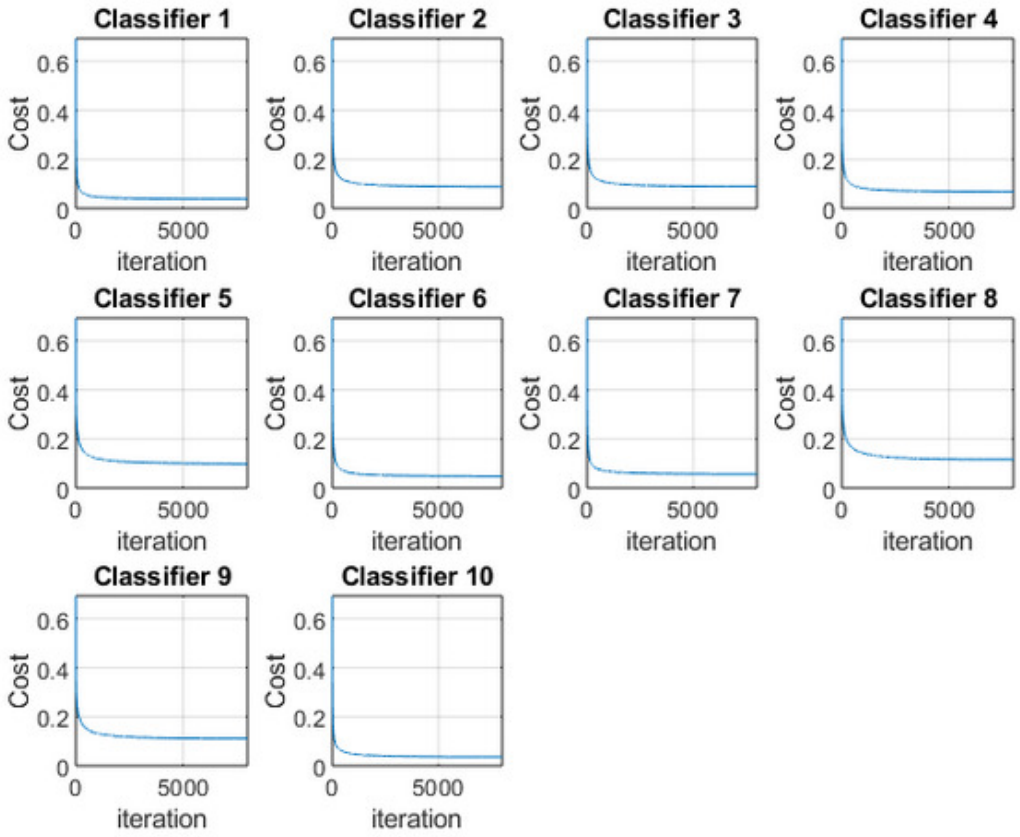
$$\Delta w_j = \frac{1}{m}\sum_{i=1}^{m}(h_w(x^i) - y^i)\, x_j^i + \frac{\lambda}{m} w_j$$

$$w_j = w_j - \partial * \Delta w_j$$

- To implement the regularization term, simply add the term, $\frac{\lambda}{m} w_j$ to the partial derivative of the cost function w.r.t w_j **except** w_0**.**

- At the end of the for loop iteration, you have the final log loss cost and the final parameters, w for the binary classification model for each class t.

5. **Plot the Log Loss cost convergence graph for each classifier.** Generate plots that shows how the cost changes over the iterations for each classifier t. You can use the ***subplot()*** function in the Matlab to display the multiple plots. The plots of the cost versus the iteration look like below:



1. Data source & displayData function:modified from Andrew Ng

- Complete the lines of code to implement **the Log Loss cost convergence graph** in Section 5.

6. The training accuracy of the one-vs-all classifier

- After the training is completed, the one-vs-all classifier can predict the digit contained in a given image. For the given input image, the probability that the input image belongs to each class is computed using the trained logistic regression classifiers. The one-vs-all classifier will pick the class for which the corresponding classifier outputs the highest probability.

- To predict the digits in given images,

  - Calculate the linear model $Z = \sum_{i=0}^{k} w_i x_i$
  - Calculate the sigmoid output of the input Z
  - Pick the class for which the corresponding classifier outputs the highest probability

- Write the lines of code for the calculation of the training accuracy in Section7.

Question 1: what is the accuracy of the developed model on the training data?

7. The one-vs-all classifier evaluation on the test data

- Add the intercept term (Bias term) in the input vector. A new input matrix is generated by concatenating the bias term vector and the new normalized input feature vector.
- Make the predictions on the new testing data using the one-vs-all classifier;

- **Calculate the accuracy of the developed classifier on the test data**

  - Calculate the linear model $Z = \sum_{i=0}^{k} w_i x_i$
  - Calculate the sigmoid output of the input Z
  - Pick the class for which the corresponding classifier outputs the highest probability

- Complete the lines of code to evaluate the multiclass classifier on the test data set
- **Question 2:** what is the accuracy of the multiclass classifier on the test data set?

8. Regularization

- Complex model (has lots of parameters) often prone to **overfitting.** Overfitting can be reduced by imposing a constraint on the overall magnitude of the parameters (weights),w. It introduces a penalty cost term for bringing in more features with the objective function.

- The regularized cost function in logistic regression is the log loss function with the regularized term :

$$J^{(t)}(w) = -[\frac{1}{m}\sum_{i=1}^{m} \boldsymbol{y}^{(i)} \cdot \log\left(\boldsymbol{h_w}(\boldsymbol{x}^{(i)})\right) + (\mathbf{1} - \boldsymbol{y}^{(i)}) \cdot \log\left(\mathbf{1} - \boldsymbol{h_w}(\boldsymbol{x}^{(i)})\right)] + \frac{\lambda}{2m}\sum_{j=1}^{k} \boldsymbol{w_j^2} \ .$$

- **The parameter $\boldsymbol{\lambda}$** controls the trade-off between training well and keeping the parameters small. When the value of λ is large, values of the weights will be small. When the value of λ is small, values of the weights will become large.
- **Train the one-vs-all classifier with different $\boldsymbol{\lambda}$ values**. Repeat Section 4, Section 6, and Section 7 with different λ values when λ =1, λ=5, and λ =10.
- **Question 3:** Find the accuracies of the classifier on the training data and the testing data with the different λ values when λ =1, λ=5, and λ =10.

You can work in groups of up to two people

What to Submit to Blackboard:

- **one** ZIP file that includes :
  1) **A report (15pts): contains answers for Question1 –Question3.**
  2) (85 pts): The live script file, "Lab4.mlx", in that all outputs are generated either inline or on the right side.

- **Pay attention to the zip file name convention:**
  Lab4_Student1 Lastname_Student2 Lastname.zip
  Ex) Lab4_Green_Smith.zip

1. Data source & displayData function:modified from Andrew Ng

Laboratory 4

Multivariable Logistic Regression Classification

Due Date: Beginning of Week 6 Lab

Concepts:

- Multivariable Logistic Regression Classification
- Regularization

Total Points: 100 points

Objectives:

Logistic regression is one of the most popular supervised classification algorithm. The one-vs-all multiclass logistic regression classifier is an extended version of logistic regression to solve multiclass classification problems. Students will implement the one-vs-all logistic regression classifier to distinguish ten digits in the MNIST hand written digit dataset and evaluate performance of the developed model on the test data that is not used during the training.

Files Needed:

- Lab4.mlx
- sigmoid.m
- displayData.m
- MNIST.mat

Assignment: Multiclass classification using regularized logistic regression

- **Multiclass classification** means a classification task with more than two classes. Logistic regression algorithm can also use to solve **multiclass classification problems.** In multiclass classification problems, a training set consisting of data points belonging to N different classes is given, and the goal is to construct a classifier which, given a new data sample, will correctly predict the class to which the new data belongs.

- The one-vs-all logistic regression classifier is implemented by training **multiple logistic regression classifiers,** one for each of the *C* classes in the training dataset. The algorithm turns the multiclass classification problem into **C binary classification problems** (i.e. where we predict only $y \in \{0, 1\}$) by taking values of one class and turn them into positive examples, and the rest of classes - into negatives**.**
- In Lab4, you will implement one-vs-all logistic regression to recognize hand-written digits on the MNIST dataset. Automated handwritten digit recognition is widely used today from

1. Data source & displayData function:modified from Andrew Ng

recognizing postal codes on mail envelopes to recognizing amounts written on bank checks. The MNIST dataset is a set of handwritten digits categorized 0-9 and is available at http://yann.lecun.com/exdb/mnist/.

1. Load the data from a file[1]

- The file "MNIST.mat " contains a training set of 4,000 handwritten digits and a test set of 1,000 handwritten digits. The images have a size of 20 by 20 grayscale images of digits. The 20 by 20 digit image is flattened into a 1 by 400 row vector. Each of these digit images becomes a single row in the data. The data is already normalized in the range of [0, 1] and separated into the training set and the test data.

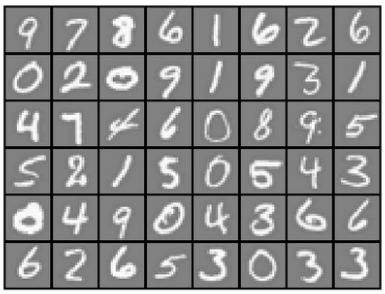
- Using the ***load( )*** function in Matlab, load the "MNIST.mat' file. The .mat format means that the data has been saved from the Matlab workspace. The matrix for the training data set is already named as 'xTrain' and the corresponding labels for the training set is named as 'yTrain', so you do not need to assign names to them. Similarly, the matrices for the test data set and the corresponding labels are names as 'xTest' and ' yTest', respectively.
- Note: There is no zero indexing in the Matlab. So the digits "1" to "9" are labeled as "1" to "9" and the digit "0 " is labeled as "10".

- Complete the lines of code in Section 1.

2. Visualize the data[1]

- Before starting on any prediction model, it is often useful to understand the data by plotting it.
- Display the randomly selected image samples from the training dataset. The **randperm**() function in Matlab returns a row vector containing a random permutation of the integers from 1 to n.
- Generate 48 numbers in the range of [1, 4000] using the **randperm**() function. Then, extract the image samples from the training data using the indexes generated by the **randperm**() function.

- Display the selected image samples by calling the provided function 'dispalyData()'. The plot is similar with the below figure. However the selected digit images will be different because the indexes are selected by the random number generator.



- Complete the lines of code in Section 2.

1. Data source & displayData function:modified from Andrew Ng

3. Add Bias term

- The "**bias term**" or "intercept parameter" allows us to move the linear model along the y-axis.
- Complete the lines of code in Section 3.

4. Training the one-vs-all classifier to find parameters using gradient descent

- To implement the one-vs-all classifier, multiple regularized logistic regression classifiers are trained one for each of $\boldsymbol{C}$ classes. Each classifier is trained independently using a " for"- loop from t=1 to $\boldsymbol{C}$. The training of the one-vs-all logistic regression with $\boldsymbol{C}$ classes is described as below.

  For each class t ,where t=1,..,C
  - Train a regularized logistic regression classifier $h_w^{(t)}(x)$ to predict the probability, p(y = $t|x$). The procedures for the regularized logistic regression are explained in the subsection 4.1.
  - To do this, all class t labels as positive samples (y=1) and all other classes as negative samples (y=0).

4.1 procedures for the logistic regression binary classifier

a) Sigmoid function

- The logistic regression hypothesis is defined as a sigmoid function of an input Z where Z is a liner model:

$$h_w^{(t)}(x) = \frac{1}{1+e^{-z}} \text{ , where } Z = \sum_{i=0}^{k} \boldsymbol{w_i x_i}$$

b) Log loss cost function

- The regularized cost function in logistic regression is the log loss function with the regularized term :

$$J^{(t)}(w) = -[\frac{1}{m}\sum_{i=1}^{m} \boldsymbol{y}^{(i)} \cdot \log\left(\boldsymbol{h_w}(\boldsymbol{x}^{(i)})\right) + (\boldsymbol{1} - \boldsymbol{y}^{(i)}) \cdot \log\left(\boldsymbol{1} - \boldsymbol{h_w}(\boldsymbol{x}^{(i)})\right)] + \frac{\lambda}{2m}\sum_{j=1}^{k} \boldsymbol{w_j^2}$$

- You will fit the classifier parameters, w with the training data using the gradient descent method. The objective of the training is to find the parameters, w that minimize the cost function:

$$\underset{w}{\text{Minimize}} \quad J^{(t)}(w)$$

- The gradient decent method will find the optimal weight , w using a " for"- loop from i=1 to iteration. The equations for the weight update is given:

$$\textbf{1) } \Delta \boldsymbol{w_0} = \frac{\boldsymbol{1}}{\boldsymbol{m}}\sum_{\boldsymbol{i=1}}^{\boldsymbol{m}}(\boldsymbol{h_w}(\boldsymbol{x^i}) - \boldsymbol{y^i})$$

1. Data source & displayData function:modified from Andrew Ng

$$w_0 = w_0 - \partial * \Delta w_0$$

2) for each w_j (j=1,…,k)

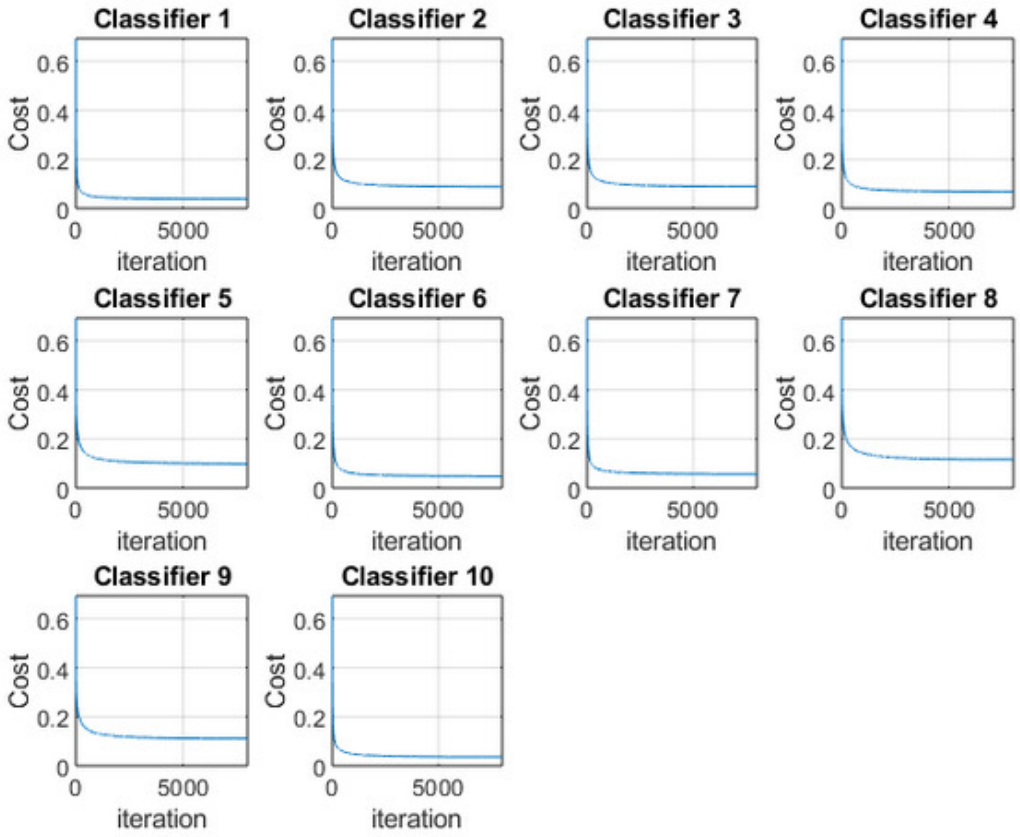
$$\Delta w_j = \frac{1}{m}\sum_{i=1}^{m}(h_w(x^i) - y^i)\, x_j^i + \frac{\lambda}{m} w_j$$

$$w_j = w_j - \partial * \Delta w_j$$

- To implement the regularization term, simply add the term, $\frac{\lambda}{m} w_j$ to the partial derivative of the cost function w.r.t w_j **except** w_0**.**

- At the end of the for loop iteration, you have the final log loss cost and the final parameters, w for the binary classification model for each class t.

5. **Plot the Log Loss cost convergence graph for each classifier.** Generate plots that shows how the cost changes over the iterations for each classifier t. You can use the ***subplot()*** function in the Matlab to display the multiple plots. The plots of the cost versus the iteration look like below:



1. Data source & displayData function:modified from Andrew Ng

- Complete the lines of code to implement **the Log Loss cost convergence graph** in Section 5.

6. The training accuracy of the one-vs-all classifier

- After the training is completed, the one-vs-all classifier can predict the digit contained in a given image. For the given input image, the probability that the input image belongs to each class is computed using the trained logistic regression classifiers. The one-vs-all classifier will pick the class for which the corresponding classifier outputs the highest probability.

- To predict the digits in given images,

  - Calculate the linear model $Z = \sum_{i=0}^{k} w_i x_i$
  - Calculate the sigmoid output of the input Z
  - Pick the class for which the corresponding classifier outputs the highest probability

- Write the lines of code for the calculation of the training accuracy in Section7.

Question 1: what is the accuracy of the developed model on the training data?

7. The one-vs-all classifier evaluation on the test data

- Add the intercept term (Bias term) in the input vector. A new input matrix is generated by concatenating the bias term vector and the new normalized input feature vector.
- Make the predictions on the new testing data using the one-vs-all classifier;

- **Calculate the accuracy of the developed classifier on the test data**

  - Calculate the linear model $Z = \sum_{i=0}^{k} w_i x_i$
  - Calculate the sigmoid output of the input Z
  - Pick the class for which the corresponding classifier outputs the highest probability

- Complete the lines of code to evaluate the multiclass classifier on the test data set
- **Question 2:** what is the accuracy of the multiclass classifier on the test data set?

8. Regularization

- Complex model (has lots of parameters) often prone to **overfitting.** Overfitting can be reduced by imposing a constraint on the overall magnitude of the parameters (weights),w. It introduces a penalty cost term for bringing in more features with the objective function.

- The regularized cost function in logistic regression is the log loss function with the regularized term :

$$J^{(t)}(w) = -[\frac{1}{m}\sum_{i=1}^{m} \boldsymbol{y}^{(i)} \cdot \log\left(\boldsymbol{h_w}(\boldsymbol{x}^{(i)})\right) + (\mathbf{1} - \boldsymbol{y}^{(i)}) \cdot \log\left(\mathbf{1} - \boldsymbol{h_w}(\boldsymbol{x}^{(i)})\right)] + \frac{\lambda}{2m}\sum_{j=1}^{k} \boldsymbol{w_j^2} \ .$$

- **The parameter $\boldsymbol{\lambda}$** controls the trade-off between training well and keeping the parameters small. When the value of λ is large, values of the weights will be small. When the value of λ is small, values of the weights will become large.
- **Train the one-vs-all classifier with different $\boldsymbol{\lambda}$ values**. Repeat Section 4, Section 6, and Section 7 with different λ values when λ =1, λ=5, and λ =10.
- **Question 3:** Find the accuracies of the classifier on the training data and the testing data with the different λ values when λ =1, λ=5, and λ =10.

You can work in groups of up to two people

What to Submit to Blackboard:

- **one** ZIP file that includes :
  1) **A report (15pts): contains answers for Question1 –Question3.**
  2) (85 pts): The live script file, "Lab4.mlx", in that all outputs are generated either inline or on the right side.

- **Pay attention to the zip file name convention:**
  Lab4_Student1 Lastname_Student2 Lastname.zip
  Ex) Lab4_Green_Smith.zip

1. Data source & displayData function:modified from Andrew Ng

Laboratory 4

Multivariable Logistic Regression Classification

Due Date: Beginning of Week 6 Lab

Concepts:

- Multivariable Logistic Regression Classification
- Regularization

Total Points: 100 points

Objectives:

Logistic regression is one of the most popular supervised classification algorithm. The one-vs-all multiclass logistic regression classifier is an extended version of logistic regression to solve multiclass classification problems. Students will implement the one-vs-all logistic regression classifier to distinguish ten digits in the MNIST hand written digit dataset and evaluate performance of the developed model on the test data that is not used during the training.

Files Needed:

- Lab4.mlx
- sigmoid.m
- displayData.m
- MNIST.mat

Assignment: Multiclass classification using regularized logistic regression

- **Multiclass classification** means a classification task with more than two classes. Logistic regression algorithm can also use to solve **multiclass classification problems.** In multiclass classification problems, a training set consisting of data points belonging to N different classes is given, and the goal is to construct a classifier which, given a new data sample, will correctly predict the class to which the new data belongs.

- The one-vs-all logistic regression classifier is implemented by training **multiple logistic regression classifiers,** one for each of the *C* classes in the training dataset. The algorithm turns the multiclass classification problem into **C binary classification problems** (i.e. where we predict only $y \in \{0, 1\}$) by taking values of one class and turn them into positive examples, and the rest of classes - into negatives**.**
- In Lab4, you will implement one-vs-all logistic regression to recognize hand-written digits on the MNIST dataset. Automated handwritten digit recognition is widely used today from

1. Data source & displayData function:modified from Andrew Ng

recognizing postal codes on mail envelopes to recognizing amounts written on bank checks. The MNIST dataset is a set of handwritten digits categorized 0-9 and is available at http://yann.lecun.com/exdb/mnist/.

1. Load the data from a file[1]

- The file "MNIST.mat " contains a training set of 4,000 handwritten digits and a test set of 1,000 handwritten digits. The images have a size of 20 by 20 grayscale images of digits. The 20 by 20 digit image is flattened into a 1 by 400 row vector. Each of these digit images becomes a single row in the data. The data is already normalized in the range of [0, 1] and separated into the training set and the test data.

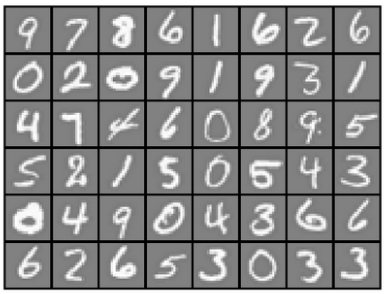
- Using the ***load( )*** function in Matlab, load the "MNIST.mat' file. The .mat format means that the data has been saved from the Matlab workspace. The matrix for the training data set is already named as 'xTrain' and the corresponding labels for the training set is named as 'yTrain', so you do not need to assign names to them. Similarly, the matrices for the test data set and the corresponding labels are names as 'xTest' and ' yTest', respectively.
- Note: There is no zero indexing in the Matlab. So the digits "1" to "9" are labeled as "1" to "9" and the digit "0 " is labeled as "10".

- Complete the lines of code in Section 1.

2. Visualize the data[1]

- Before starting on any prediction model, it is often useful to understand the data by plotting it.
- Display the randomly selected image samples from the training dataset. The **randperm**() function in Matlab returns a row vector containing a random permutation of the integers from 1 to n.
- Generate 48 numbers in the range of [1, 4000] using the **randperm**() function. Then, extract the image samples from the training data using the indexes generated by the **randperm**() function.

- Display the selected image samples by calling the provided function 'dispalyData()'. The plot is similar with the below figure. However the selected digit images will be different because the indexes are selected by the random number generator.



- Complete the lines of code in Section 2.

1. Data source & displayData function:modified from Andrew Ng

3. Add Bias term

- The "**bias term**" or "intercept parameter" allows us to move the linear model along the y-axis.
- Complete the lines of code in Section 3.

4. Training the one-vs-all classifier to find parameters using gradient descent

- To implement the one-vs-all classifier, multiple regularized logistic regression classifiers are trained one for each of $\boldsymbol{C}$ classes. Each classifier is trained independently using a " for"- loop from t=1 to $\boldsymbol{C}$. The training of the one-vs-all logistic regression with $\boldsymbol{C}$ classes is described as below.

  For each class t ,where t=1,..,C
  - Train a regularized logistic regression classifier $h_w^{(t)}(x)$ to predict the probability, p(y = $t|x$). The procedures for the regularized logistic regression are explained in the subsection 4.1.
  - To do this, all class t labels as positive samples (y=1) and all other classes as negative samples (y=0).

4.1 procedures for the logistic regression binary classifier

a) Sigmoid function

- The logistic regression hypothesis is defined as a sigmoid function of an input Z where Z is a liner model:

$$h_w^{(t)}(x) = \frac{1}{1+e^{-z}} \text{ , where } Z = \sum_{i=0}^{k} \boldsymbol{w_i x_i}$$

b) Log loss cost function

- The regularized cost function in logistic regression is the log loss function with the regularized term :

$$J^{(t)}(w) = -[\frac{1}{m}\sum_{i=1}^{m} \boldsymbol{y}^{(i)} \cdot \log\left(\boldsymbol{h_w}(\boldsymbol{x}^{(i)})\right) + (\boldsymbol{1} - \boldsymbol{y}^{(i)}) \cdot \log\left(\boldsymbol{1} - \boldsymbol{h_w}(\boldsymbol{x}^{(i)})\right)] + \frac{\lambda}{2m}\sum_{j=1}^{k} \boldsymbol{w_j^2}$$

- You will fit the classifier parameters, w with the training data using the gradient descent method. The objective of the training is to find the parameters, w that minimize the cost function:

$$\underset{w}{\text{Minimize}} \quad J^{(t)}(w)$$

- The gradient decent method will find the optimal weight , w using a " for"- loop from i=1 to iteration. The equations for the weight update is given:

$$\textbf{1) } \Delta \boldsymbol{w_0} = \frac{\boldsymbol{1}}{\boldsymbol{m}}\sum_{\boldsymbol{i=1}}^{\boldsymbol{m}}(\boldsymbol{h_w}(\boldsymbol{x^i}) - \boldsymbol{y^i})$$

1. Data source & displayData function:modified from Andrew Ng

$$w_0 = w_0 - \partial * \Delta w_0$$

2) for each w_j (j=1,…,k)

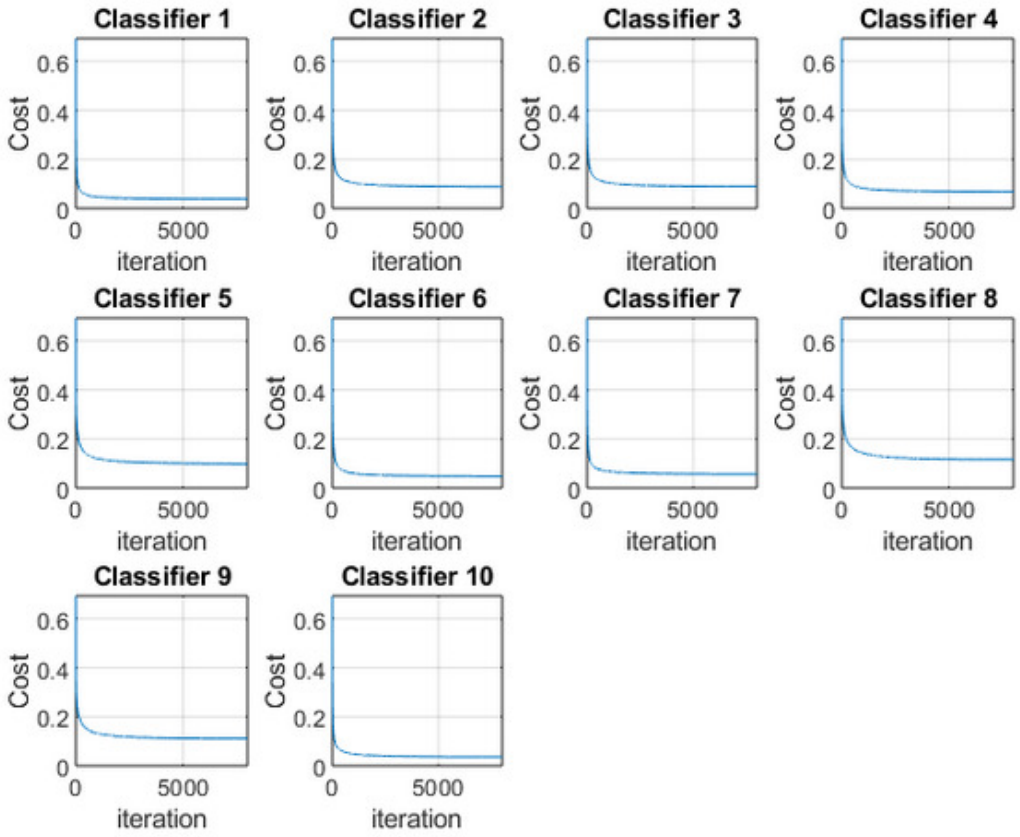
$$\Delta w_j = \frac{1}{m}\sum_{i=1}^{m}(h_w(x^i) - y^i)\, x_j^i + \frac{\lambda}{m} w_j$$

$$w_j = w_j - \partial * \Delta w_j$$

- To implement the regularization term, simply add the term, $\frac{\lambda}{m} w_j$ to the partial derivative of the cost function w.r.t w_j **except** w_0**.**

- At the end of the for loop iteration, you have the final log loss cost and the final parameters, w for the binary classification model for each class t.

5. **Plot the Log Loss cost convergence graph for each classifier.** Generate plots that shows how the cost changes over the iterations for each classifier t. You can use the ***subplot()*** function in the Matlab to display the multiple plots. The plots of the cost versus the iteration look like below:



1. Data source & displayData function:modified from Andrew Ng

- Complete the lines of code to implement **the Log Loss cost convergence graph** in Section 5.

6. The training accuracy of the one-vs-all classifier

- After the training is completed, the one-vs-all classifier can predict the digit contained in a given image. For the given input image, the probability that the input image belongs to each class is computed using the trained logistic regression classifiers. The one-vs-all classifier will pick the class for which the corresponding classifier outputs the highest probability.

- To predict the digits in given images,

  - Calculate the linear model $Z = \sum_{i=0}^{k} w_i x_i$
  - Calculate the sigmoid output of the input Z
  - Pick the class for which the corresponding classifier outputs the highest probability

- Write the lines of code for the calculation of the training accuracy in Section7.

Question 1: what is the accuracy of the developed model on the training data?

7. The one-vs-all classifier evaluation on the test data

- Add the intercept term (Bias term) in the input vector. A new input matrix is generated by concatenating the bias term vector and the new normalized input feature vector.
- Make the predictions on the new testing data using the one-vs-all classifier;

- **Calculate the accuracy of the developed classifier on the test data**

  - Calculate the linear model $Z = \sum_{i=0}^{k} w_i x_i$
  - Calculate the sigmoid output of the input Z
  - Pick the class for which the corresponding classifier outputs the highest probability

- Complete the lines of code to evaluate the multiclass classifier on the test data set
- **Question 2:** what is the accuracy of the multiclass classifier on the test data set?

8. Regularization

- Complex model (has lots of parameters) often prone to **overfitting.** Overfitting can be reduced by imposing a constraint on the overall magnitude of the parameters (weights),w. It introduces a penalty cost term for bringing in more features with the objective function.

- The regularized cost function in logistic regression is the log loss function with the regularized term :

$$J^{(t)}(w) = -[\frac{1}{m}\sum_{i=1}^{m} \boldsymbol{y}^{(i)} \cdot \log\left(\boldsymbol{h_w}(\boldsymbol{x}^{(i)})\right) + (\mathbf{1} - \boldsymbol{y}^{(i)}) \cdot \log\left(\mathbf{1} - \boldsymbol{h_w}(\boldsymbol{x}^{(i)})\right)] + \frac{\lambda}{2m}\sum_{j=1}^{k} \boldsymbol{w_j^2} \ .$$

- **The parameter $\boldsymbol{\lambda}$** controls the trade-off between training well and keeping the parameters small. When the value of λ is large, values of the weights will be small. When the value of λ is small, values of the weights will become large.
- **Train the one-vs-all classifier with different $\boldsymbol{\lambda}$ values**. Repeat Section 4, Section 6, and Section 7 with different λ values when λ =1, λ=5, and λ =10.
- **Question 3:** Find the accuracies of the classifier on the training data and the testing data with the different λ values when λ =1, λ=5, and λ =10.

You can work in groups of up to two people

What to Submit to Blackboard:

- **one** ZIP file that includes :
  1) **A report (15pts): contains answers for Question1 –Question3.**
  2) (85 pts): The live script file, "Lab4.mlx", in that all outputs are generated either inline or on the right side.

- **Pay attention to the zip file name convention:**
  Lab4_Student1 Lastname_Student2 Lastname.zip
  Ex) Lab4_Green_Smith.zip

1. Data source & displayData function:modified from Andrew Ng

Laboratory 4

Multivariable Logistic Regression Classification

Due Date: Beginning of Week 6 Lab

Concepts:

- Multivariable Logistic Regression Classification
- Regularization

Total Points: 100 points

Objectives:

Logistic regression is one of the most popular supervised classification algorithm. The one-vs-all multiclass logistic regression classifier is an extended version of logistic regression to solve multiclass classification problems. Students will implement the one-vs-all logistic regression classifier to distinguish ten digits in the MNIST hand written digit dataset and evaluate performance of the developed model on the test data that is not used during the training.

Files Needed:

- Lab4.mlx
- sigmoid.m
- displayData.m
- MNIST.mat

Assignment: Multiclass classification using regularized logistic regression

- **Multiclass classification** means a classification task with more than two classes. Logistic regression algorithm can also use to solve **multiclass classification problems.** In multiclass classification problems, a training set consisting of data points belonging to N different classes is given, and the goal is to construct a classifier which, given a new data sample, will correctly predict the class to which the new data belongs.

- The one-vs-all logistic regression classifier is implemented by training **multiple logistic regression classifiers,** one for each of the *C* classes in the training dataset. The algorithm turns the multiclass classification problem into **C binary classification problems** (i.e. where we predict only $y \in \{0, 1\}$) by taking values of one class and turn them into positive examples, and the rest of classes - into negatives**.**
- In Lab4, you will implement one-vs-all logistic regression to recognize hand-written digits on the MNIST dataset. Automated handwritten digit recognition is widely used today from

1. Data source & displayData function:modified from Andrew Ng

recognizing postal codes on mail envelopes to recognizing amounts written on bank checks. The MNIST dataset is a set of handwritten digits categorized 0-9 and is available at http://yann.lecun.com/exdb/mnist/.

1. Load the data from a file[1]

- The file "MNIST.mat " contains a training set of 4,000 handwritten digits and a test set of 1,000 handwritten digits. The images have a size of 20 by 20 grayscale images of digits. The 20 by 20 digit image is flattened into a 1 by 400 row vector. Each of these digit images becomes a single row in the data. The data is already normalized in the range of [0, 1] and separated into the training set and the test data.

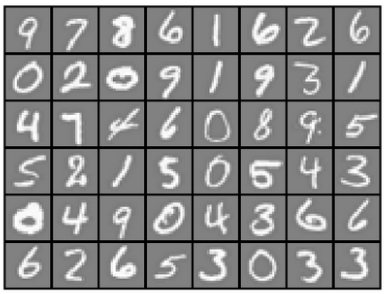
- Using the ***load( )*** function in Matlab, load the "MNIST.mat' file. The .mat format means that the data has been saved from the Matlab workspace. The matrix for the training data set is already named as 'xTrain' and the corresponding labels for the training set is named as 'yTrain', so you do not need to assign names to them. Similarly, the matrices for the test data set and the corresponding labels are names as 'xTest' and ' yTest', respectively.
- Note: There is no zero indexing in the Matlab. So the digits "1" to "9" are labeled as "1" to "9" and the digit "0 " is labeled as "10".

- Complete the lines of code in Section 1.

2. Visualize the data[1]

- Before starting on any prediction model, it is often useful to understand the data by plotting it.
- Display the randomly selected image samples from the training dataset. The **randperm**() function in Matlab returns a row vector containing a random permutation of the integers from 1 to n.
- Generate 48 numbers in the range of [1, 4000] using the **randperm**() function. Then, extract the image samples from the training data using the indexes generated by the **randperm**() function.

- Display the selected image samples by calling the provided function 'dispalyData()'. The plot is similar with the below figure. However the selected digit images will be different because the indexes are selected by the random number generator.



- Complete the lines of code in Section 2.

1. Data source & displayData function:modified from Andrew Ng

3. Add Bias term

- The "**bias term**" or "intercept parameter" allows us to move the linear model along the y-axis.
- Complete the lines of code in Section 3.

4. Training the one-vs-all classifier to find parameters using gradient descent

- To implement the one-vs-all classifier, multiple regularized logistic regression classifiers are trained one for each of $\boldsymbol{C}$ classes. Each classifier is trained independently using a " for"- loop from t=1 to $\boldsymbol{C}$. The training of the one-vs-all logistic regression with $\boldsymbol{C}$ classes is described as below.

  For each class t ,where t=1,..,C
  - Train a regularized logistic regression classifier $h_w^{(t)}(x)$ to predict the probability, p(y = $t|x$). The procedures for the regularized logistic regression are explained in the subsection 4.1.
  - To do this, all class t labels as positive samples (y=1) and all other classes as negative samples (y=0).

4.1 procedures for the logistic regression binary classifier

a) Sigmoid function

- The logistic regression hypothesis is defined as a sigmoid function of an input Z where Z is a liner model:

$$h_w^{(t)}(x) = \frac{1}{1+e^{-z}} \text{ , where } Z = \sum_{i=0}^{k} \boldsymbol{w_i x_i}$$

b) Log loss cost function

- The regularized cost function in logistic regression is the log loss function with the regularized term :

$$J^{(t)}(w) = -[\frac{1}{m}\sum_{i=1}^{m} \boldsymbol{y}^{(i)} \cdot \log\left(\boldsymbol{h_w}(\boldsymbol{x}^{(i)})\right) + (\boldsymbol{1} - \boldsymbol{y}^{(i)}) \cdot \log\left(\boldsymbol{1} - \boldsymbol{h_w}(\boldsymbol{x}^{(i)})\right)] + \frac{\lambda}{2m}\sum_{j=1}^{k} \boldsymbol{w_j^2}$$

- You will fit the classifier parameters, w with the training data using the gradient descent method. The objective of the training is to find the parameters, w that minimize the cost function:

$$\underset{w}{\text{Minimize}} \quad J^{(t)}(w)$$

- The gradient decent method will find the optimal weight , w using a " for"- loop from i=1 to iteration. The equations for the weight update is given:

$$\textbf{1) } \Delta \boldsymbol{w_0} = \frac{\boldsymbol{1}}{\boldsymbol{m}}\sum_{\boldsymbol{i=1}}^{\boldsymbol{m}}(\boldsymbol{h_w}(\boldsymbol{x^i}) - \boldsymbol{y^i})$$

1. Data source & displayData function:modified from Andrew Ng

$$w_0 = w_0 - \partial * \Delta w_0$$

2) for each w_j (j=1,…,k)

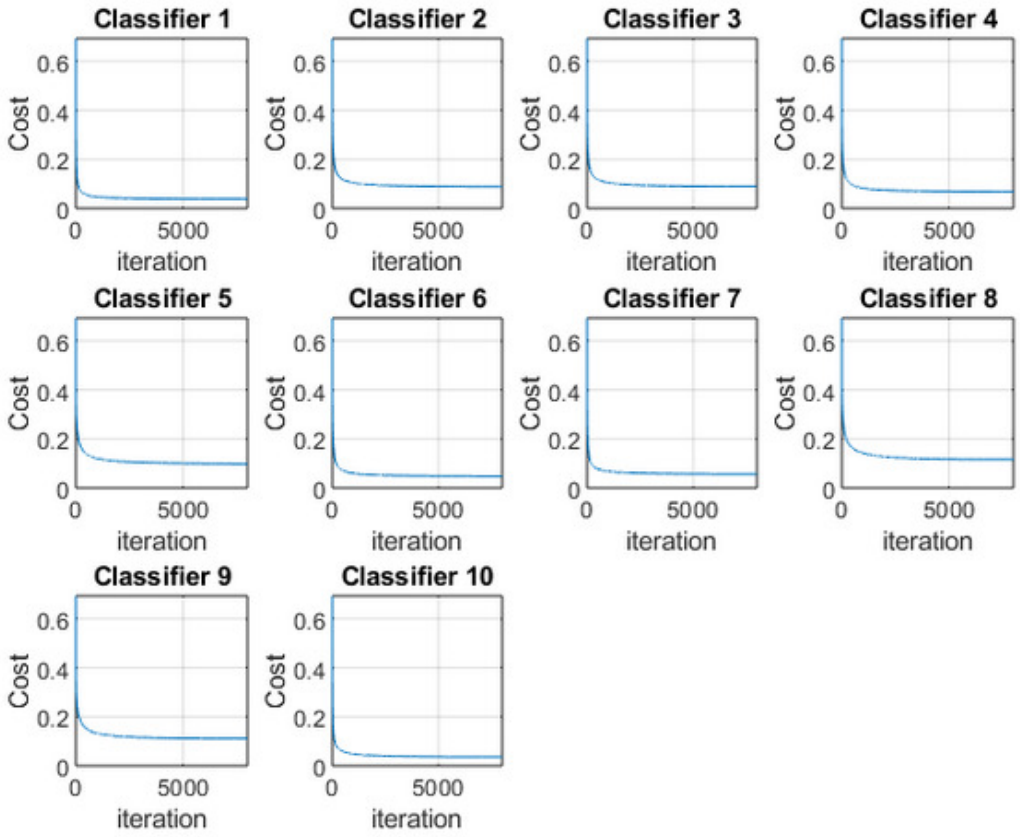
$$\Delta w_j = \frac{1}{m}\sum_{i=1}^{m}(h_w(x^i) - y^i)\, x_j^i + \frac{\lambda}{m} w_j$$

$$w_j = w_j - \partial * \Delta w_j$$

- To implement the regularization term, simply add the term, $\frac{\lambda}{m} w_j$ to the partial derivative of the cost function w.r.t w_j **except** w_0**.**

- At the end of the for loop iteration, you have the final log loss cost and the final parameters, w for the binary classification model for each class t.

5. **Plot the Log Loss cost convergence graph for each classifier.** Generate plots that shows how the cost changes over the iterations for each classifier t. You can use the ***subplot()*** function in the Matlab to display the multiple plots. The plots of the cost versus the iteration look like below:



1. Data source & displayData function:modified from Andrew Ng

- Complete the lines of code to implement **the Log Loss cost convergence graph** in Section 5.

6. The training accuracy of the one-vs-all classifier

- After the training is completed, the one-vs-all classifier can predict the digit contained in a given image. For the given input image, the probability that the input image belongs to each class is computed using the trained logistic regression classifiers. The one-vs-all classifier will pick the class for which the corresponding classifier outputs the highest probability.

- To predict the digits in given images,

  - Calculate the linear model $Z = \sum_{i=0}^{k} w_i x_i$
  - Calculate the sigmoid output of the input Z
  - Pick the class for which the corresponding classifier outputs the highest probability

- Write the lines of code for the calculation of the training accuracy in Section7.

Question 1: what is the accuracy of the developed model on the training data?

7. The one-vs-all classifier evaluation on the test data

- Add the intercept term (Bias term) in the input vector. A new input matrix is generated by concatenating the bias term vector and the new normalized input feature vector.
- Make the predictions on the new testing data using the one-vs-all classifier;

- **Calculate the accuracy of the developed classifier on the test data**

  - Calculate the linear model $Z = \sum_{i=0}^{k} w_i x_i$
  - Calculate the sigmoid output of the input Z
  - Pick the class for which the corresponding classifier outputs the highest probability

- Complete the lines of code to evaluate the multiclass classifier on the test data set
- **Question 2:** what is the accuracy of the multiclass classifier on the test data set?

8. Regularization

- Complex model (has lots of parameters) often prone to **overfitting.** Overfitting can be reduced by imposing a constraint on the overall magnitude of the parameters (weights),w. It introduces a penalty cost term for bringing in more features with the objective function.

- The regularized cost function in logistic regression is the log loss function with the regularized term :

$$J^{(t)}(w) = -[\frac{1}{m}\sum_{i=1}^{m} \boldsymbol{y}^{(i)} \cdot \log\left(\boldsymbol{h_w}(\boldsymbol{x}^{(i)})\right) + (\mathbf{1} - \boldsymbol{y}^{(i)}) \cdot \log\left(\mathbf{1} - \boldsymbol{h_w}(\boldsymbol{x}^{(i)})\right)] + \frac{\lambda}{2m}\sum_{j=1}^{k} \boldsymbol{w_j^2} \ .$$

- **The parameter $\boldsymbol{\lambda}$** controls the trade-off between training well and keeping the parameters small. When the value of λ is large, values of the weights will be small. When the value of λ is small, values of the weights will become large.
- **Train the one-vs-all classifier with different $\boldsymbol{\lambda}$ values**. Repeat Section 4, Section 6, and Section 7 with different λ values when λ =1, λ=5, and λ =10.
- **Question 3:** Find the accuracies of the classifier on the training data and the testing data with the different λ values when λ =1, λ=5, and λ =10.

You can work in groups of up to two people

What to Submit to Blackboard:

- **one** ZIP file that includes :
  1) **A report (15pts): contains answers for Question1 –Question3.**
  2) (85 pts): The live script file, "Lab4.mlx", in that all outputs are generated either inline or on the right side.

- **Pay attention to the zip file name convention:**
  Lab4_Student1 Lastname_Student2 Lastname.zip
  Ex) Lab4_Green_Smith.zip

1. Data source & displayData function:modified from Andrew Ng

Laboratory 4

Multivariable Logistic Regression Classification

Due Date: Beginning of Week 6 Lab

Concepts:

- Multivariable Logistic Regression Classification
- Regularization

Total Points: 100 points

Objectives:

Logistic regression is one of the most popular supervised classification algorithm. The one-vs-all multiclass logistic regression classifier is an extended version of logistic regression to solve multiclass classification problems. Students will implement the one-vs-all logistic regression classifier to distinguish ten digits in the MNIST hand written digit dataset and evaluate performance of the developed model on the test data that is not used during the training.

Files Needed:

- Lab4.mlx
- sigmoid.m
- displayData.m
- MNIST.mat

Assignment: Multiclass classification using regularized logistic regression

- **Multiclass classification** means a classification task with more than two classes. Logistic regression algorithm can also use to solve **multiclass classification problems.** In multiclass classification problems, a training set consisting of data points belonging to N different classes is given, and the goal is to construct a classifier which, given a new data sample, will correctly predict the class to which the new data belongs.

- The one-vs-all logistic regression classifier is implemented by training **multiple logistic regression classifiers,** one for each of the *C* classes in the training dataset. The algorithm turns the multiclass classification problem into **C binary classification problems** (i.e. where we predict only $y \in \{0, 1\}$) by taking values of one class and turn them into positive examples, and the rest of classes - into negatives**.**
- In Lab4, you will implement one-vs-all logistic regression to recognize hand-written digits on the MNIST dataset. Automated handwritten digit recognition is widely used today from

1. Data source & displayData function:modified from Andrew Ng

recognizing postal codes on mail envelopes to recognizing amounts written on bank checks. The MNIST dataset is a set of handwritten digits categorized 0-9 and is available at http://yann.lecun.com/exdb/mnist/.

1. Load the data from a file[1]

- The file "MNIST.mat " contains a training set of 4,000 handwritten digits and a test set of 1,000 handwritten digits. The images have a size of 20 by 20 grayscale images of digits. The 20 by 20 digit image is flattened into a 1 by 400 row vector. Each of these digit images becomes a single row in the data. The data is already normalized in the range of [0, 1] and separated into the training set and the test data.

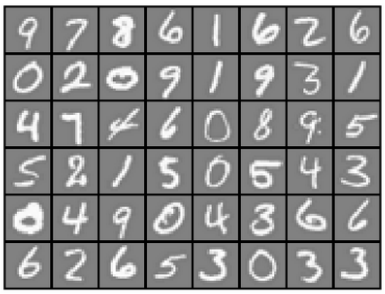
- Using the ***load( )*** function in Matlab, load the "MNIST.mat' file. The .mat format means that the data has been saved from the Matlab workspace. The matrix for the training data set is already named as 'xTrain' and the corresponding labels for the training set is named as 'yTrain', so you do not need to assign names to them. Similarly, the matrices for the test data set and the corresponding labels are names as 'xTest' and ' yTest', respectively.
- Note: There is no zero indexing in the Matlab. So the digits "1" to "9" are labeled as "1" to "9" and the digit "0 " is labeled as "10".

- Complete the lines of code in Section 1.

2. Visualize the data[1]

- Before starting on any prediction model, it is often useful to understand the data by plotting it.
- Display the randomly selected image samples from the training dataset. The **randperm**() function in Matlab returns a row vector containing a random permutation of the integers from 1 to n.
- Generate 48 numbers in the range of [1, 4000] using the **randperm**() function. Then, extract the image samples from the training data using the indexes generated by the **randperm**() function.

- Display the selected image samples by calling the provided function 'dispalyData()'. The plot is similar with the below figure. However the selected digit images will be different because the indexes are selected by the random number generator.



- Complete the lines of code in Section 2.

1. Data source & displayData function:modified from Andrew Ng

3. Add Bias term

- The "**bias term**" or "intercept parameter" allows us to move the linear model along the y-axis.
- Complete the lines of code in Section 3.

4. Training the one-vs-all classifier to find parameters using gradient descent

- To implement the one-vs-all classifier, multiple regularized logistic regression classifiers are trained one for each of $\boldsymbol{C}$ classes. Each classifier is trained independently using a " for"- loop from t=1 to $\boldsymbol{C}$. The training of the one-vs-all logistic regression with $\boldsymbol{C}$ classes is described as below.

  For each class t ,where t=1,..,C
  - Train a regularized logistic regression classifier $h_w^{(t)}(x)$ to predict the probability, p(y = $t|x$). The procedures for the regularized logistic regression are explained in the subsection 4.1.
  - To do this, all class t labels as positive samples (y=1) and all other classes as negative samples (y=0).

4.1 procedures for the logistic regression binary classifier

a) Sigmoid function

- The logistic regression hypothesis is defined as a sigmoid function of an input Z where Z is a liner model:

$$h_w^{(t)}(x) = \frac{1}{1+e^{-z}} \text{ , where } Z = \sum_{i=0}^{k} \boldsymbol{w_i x_i}$$

b) Log loss cost function

- The regularized cost function in logistic regression is the log loss function with the regularized term :

$$J^{(t)}(w) = -[\frac{1}{m}\sum_{i=1}^{m} \boldsymbol{y}^{(i)} \cdot \log\left(\boldsymbol{h_w}(\boldsymbol{x}^{(i)})\right) + (\boldsymbol{1} - \boldsymbol{y}^{(i)}) \cdot \log\left(\boldsymbol{1} - \boldsymbol{h_w}(\boldsymbol{x}^{(i)})\right)] + \frac{\lambda}{2m}\sum_{j=1}^{k} \boldsymbol{w_j^2}$$

- You will fit the classifier parameters, w with the training data using the gradient descent method. The objective of the training is to find the parameters, w that minimize the cost function:

$$\underset{w}{\text{Minimize}} \quad J^{(t)}(w)$$

- The gradient decent method will find the optimal weight , w using a " for"- loop from i=1 to iteration. The equations for the weight update is given:

$$\textbf{1) } \Delta \boldsymbol{w_0} = \frac{\boldsymbol{1}}{\boldsymbol{m}}\sum_{\boldsymbol{i=1}}^{\boldsymbol{m}}(\boldsymbol{h_w}(\boldsymbol{x^i}) - \boldsymbol{y^i})$$

1. Data source & displayData function:modified from Andrew Ng

$$w_0 = w_0 - \partial * \Delta w_0$$

2) for each w_j (j=1,…,k)

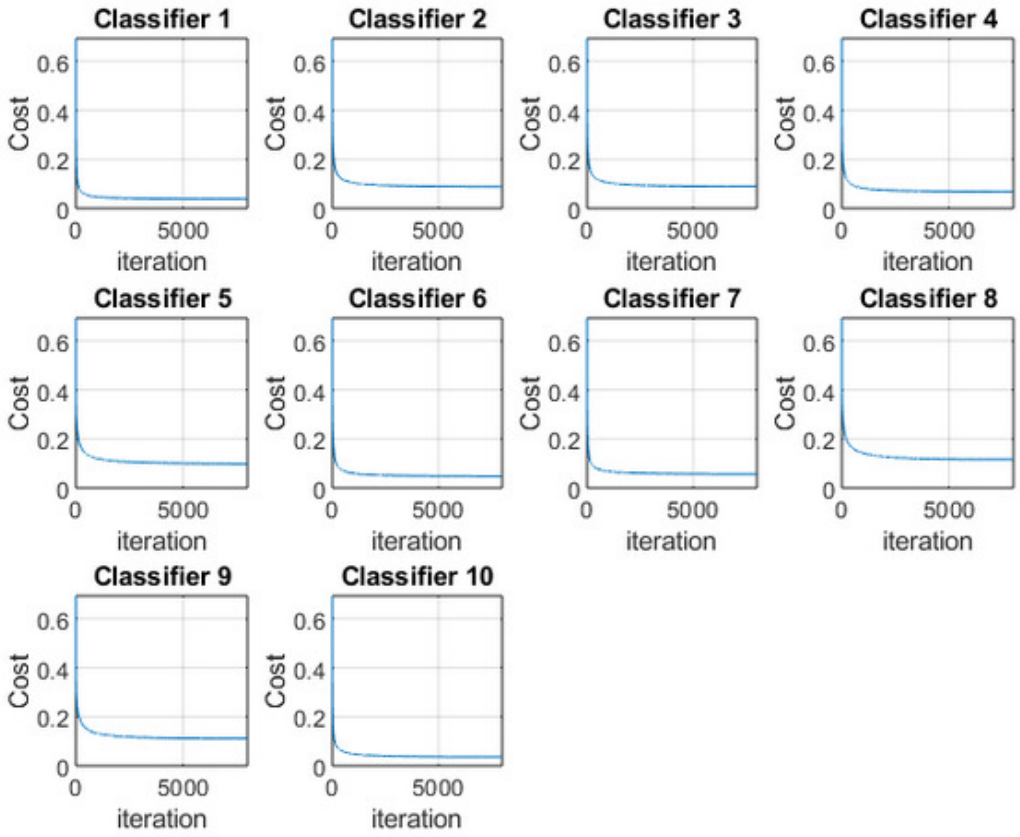
$$\Delta w_j = \frac{1}{m}\sum_{i=1}^{m}(h_w(x^i) - y^i)\, x_j^i + \frac{\lambda}{m} w_j$$

$$w_j = w_j - \partial * \Delta w_j$$

- To implement the regularization term, simply add the term, $\frac{\lambda}{m} w_j$ to the partial derivative of the cost function w.r.t w_j **except** w_0**.**

- At the end of the for loop iteration, you have the final log loss cost and the final parameters, w for the binary classification model for each class t.

5. **Plot the Log Loss cost convergence graph for each classifier.** Generate plots that shows how the cost changes over the iterations for each classifier t. You can use the ***subplot()*** function in the Matlab to display the multiple plots. The plots of the cost versus the iteration look like below:



1. Data source & displayData function:modified from Andrew Ng

- Complete the lines of code to implement **the Log Loss cost convergence graph** in Section 5.

6. The training accuracy of the one-vs-all classifier

- After the training is completed, the one-vs-all classifier can predict the digit contained in a given image. For the given input image, the probability that the input image belongs to each class is computed using the trained logistic regression classifiers. The one-vs-all classifier will pick the class for which the corresponding classifier outputs the highest probability.

- To predict the digits in given images,

  - Calculate the linear model $Z = \sum_{i=0}^{k} w_i x_i$
  - Calculate the sigmoid output of the input Z
  - Pick the class for which the corresponding classifier outputs the highest probability

- Write the lines of code for the calculation of the training accuracy in Section7.

Question 1: what is the accuracy of the developed model on the training data?

7. The one-vs-all classifier evaluation on the test data

- Add the intercept term (Bias term) in the input vector. A new input matrix is generated by concatenating the bias term vector and the new normalized input feature vector.
- Make the predictions on the new testing data using the one-vs-all classifier;

- **Calculate the accuracy of the developed classifier on the test data**

  - Calculate the linear model $Z = \sum_{i=0}^{k} w_i x_i$
  - Calculate the sigmoid output of the input Z
  - Pick the class for which the corresponding classifier outputs the highest probability

- Complete the lines of code to evaluate the multiclass classifier on the test data set
- **Question 2:** what is the accuracy of the multiclass classifier on the test data set?

8. Regularization

- Complex model (has lots of parameters) often prone to **overfitting.** Overfitting can be reduced by imposing a constraint on the overall magnitude of the parameters (weights),w. It introduces a penalty cost term for bringing in more features with the objective function.

- The regularized cost function in logistic regression is the log loss function with the regularized term :

$$J^{(t)}(w) = -[\frac{1}{m}\sum_{i=1}^{m} \boldsymbol{y}^{(i)} \cdot \log\left(\boldsymbol{h_w}(\boldsymbol{x}^{(i)})\right) + (\mathbf{1} - \boldsymbol{y}^{(i)}) \cdot \log\left(\mathbf{1} - \boldsymbol{h_w}(\boldsymbol{x}^{(i)})\right)] + \frac{\lambda}{2m}\sum_{j=1}^{k} \boldsymbol{w_j^2} \ .$$

- **The parameter $\boldsymbol{\lambda}$** controls the trade-off between training well and keeping the parameters small. When the value of λ is large, values of the weights will be small. When the value of λ is small, values of the weights will become large.
- **Train the one-vs-all classifier with different $\boldsymbol{\lambda}$ values**. Repeat Section 4, Section 6, and Section 7 with different λ values when λ =1, λ=5, and λ =10.
- **Question 3:** Find the accuracies of the classifier on the training data and the testing data with the different λ values when λ =1, λ=5, and λ =10.

You can work in groups of up to two people

What to Submit to Blackboard:

- **one** ZIP file that includes :
  1) **A report (15pts): contains answers for Question1 –Question3.**
  2) (85 pts): The live script file, "Lab4.mlx", in that all outputs are generated either inline or on the right side.

- **Pay attention to the zip file name convention:**
  Lab4_Student1 Lastname_Student2 Lastname.zip
  Ex) Lab4_Green_Smith.zip

1. Data source & displayData function:modified from Andrew Ng