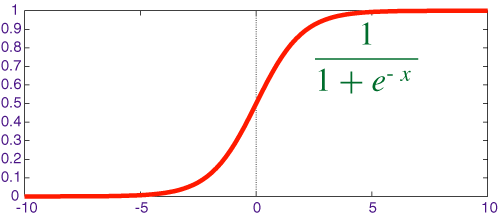


**Summary**: Logistic regression is used in a classification problem. For example: determine whether a tumor is benign or malignant. The output of the classifier is a value 0 or 1 (for a binary classification problem). That means:

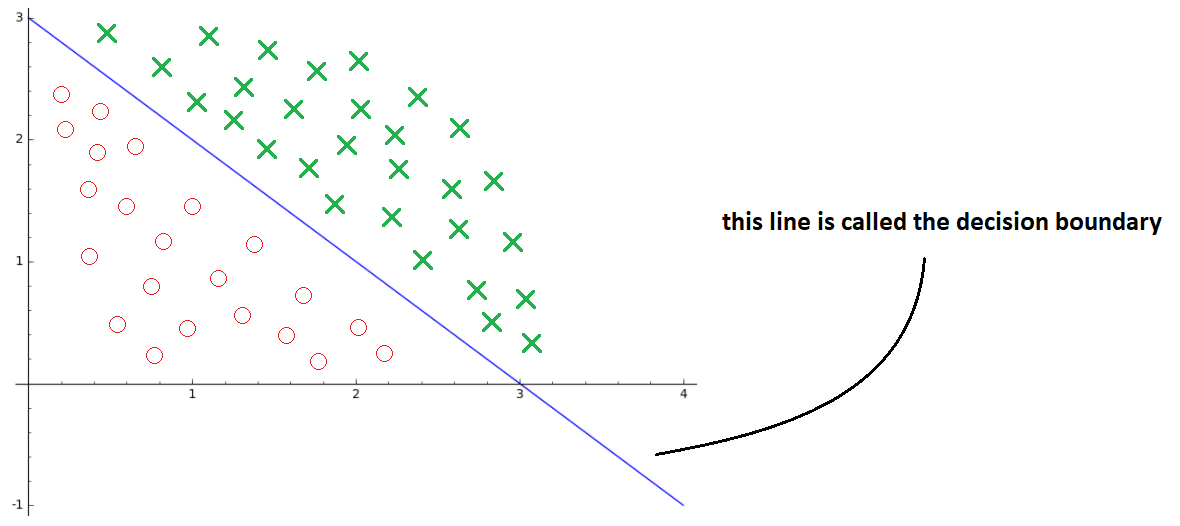
If, we call this a multi-class classification problem. In classification problem, the output should be either 0 or 1 (for binary classification problem). It should not output value.

1. **Notation**
   * : number of training examples
   * : number of features
   * : input variable
   * : output variable
   * : input features of ith training example
   * : value of feature j in ith training example
   * : output of ith training example
   * (: ith training example
2. **Hypothesis**
   * The function is called a sigmoid function.

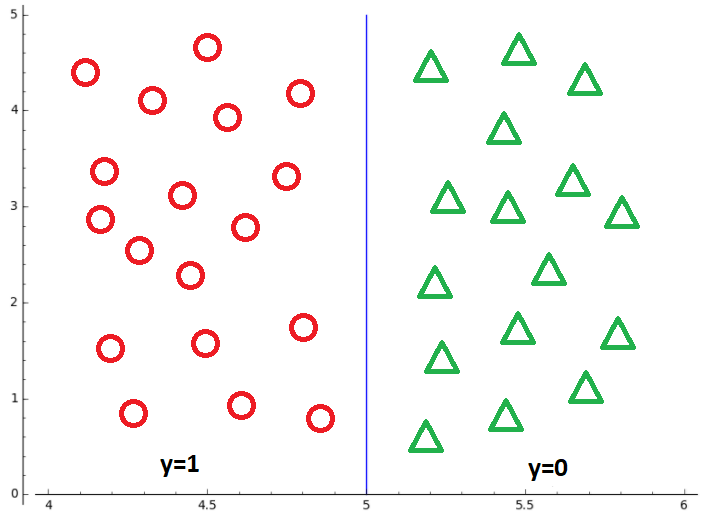


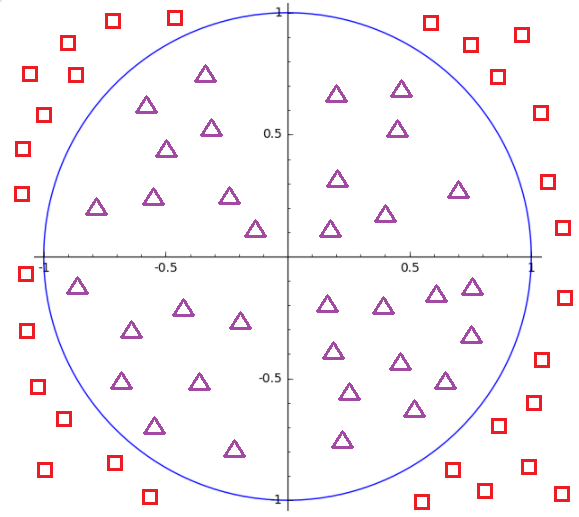
* + **Interpretation of the output of the hypothesis function:**
    - is the estimated probability that on input x
    - Example:
      * If
      * It tells the patient that there is a 70% chance the tumor is malignant
    - 🡪 The probability that, given, parameterized by.

1. **Decision Boundary**
   * Consider this graph below:

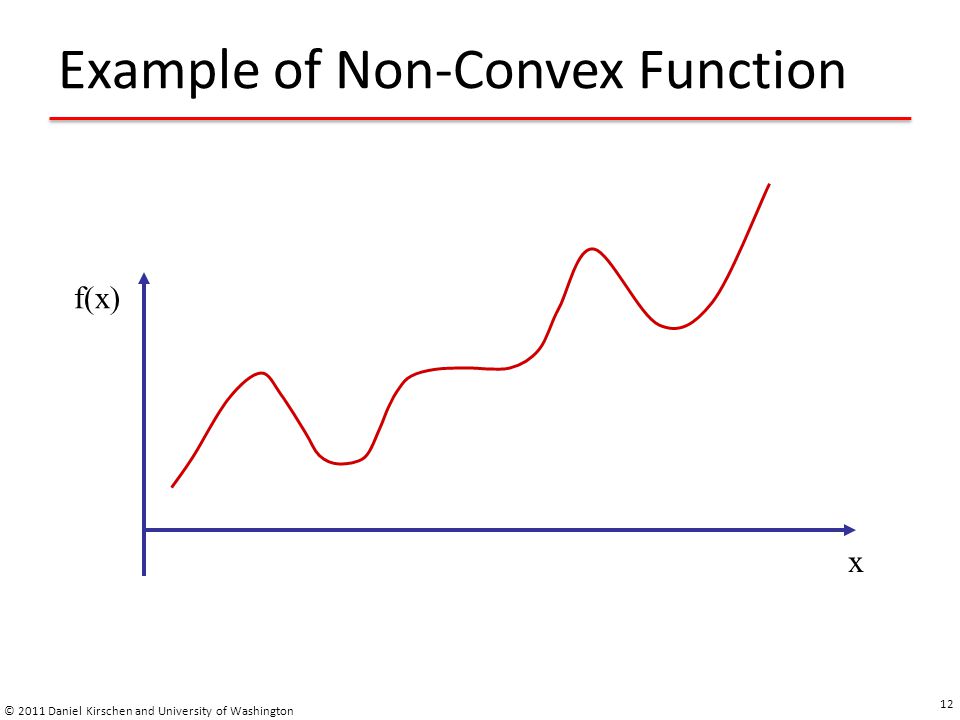


* + We have the decision boundary of the form and the hypothesis. That means this graphs is predicting “ if
  + Some other examples:





1. **Cost Function**
   * Given a training set: with examples and:
   * Consider the cost function in linear regression:
   * If we apply the cost function from simple linear regression to logistic regression, it would result in a non-convex function.



* + Therefore, we need to come up with a new cost function.

|  |  |
| --- | --- |
|  | When :   * As |
|  | When :  As |

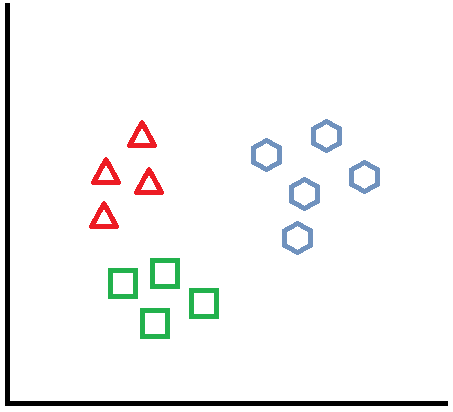
* + We can re-write the function above as following:
  + To fit parameters , we need to minimize
  + To make a prediction on a new input x:

1. **Gradient Descent Algorithm**
   * We have:
   * We then have the gradient descent algorithm as follow:

Repeat {

}

1. **Vectorized Implementation**
   * A strategy to optimize gradient descent:
2. **Optimization Algorithm**
   * Some available optimization algorithms:
     + Gradient descent
     + Conjugate gradient
     + BFGS
     + L-BFGS
   * Advantages of the last 3 algorithms:
     + No need to manually pick learning rate . They automatically pick the best learning rate for each iteration
     + Often faster than gradient descent
   * Disadvantages:
     + More complex
3. **Multiclass Classification**
   * Examples:
     + Email foldering/tagging: work, friends, family, etc
     + Weather: sunny, cloudy, rain, snow



* + One strategy to solve a multiclass classification problem is One-vs-all (one-vs-rest) classification. In the example above, we can set:
    - Class 1: triangle
    - Class 2: hexagon
    - Class 3: square
  + Then break the problem above into 3 smaller binary classification problems

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |

* + In one-vs-all classification, we train logistic regression classifiers for each class to predict the probability that
  + On new input , to make a prediction, pick the class that maximizes the probability
  + For a logistic regression problem with classes, using one-vs-all classification, we’ll need to break it into binary classification problems.

1. **Problem of Overfitting**
   * Underfitting: poor performance on the training data, high bias, and poor performance on test data
   * Overfitting: fit the training data very well but has a poor performance on the test set, which fail to generalize new examples. Overfitting has a high variance.
   * There are 2 options to solve overfitting:
     + Reduce the number of features (manually or automatically)
     + Regularization
       - Keep all the features, but reduce the magnitude/values of the parameters
       - Work well when we have lots of features, each of which contributes a bit to the prediction of
2. **Regularization – Cost Function**
   * The ideas of regularization:
     + Penalize the parameters (make the values small)
     + 🡪 Get a “simpler” hypothesis
     + 🡪 Less prone to overfitting
   * If is set too large, we’ll have
     + 🡪
     + 🡪 We’ll encounter underfitting problem
   * Note:
     + We only penalize (Since is set to be 1 by default)
3. **Regularized Linear Regression**
   * Consider the regularized cost function that we want to minimize
   * Gradient descent for linear regression is written as follows:



* + Consequently, we can re-write the gradient descent algorithm for regularized linear regression as follows:



* + Notes:
    - 🡪 which means this term will shrink
  + Consider normal equation method for linear regression, applying regularization to this method, we’ll have:
  + Regularization will take care of the non-invertibility issue in normal equation method, making the matrix become an invertible.

1. **Regularized Logistic Regression**
   * Consider the cost function for logistic regression in section 4 together with the regularized cost function in section 10:
   * We can come up with a regularized cost function for logistic regression as follows:
   * The gradient descent algorithm for regularized logistic regression can be written as follows:



* + Note: the algorithm looks identical to the gradient descent algorithm for regularized linear regression, however, the hypothesis for logistic regression is not the same.

1. **Mathematical Interpretation of The Partial Derivative of The Cost Function**