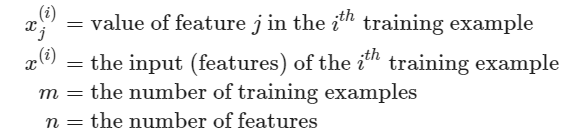
总结报告9

1. **学习内容：机器学习 WEEK2，WEEK3**

**WEEK2**

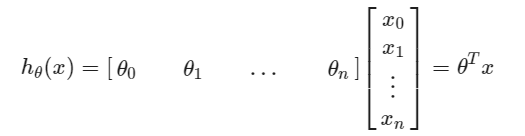
1. **Multivariate linear regression**: linear regression with multiple variables
   1. **Notation：**



* 1. **The multivariable form of the hypothesis function：**



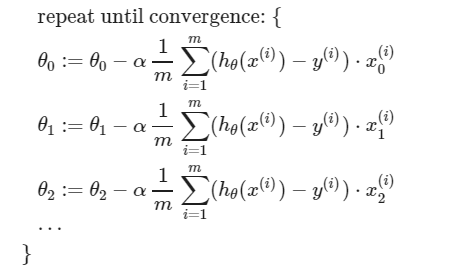
**Matrix form:**



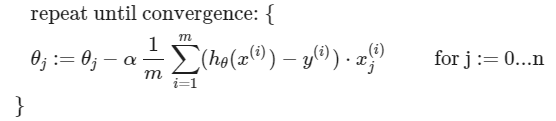
**Notice:**  for convenience reasons in this course, assume

1. **Gradient Descent for Multiple Variables**

The gradient descent equation itself is generally the same form; we just have to repeat it for our 'n' features:



**In other words,**



**More information in next image:**

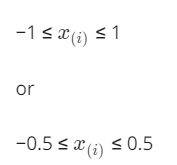


1. **Gradient Descent in Practice I - Feature Scaling (to faster Gradient Descent)**

**① IDEA:** make sure features are on a similar scale

We can **speed up gradient descent** by having each of our input values **in roughly the same range**. This is because θ will descend quickly on small ranges and slowly on large ranges, and so will oscillate inefficiently（无效振荡）down to the optimum when the variables are very uneven(不均匀).

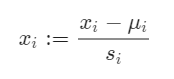
**② Ideally:** get every feature into **approximately a  range**



③ Two techniques to help with this are **feature scaling** and **mean normalization**.

**Feature scaling** involves dividing the input values by the range (i.e. the maximum value minus the minimum value) of the input variable, resulting in a new range of just 1.

**Mean normalization(均值归一化)** involves subtracting the average value for an input variable from the values for that input variable resulting in a new average value for the input variable of just zero.



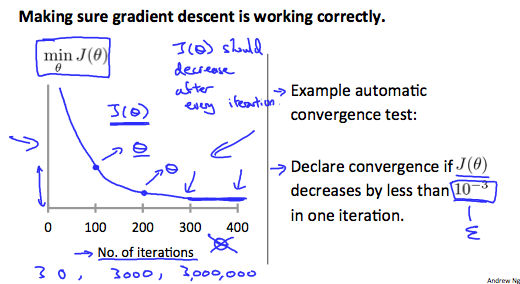
Where *μi* is the **average** of all the values for feature (i) and is the range of values (max - min), or​ is the standard deviation.

1. **Gradient Descent in Practice II - Learning Rate**
   1. **Debugging gradient descent.**

Make a plot with **number of iterations** on the x-axis. Now plot the cost function, J(θ) over the number of iterations of gradient descent. If J(θ) ever increases, then probably need to decrease α.

* 1. **Automatic convergence test.**

Declare convergence if J(θ) decreases by less than  in one iteration, where is some small value such as 10−3.In practice it's difficult to choose this threshold value.



1. **Features and Polynomial Regression**
   1. **Housing prediction**

****

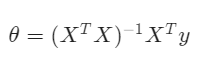
* 1. **Polynomial Regression**

****

1. **Normal Equation：**method to solve for  analytically
   1. **Idea**

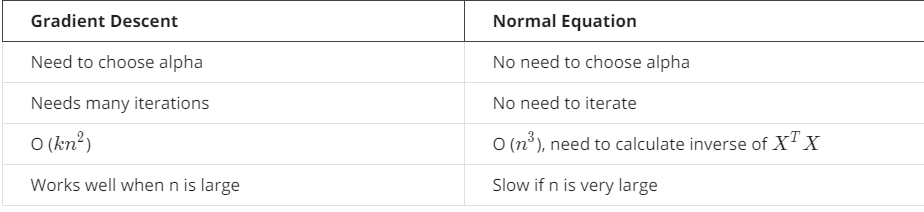
In the "Normal Equation" method, we will minimize  by explicitly taking its **derivatives** with respect to the  ’s, and **setting them to zero**.

**② The formula:**



Notice: There is **no need** to do feature scaling with the normal equation.

③ A **comparison** of gradient descent and the normal equation:



**7.** **Normal Equation Non-invertibility**

The 'pinv' function will give you a value of *θ* even if  is not invertible.

1. **Causes:** non-invertible

Ⅰ.Redundant features, where two features are very closely related (i.e. they are linearly dependent)

Ⅱ.Too many features (e.g. m ≤ n). In this case, delete some features or use "regularization".

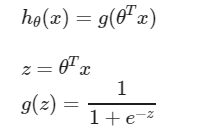
1. **Solutions：**

Ⅰ.deleting a feature that is linearly dependent with another.

Ⅱ.deleting one or more features when there are too many features.

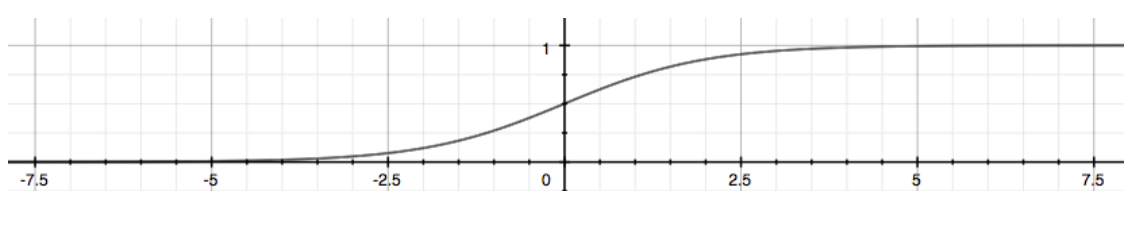
**WEEK 3**

1. **Classification（logistic regression）and representation**
2. **want** 0 ≤ *hθ​(x)* ≤ 1



****

1. **g（z）sigmoid function:**



**Notice:** maps any real number to the (0, 1) interval, making it useful for transforming an arbitrary-valued function into a function better suited for classification

1. **Interpretation of Hypothesis Output**

*hθ​(x)* estimated the **probability** that output is 1, probability that prediction is 0 is just the complement of our probability that it is 1.

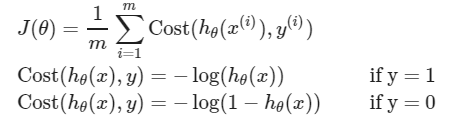
For example, *hθ​(x)*=0.7 gives us a probability of 70% that our output is, then the probability that it is 0 is 30%.

1. **Logistic Regression Model**
   1. **Cost function:**

**Linear Regression: **

**Define: **

* 1. **Logistic Regression:（用原来的Cost Function会出现non-convex）**



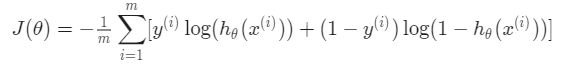
**Notice：**y=0 or y=1 always (Classification)

* 1. **Simplified Cost Function and Gradient Descent:**

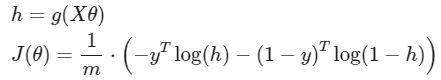
合并上述Cost 分段函数，有：



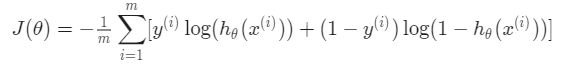
Then, Cost Function is:



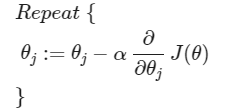
A **vectorized** implementation is:



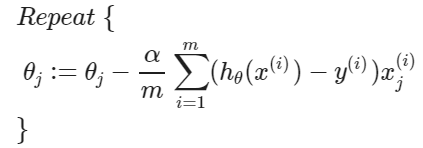
* 1. **Gradient Descent:**



Want 



**由于**，得到

（Simultaneously update all ）

1. **Solving the problem of Overfitting**
   1. 
   2. **Underfitting (high bias)：**maps poorly to the trend of data.

**Overfitting (high variance):** a hypothesis function that fits the available data but does not generalize well to predict new data.

**③ Address the issue of overfitting:**

**1) Reduce the number of features:**

·Manually（手动地）select which features to keep.

·Use a model selection algorithm (later in the course).

**2) Regularization：**

·Keep all the features, but reduce the magnitude/values of parameters .

·Regularization works well when we have a lot of slightly useful features.

**④ Cost Function:**



**Notice:** λ, is the **regularization parameter**. It determines how much the costs of our theta parameters are inflated.

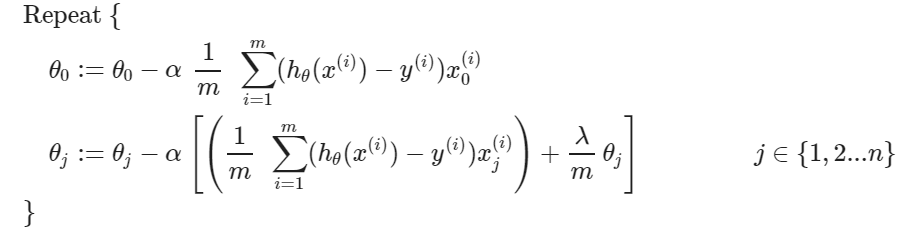
If λ is very large, maybe underfitting.

1. **Regularized Linear Regression**
   1. **Two learning algorithms of linear regression:**

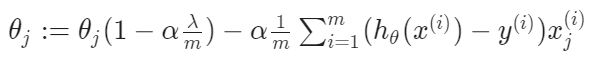




* 1. **Gradient Descent:**

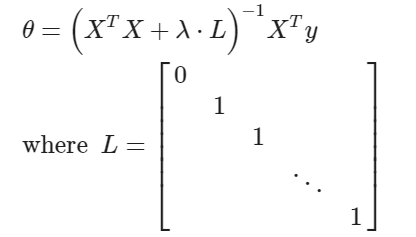


其中，整理第二行，



Notice:  is a little bit less than 1.

* 1. **Normal Equation:**



(n+1)×(n+1)

If m < n, then  is non-invertible. However, when adding the term λ⋅ L, then  + λ⋅ L becomes invertible.

1. **Regularized Logistic Regression**
   1. **Cost Function：**



* 1. **Gradient Descent:**

