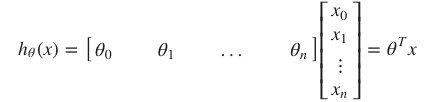
Multiple Features

The multivariable form of the hypothesis function accommodating these multiple features is as follows:

hθ(x)=θ0+θ1x1+θ2x2+θ3x3+⋯+θnxn

In order to develop intuition about this function, we can think about *θ*0​ as the basic price of a house, *θ*1​ as the price per square meter, *θ*2​ as the price per floor, etc. *x*1​ will be the number of square meters in the house, *x*2​ the number of floors, etc.

Using the definition of matrix multiplication, our multivariable hypothesis function can be concisely represented as:

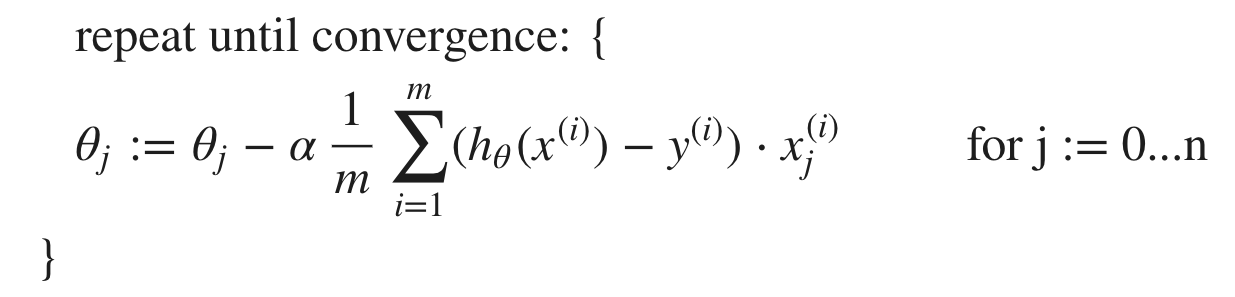


**Gradient Descent for Multiple Variables**

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

A close up of text on a white background

Description automatically generated



# Gradient Descent in Practice I - Feature Scaling

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 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. To implement both of these techniques, adjust your input values as shown in this formula:

A picture containing object

Description automatically generated

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

A screenshot of a cell phone

Description automatically generated

### **Polynomial Regression**

Our hypothesis function need not be linear (a straight line) if that does not fit the data well.

We can **change the behavior or curve** of our hypothesis function by making it a quadratic, cubic or square root function (or any other form).

A screenshot of a cell phone

Description automatically generated

# Normal Equation

Gradient descent gives one way of minimizing J. Let’s discuss a second way of doing so, this time performing the minimization explicitly and without resorting to an iterative algorithm. In the "Normal Equation" method, we will minimize J by explicitly taking its derivatives with respect to the θj ’s, and setting them to zero. This allows us to find the optimum theta without iteration. The normal equation formula is given below:

A picture containing object

Description automatically generated

The following is a comparison of gradient descent and the normal equation:

A screenshot of a cell phone

Description automatically generated