Gradient Descent in Practice I - Feature Scaling

Note: [6:20 - The average size of a house is 1000 but 100 is accidentally written instead]

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 down to the optimum when the variables are very uneven.

The way to prevent this is to modify the ranges of our input variables so that they are all roughly the same. Ideally:

$$-1 \leqslant \mathbf{x_{(i)}} \leqslant 1$$

or

$$-0.5 \leqslant \mathbf{x_{(i)}} \leqslant 0.5$$

These aren't exact requirements; we are only trying to speed things up. The goal is to get all input variables into roughly one of these ranges, give or take a few.

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:

$$\mathbf{x_i} := \frac{\mathbf{x_i} - \mathbf{u_i}}{\mathbf{s_i}}$$

Where $\boldsymbol{\mu_i}$ is the **average** of all the values for feature (i) and $\mathbf{s_i}$ is the range of values (max - min), or $\mathbf{s_i}$ is the standard deviation.

Note that dividing by the range, or dividing by the standard deviation, give different results. The quizzes in this course use range - the programming exercises use standard deviation.

For example, if $\mathbf{x_i}$ represents housing prices with a range of 100 to 2000 and a mean value of 1000, then, $\mathbf{x_i} := \frac{\text{price} - 1000}{1900}$.