3. Gradient Descent in Practice

August 6, 2021

1 Introduction

2 ways of making gradient descent more efficient in practice (to make gradient descent converge faster towards global min) is using something called feature scaling and a specific learning rate.

2 Feature Scaling

Feature scaling is the practice of making features lie within the same range of values (because by nature features wont lie in the same range, the number of rooms in a house will have a different range of values than the distance of the house from sea-level). Applying feature scaling we can be assured that gradient descent will converge faster.

2.1 Example of Feature Scaling

Consider 2 features of a house, its size in feet ^{2} (in the range of $0 - 20000$) and the number of bedrooms (in the range $1 - 5$).

In finding values for the parameters θ_1 and θ_2, the gradient descent function will converge very slowly towards the global minimum of the cost function.

The cost function for the different ranges of x_1 and x_2 will be elliptical as depicted below

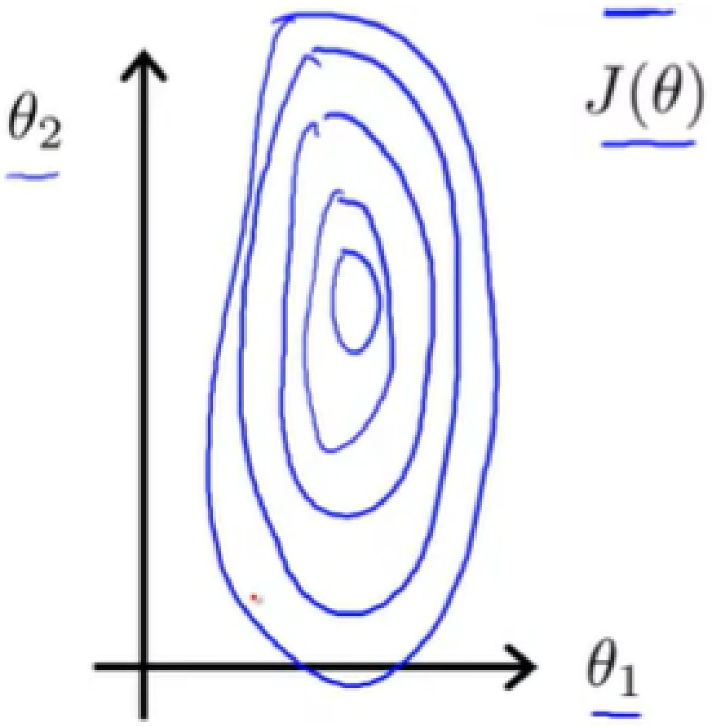


Figure 1: An elliptical shaped cost function for features that are not in the same range - with such a cost function, gradients will oscillate back and forth in search of the global min resulting in a longer time to converge.

If we apply feature scaling to the 2 variables according to the equation below, we ensure that both x_1 and x_2 lie in the range $0 \leq x_1, x_2 \leq 1$

$$\begin{aligned} x_1 &= \frac{size(feet^2)}{2000} \\ x_2 &= \frac{x_2}{5} \end{aligned} \tag{1}$$

With the feature scaling applied as above we get a cost function as shown below

$$\rightarrow x_1 = \frac{\text{size (feet}^2\text{)}}{2000}$$

$$\rightarrow x_2 = \frac{\text{number of bedrooms}}{5}$$

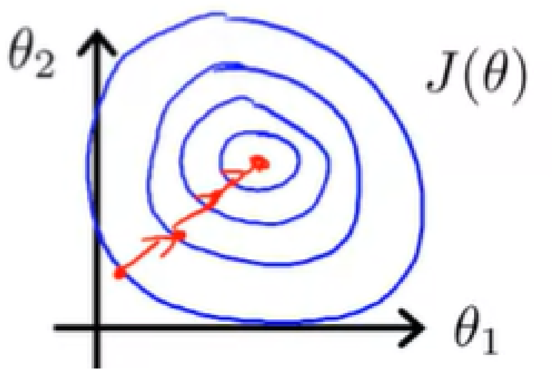


Figure 2: A circular shaped cost function arises from features that are in range

2.2 More on Feature Scaling

We are not restricted to making features lie between 0 and 1, so long as the range is not too far away from -1 and +1. Examples of valid ranges include but are not limited to:

$$\begin{aligned} -1 \leq x_1, x_2 \leq 1 \\ -2 \leq x_1, x_2 \leq 1 \\ -5 \leq x_1, x_2 \leq 5 \end{aligned} \tag{2}$$

Ranges that would constitute being too far from -1 and +1 include:

$$\begin{aligned} -100 \leq x_1, x_2 \leq 100 \\ -0.00001 \leq x_1, x_2 \leq 0.00001 \end{aligned} \tag{3}$$

2.3 Other ways of Feature Scaling

The way we applied feature scaling previously was to divide the values of x_1 and x_2 by the max value of each respective variable's previous scale. This is not the only way to scale the features however, another way is by using mean normalization which is given as

$$x \leftarrow \tfrac{x-\mu}{\sigma} \tag{4}$$

where x is the value of the feature to be scalled to the new scale, μ is the training set average and σ is the standard deviation or the max initial range minus the min initial range.

3 Learning Rate

We can make sure gradient descent works (converges) by plotting the cost function versus the number of iterations performed in finding the optimal parameter values.

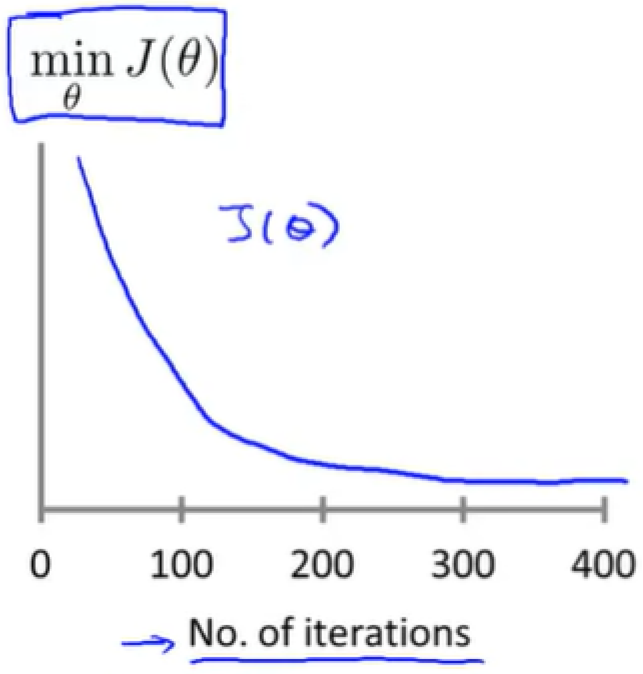
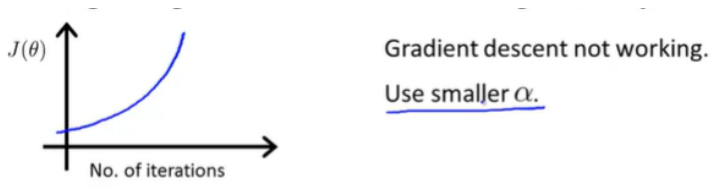
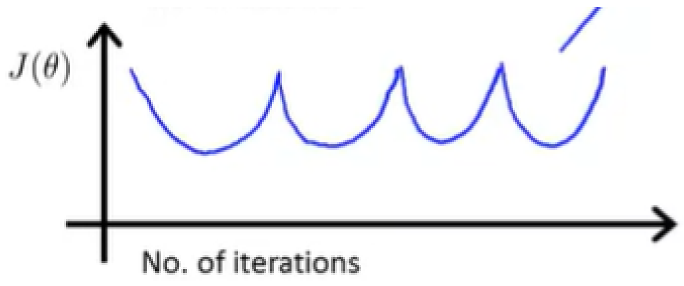


Figure 3: Plot of cost function versus the number of iterations performed in finding parameter values resulting in the smallest cost

From the plot in figure 3 we can also tell after how many iterations gradient descent converges (where the graph starts to become constant, between 300 and 400 iterations).

The graph also tells us whether our value for the learning rate is too large. For too large a learning rate we could get graphs that look like any of the below plots:





With both figures above, too large a learning rate means we are overshooting our minimum for cost.

So looking at the extreme (that α can be too large), what if α is too small ? A proof exists that shows that if α is too small we are guaranteed that $J()$ will still converge but at a slow rate.

We can choose a learning rate starting at 0.0001 and increase it to 3 times the previous value to find convergence at a reasonable rate - at least this is a suggestion from Andrew Ng.