Classes of Gradient Descent Algorithm

The three types of gradient descent algorithms are

- Batch gradient descent
- Mini-batch gradient descent
- Stochastic gradient descent

The ***batch gradient descent*** algorithm uses the entire training data in computing each step of the gradient in the direction of steepest descent. Batch gradient descent is most likely to converge to the global minimum. However, the disadvantage of this method is that, for massive datasets, the optimization process can be prolonged.

In ***stochastic gradient descent (SGD)***, the algorithm quickly learns the direction of steepest descent using a single example of the training set at each time step. While this method has the distinct advantage of being fast, it may never converge to the global minimum. However, it approximates the global minimum closely enough. In practice, SGD is enhanced by gradually reducing the learning rate over time as the algorithm converges. In doing this, we can take advantage of large step sizes to go downhill more quickly and then slow down so as not to miss the global minimum. Due to its speed when dealing with humongous datasets, SGD is often preferred to batch gradient descent.

Mini-batch gradient descent on the other hand randomly splits the dataset into manageable chunks called mini-batches. It operates on a mini-batch in each time step to learn the direction of steepest descent of the function. This method is a compromise between stochastic and batch gradient descent. Just like SGD, mini-batch gradient descent does not converge to the global minimum. However, it is more robust in avoiding local minimum. The advantage of mini-batch gradient descent over stochastic gradient descent is that it is more computational efficient by taking advantage of matrix vectorization under the hood to efficiently compute the algorithm updates.

Optimizing Gradient Descent with Feature Scaling

This process involves making sure that the features in the dataset are all on the same scale. Typically all real-valued features in the dataset should lie between $-1 \leq x_i \leq 1$ or a range around that region. Any range too large or arbitrarily too small can generate a contour plot that is too narrow and hence will take a longer time for gradient descent to converge to the optimal solution. The plot in Figure 16-3 is called a contour plot. Contour plots are used to represent 3-D surfaces on a 2-D plane. The smaller circles represent the lowest point (or the global optimum) of the convex function.

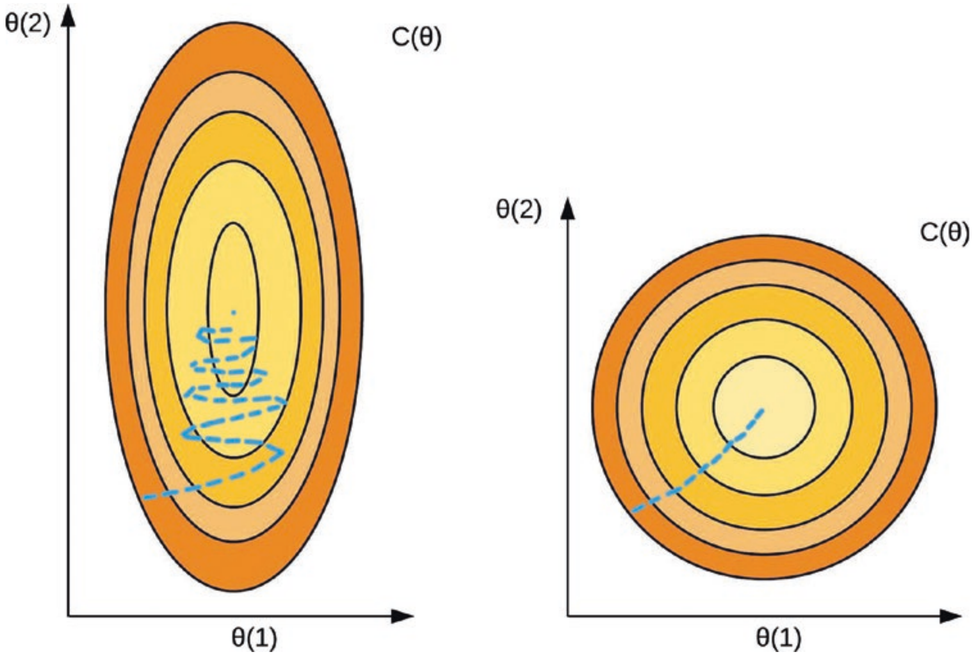
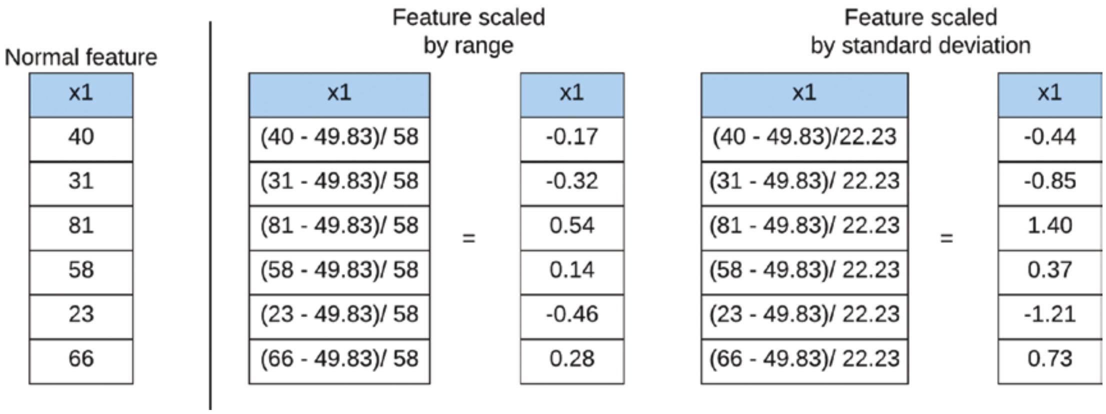


Figure 16-3. *Feature scaling – contour plots.* ***Left:*** *without feature scaling.* ***Right:*** *with feature scaling*

A popular technique for feature scaling is called mean normalization. In mean normalization, for each feature, the mean of the feature is subtracted from each record and divided by the feature's range (i.e., the difference between the maximum and minimum elements in the feature). Alternatively, it can be divided by the standard deviation of the features. Feature scaling is formally written as

$$x_i = \frac{x_i - \mu_i}{\max - \min} \ \textit{divided by range} \quad x_i = \frac{x_i - \mu_i}{\sigma} \ \textit{divided by standard deviation}$$

Figure 16-4 is an example of a dataset with feature scaling.



Normal feature

x1
40
31
81
58
23
66

Feature scaled by range

x1		x1
(40 - 49.83)/ 58		-0.17
(31 - 49.83)/ 58		-0.32
(81 - 49.83)/ 58	=	0.54
(58 - 49.83)/ 58		0.14
(23 - 49.83)/ 58		-0.46
(66 - 49.83)/ 58		0.28

Feature scaled by standard deviation

x1		x1
(40 - 49.83)/22.23		-0.44
(31 - 49.83)/ 22.23		-0.85
(81 - 49.83)/ 22.23	=	1.40
(58 - 49.83)/ 22.23		0.37
(23 - 49.83)/ 22.23		-1.21
(66 - 49.83)/ 22.23		0.73

Figure 16-4. *Feature scaling example*

In this chapter, we discussed gradient descent, an important algorithm for optimizing machine learning models. In the next chapter, we will introduce a suite of supervised and unsupervised machine learning algorithms.

CHAPTER 17

Learning Algorithms

In this section, we introduce a variety of supervised and unsupervised machine learning algorithms. The algorithms presented here provide a foundation for understanding other machine learning methods (e.g., linear and logistic regression), and others like Random forests and Extreme Stochastic Gradient Boosting (XGBoost) are widely used in applied machine learning.

We will survey the various learning algorithms from a conceptual level. In general, the discussion will cut across

- What a particular algorithm is all about and how it works.
- How we interpret the results of the learning algorithm.
- What various ways it can be optimized to improve performance in certain circumstances.

Classes of Supervised Algorithms

Supervised machine learning algorithms are broadly classified into

- Linear
- Non-linear
- Ensemble methods

E. Bisong, *Building Machine Learning and Deep Learning Models on Google Cloud Platform*,
https://doi.org/10.1007/978-1-4842-4470-8_17