Gradient Descent Algorithm in Machine Learning

Dr. P. K. Chaurasia

Associate Professor,
Department of Computer Science and
Information Technology
MGCUB, Motihari, Bihar
Objectives

- Introduction
- Optimization
- Gradient Descent
- Types of Gradient Descent
- Batch Gradient Descent
- Stochastic Gradient Descent
- Review Questions
- References
The objective of optimization is to deal with real life problems. It means getting the optimal output for your problem. In machine learning, optimization is slightly different. Generally, while optimizing, we know exactly how our data looks like and what areas we want to improve. But in machine learning we have no clue how our “new data” looks like, let alone try to optimize on it. Therefore, in machine learning, we perform optimization on the training data and check its performance on a new validation data.
Optimization Techniques

There are various kinds of optimization techniques, which is as follows:

- **Mechanics**: Deciding the surface of aerospace design.
- **Economics**: Cost Optimization
- **Physics**: Time optimization in quantum computing.

- Various popular machine algorithm depends upon optimization techniques like linear regression, neural network, K-nearest neighbor etc.
- Gradient descent is the most common used optimization techniques in machine learning.
Gradient descent is an optimization algorithm used to find the values of parameters (coefficients) of a function (f) that minimizes a cost function (cost).

Gradient descent is best used when the parameters cannot be calculated analytically (e.g. using linear algebra) and must be searched for by an optimization algorithm.
Gradient Descent

* Suppose a large bowl like what you would eat cereal out of or store fruit in. This bowl is a plot of the cost function (f).
* A random position on the surface of the bowl is the cost of the current values of the coefficients (cost).
* The bottom of the bowl is the cost of the best set of coefficients, the minimum of the function.
The goal is to continue to try different values for the coefficients, evaluate their cost and select new coefficients that have a slightly better (lower) cost.

Repeating this process enough times will lead to the bottom of the bowl and you will know the values of the coefficients that result in the minimum cost.
• Given function is $f(x) = \frac{1}{2} x^2$ which has a bowl shape with global minimum at $x=0$
  
  – Since $f'(x) = x$
    
    • For $x > 0$, $f(x)$ increases with $x$ and $f'(x) > 0$
    • For $x < 0$, $f(x)$ decreases with $x$ and $f'(x) < 0$

• Use $f'(x)$ to follow function downhill
  
  – Reduce $f(x)$ by going in direction opposite sign of derivative $f'(x)$
We often minimize functions with multiple inputs:

\[ f: \mathbb{R}^n \rightarrow \mathbb{R} \]

For minimization to make sense there must still be only one (scalar) output.
It determines a weight vector $w$ that minimizes $E(w)$ by

- Starting with an arbitrary initial weight vector.
- Repeatedly modifying it in small steps.
- At each step, weight vector is modified in the direction that produces the steepest descent along the error surface.
Method of Gradient Descent

• The gradient points directly uphill, and the negative gradient points directly downhill.
• Thus we can decrease function $f$ by moving in the direction of the negative gradient.
  – This is known as the method of steepest descent or gradient descent
• Steepest descent proposes a new point
  \[ x' = x - \eta \nabla_x f (x) \]
  – where $\epsilon$ is the learning rate, a positive scalar.
  Set to a small constant.
**Simple Gradient Descent**

**Procedure** Gradient-Descent (  
\[ \theta^1 \quad //\text{Initial starting point} \]  
\[ f \quad //\text{Function to be minimized} \]  
\[ \delta \quad //\text{Convergence threshold} \]  
)  
1 \[ t \leftarrow 1 \]  
2do  
3 \[ \theta^{t+1} \leftarrow \theta^t - \eta \nabla f \left( \theta^t \right) \]  
4 \[ t \leftarrow t + 1 \]  
5 while \[ \| \theta^t - \theta^{t-1} \| > \delta \]  
6 return( \( \theta^t \) )

**Intuition**

Taylor's expansion of function \( f(\theta) \) in the neighborhood of \( \theta \) is  
\[ f(\theta) \approx f(\theta^t) + (\theta - \theta^t)^\top \nabla f(\theta^t) \]

Let \( \theta = \theta^{t+1} = \theta^t + h \), thus  
\[ f(\theta^{t+1}) = f(\theta^t) + h \nabla f(\theta^t) \]

Derivative of \( f(\theta^{t+1}) \) wrt \( h \) is \( \nabla f(\theta^t) \)

At \( h = \nabla f(\theta^t) \) a maximum occurs (since \( h^2 \) is positive) and at \( h = -\nabla f(\theta^t) \) a minimum occurs. Alternatively, the slope \( \nabla f(\theta^t) \) points to the direction of steepest ascent. If we take a step \( \eta \) in the opposite direction we decrease the value of \( f \).

**One-dimensional example**

Let \( f(\theta) = \theta^2 \)

This function has minimum at \( \theta = 0 \) which we want to determine using gradient descent.

We have \( f'(\theta) = 2\theta \)

For gradient descent, we update by \( -f'(\theta) \)

If \( \theta^t > 0 \) then \( \theta^{t+1} < \theta^t \)

If \( \theta^t < 0 \) then \( f'(\theta^t) = 2\theta^t \) is negative, thus \( \theta^{t+1} > \theta^t \)
Ex: Gradient Descent on Least Squares

• Criterion to minimize
  – Least squares regression

• The gradient is
  \[ \nabla_x f(x) = A^T (Ax - b) = A^T A x - A^T b \]

• Gradient Descent algorithm is
  
  1. Set step size \( \epsilon \), tolerance \( \delta \) to small, positive nos.
  2. \textbf{while} \( \|A^T A x - A^T b\| > \delta \) \textbf{do}
     
     \[ x \leftarrow x - \eta (A^T A x - A^T b) \]
  3. \textbf{end while}
Stationary points, Local Optima

- When $f'(x) = 0$ derivative provides no information about direction of move
- Points where $f'(x) = 0$ are known as stationary or critical points
  - Local minimum/maximum: a point where $f(x)$ lower/higher than all its neighbors
  - Saddle Points: neither maxima nor minima
Presence of Multiple Minima

- Optimization algorithms may fail to find the global minimum.
- Generally accept such solutions.
Types of Gradient Descent Algorithms

* It can be classified by two methods:
  * Batch Gradient Descent Algorithm
  * Stochastic Gradient Descent Algorithm

* Batch gradient descent algorithms, use whole data at once to compute the gradient, whereas in stochastic you take a sample while computing the gradient.
The objectives of all supervised machine learning algorithms is to best estimate a target function \( f \) that maps input data \( X \) onto output variables \( Y \).

Some machine learning algorithms have coefficients that characterize the algorithms estimate for the target function \( f \).
Different algorithms have different representations and different coefficients, but many of them require a process of optimization to find the set of coefficients that result in the best estimate of the target function.

Examples of algorithms with coefficients that can be optimized using gradient descent are:

- Linear Regression
- Logistic Regression.
Stochastic Gradient Descent

- Gradient descent can be slow to run on very large datasets.
- One iteration of the gradient descent algorithm requires a prediction for each instance in the training dataset, it can take a long time when you have many millions of instances.
- When large amounts of data, you can use a variation of gradient descent called stochastic gradient descent.
- A few samples are selected randomly instead of the whole data set for each iteration. In Gradient Descent, there is a term called “batch” which denotes the total number of samples from a dataset that is used for calculating the gradient for each iteration.
Stochastic Gradient Descent

* Stochastic gradient descent selects an observation uniformly at random, say i and uses $f_i(w)$ as an estimator for $F(w)$. While this is a noisy estimator, we are able to update the weights much more frequently and therefore hope to converge more rapidly.

* Updates take only $O(d)$ computation, though the total number of iterations, $T$, is larger than in the Gradient Descent algorithm.
* Initialize \( w_1 \)

for \( k = 1 \) to \( K \) do

Sample an observation \( i \) uniformly at random

Update \( w_{K+1} \leftarrow w_K - \alpha \nabla f_i(w_K) \)

end for

Return \( w_K \).
Review Questions

- What is Optimization in Machine Learning?
- What is Gradient Descendent? Explain
- What are the different types of GDA? Explain.
- What is Batch Gradient Descent?
- What is stochastic gradient descent?
- Write an algorithm for SGD.
List of Books

- Understanding Machine Learning: From Theory to Algorithms.
- Introductory Machine Learning notes
- Foundations of Machine Learning

List of website for references

Thank you!