Gradient Descent Very Brief Review

Bar, Alon
205476013
balon94@gmail.com

Ben Tzvi, Nehoray
206389538
nehoray@mail.tau.ac.il

Baron, Lia
036635803
liabaron1@mail.tau.ac.il

November 6, 2024

Abstract

Gradient descent is an iterative optimization algorithm for finding the local minimum of a function. It and its variants are with no doubt the most common optimization algorithms which are used in ML.

1 Introduction

Optimization refers to the task of minimizing/maximizing an objective function $f(x)$ parameterized by x. In machine/deep learning terminology, it's the task of minimizing the cost/loss function $\mathrm{L}(\theta)$ parameterized by the model's parameters $\theta \in \mathbb{R}^d$. This objective function might be represented in many ways - it could be a simple linear function such that θ is just one matrix, as well as a 175B parameters transformer architecture, such that θ is a collection of large number of matrices and bias terms. Optimization algorithms (in the case of minimization) have one of the following goals:

- Find the global minimum of the objective function. This is feasible if the objective function is convex, i.e. any local minimum is a global minimum.
- Find the lowest possible value of the objective function within its neighborhood. That's usually the case if the objective function is not convex as the case in most deep learning problems.

2 Algorithm

Let's denote our objective function by L (it is often referred to it as the loss function), and let's say it is parameterized with some parameters which we will denote with θ. Additionally, let's denote an additional hyperparameter the algorithm uses, called the learning rate, with η. The Vanilla Gradient Descent Algorithm suggests following these following steps:

1. Start by Randomly initialize values for θ_0
2. Update θ values: $\theta_{t+1} = \theta_t - \eta \nabla L(\theta)$
3. Repeat until the slope is approximately flat. Namely, $\frac{\partial \mathcal{L}(\theta)}{\partial \theta} \approx 0$

2.1 Learning Rate

How big the steps gradient descent takes into the direction of the local minimum are determined by the learning rate, which figures out how fast or slow we will move towards the optimal weights. To illustrate the effects of a wrong/right choice of the learning rate, Figure 1 is attached.

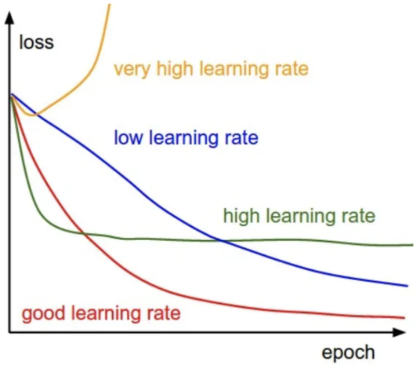

Figure 1: Effects of using too large/small learning rate while applying the GD algorithm