Gradient estimation for high-dimensional machine learning

ECE 543 Project Report

Suryanarayana Sankagiri

May 12, 2017

1 Introduction

Gradient based optimisation techniques are often used to solve optimisation problems that arise in machine learning. In many machine learning problems, the exact gradient of the optimal function is difficult/impossible to compute, so an approximation to the gradient is used instead. In the report, I shall discuss a stochastic gradient scheme that is presented in [Borkar et al., 2015]. This scheme is particularly suited for estimating the gradients of high-dimensional functions, when there is no explicit formula for the objective function or its gradient. The scheme uses two central concepts; the theory of Spall's Simultaneous Perturbation, and the theory of Compressive Sensing.

After presenting the theory behind this scheme, I shall discuss a few applications where this estimate of the gradient might be useful. The first application is in manifold learning or non-linear regression. Here, the estimate of the gradient is used to approximate the 'gradient outer product matrix', whose null space is the tangent space of the manifold. A second application is in reinforcement learning, where the the strategy of policy iteration involves calculating the gradient of the cost function with respect to the policy at each step.

2 Sparsity of high-dimensional gradients

A central idea used in the algorithm of [Borkar et al., 2015] is the fact that gradients of high-dimensional functions (functions with a large number of input variables) are

nearly sparse. In other words, The gradient can be well approximated by a sparse vector.

The intuition behind this claim is as follows. Consider a vector $x \in \mathbb{R}^n$, with $\|x\|_2 = L$. Then it is not possible for each component of the gradient to be larger than $L/\sqrt{n}$, else the norm constraint would be violated. If $L << \sqrt{n}$, then most of the components must be quite close to zero. If we have a restriction that our objective function f is L-Lipschitz continuous, then $\|\nabla f(x)\|_2 \leq L$.

In the context of gradient descent, the Lipschitz condition (or equivalently, the bound on the norm of the gradient is quite reasonable if we restrict ourselves to a small neighbourhood around a local minima. The Lipschitz assumption is also reasonable to make in the context of non-linear regression. If we wish to approximate our data as $Y = f(X) + e$, (where e denotes the noise/error), then it is reasonable to assume that the function $f(X)$ is L-Lipschitz for some reasonably small L, otherwise we might overfit the data.

3 The gradient estimation algorithm

The gradient estimation algorithm that is discussed in [Borkar et al., 2015] is useful in scenarios when there is no explicit formula for the gradient of the objective function. Such a situation arises in both reinforcement learning and manifold learning, as discussed in later sections. Rather, it is assumed that we are given a black-box into which we can feed in the arguments of the function and get the corresponding output.

Based on the black-box model, the simplest way to approximate the gradient is :

$$\nabla f(x) \approx \left(\frac{f(x + \Delta x_1) - f(x)}{\Delta x_1}, \dots \frac{f(x + \Delta x_n) - f(x)}{\Delta x_n} \right)$$

However, this method method involves $n + 1$ function evaluations and could be computationally expensive, if each function evaluation is time-consuming.

3.1 Spall's Simultaneous Perturbation

Spall's simultaneous perturbation method is a trick used to approximate the gradient using only two function evaluations [Spall, 1992]. Although such an estimate is bound to have a large error, it has certain properties that make it useful for certain applications. Below, we discuss how this method is used for stochastic gradient descent

Let us define $\{\Delta_i(k), 1 \leq i \leq n, k \geq 0\}$ to be i.i.d Rademacher variables, which are

either $+1$ or -1 with equal probability. We also define $\Delta(k) \triangleq (\Delta_1(k), \ldots, \Delta_n(k))$. By Taylor's theorem, we have for some small $\delta > 0$:

$$\widehat{\nabla f}(x(k))_i = \frac{f(x(k) + \delta\Delta(k)) - f(x(k))}{\delta\Delta_i(k)} \approx \frac{\partial f}{\partial x_i}(x(k)) + \sum_{j \neq i} \frac{\partial f}{\partial x_j}(x(k)) \frac{\Delta_j}{\Delta_i} i \forall\ i \in [n] \tag{1}$$

The key point is that In the above equation, the numerator is the same for all components, and the denominator is either $+\delta$ or $-\delta$. Thus, ignoring the sign, all the components of $\widehat{\nabla f}$ are the same, which means that $\widehat{\nabla f}$ is a rather poor estimator of ∇f in most cases. However, from our definitions, we observe that $\forall\ j \neq i$, $\mathbb{E}\left[\frac{\partial f}{\partial x_j}(x(k))\frac{\Delta_j}{\Delta_i}\right] = 0$. This means that $\widehat{\nabla f}(x(k))$ is an unbiased estimator of $\nabla f(x(k))$.

The theory of stochastic gradient descent [Bottou et al., 2016] tells us that an unbiased estimator of $\nabla f(x)$ is 'good enough' for the stochastic gradient descent (SGD) algorithm to converge to the optimal value. However, Theorems 4.6 and 4.7 in [Bottou et al., 2016] (covered in the course) also tell us that the rate of convergence of SGD depends upon the step-size chosen, which in turn depends inversely upon the variance of the noise of the estimate. Thus, it is useful to improve the estimate of the gradient from the one given in (1).

The estimator can be made more accurate by repeating the calculations in (1) multiple times, for independent realisations of the perturbation vector $\Delta(k)$ and averaging the values. However, this would lead to multiple function evaluations, which would be computationally expensive and would defeat the purpose of Simultaneous Perturbation. An alternative is to use methods from compressive sensing, which is a well-known and widely used method to retrieve sparse, high-dimensional vectors from noisy observations.

3.2 Compressive sensing

Here, we are going to use a method of compressive sensing that was first introduced in the paper [Candes and Tao, 2006]. In the most basic form, the problem formulation is as follows:

Let x be a vector in $\mathbb{R}^n$, which is unknown. Let A be an $m \times n$ matrix, which is known. Also, let $y = Ax$, and y is known. Then we can recover x by solving the optimisation problem:

$$x = \arg \min_{z \in \mathbb{R}^n} \|z\|_1 \text{ s.t. } Az = y \tag{2}$$

provided $m \sim n\log(n/s)$ and A satisfies the 'Restricted Isometry Property' (RIP) as

specified in [Candes and Tao, 2006]. It is known that matrices with i.i.d. Gaussian entries satisfy the RIP with very high probability.

The above formulation can be modified to accommodate noisy measurements of y. The exact formulation of the problem is borrowed from [Foucart and Rauhut, 2013].

Let A be an $m \times n$ matrix with RIP, and x be an s-sparse vector. Let $y = A\nabla f + \xi$ s.t. $\|\xi\|_2 \leq \eta$ be given. If for some $0 < \epsilon < 1$ and $\tau > 0$, m satisfies

$$\frac{m^2}{m+1} \geq 2s\Big(\sqrt{\log(en/s)} + \sqrt{\frac{\log(\epsilon^{-1})}{s}} + \frac{\tau}{\sqrt{s}}\Big)^2$$

Then w.p. $> 1 - \epsilon$, $\|\widehat{\nabla f} - \nabla f\|_2 \leq \eta/\tau$, where

$$\widehat{\nabla f} = \arg\min_{z \in \mathbb{R}^n} \|z\|_1 \text{ s.t. } \|y - Az\|_2 \leq \eta \tag{3}$$

3.3 The algorithm

The above theorem can be used for coming up with a good estimate of the gradient of our objective function in the following way. Essentially, we first compute (a noisy estimate of) $A\nabla f(x)$, and then use the compressive sensing optimisation problem to find a good estimate of $\nabla f(x)$. To avoid excess computation time, $A\nabla f(x)$ is computed using Spall's Simultaneous perturbation technique.

The algorithm is specified step-wise below:

- Select $A = (a_{ij})_{m \times n}$ with Gaussian entries
- Compute $y_i = \frac{f(x + \delta \sum \Delta_j a_j) - f(x)}{\delta \Delta_i}$ for $i = 1, \ldots, m$
- Repeat and average over y_i k times to get $\bar{y}_i$
- $\bar{y} = (\bar{y}_1, \ldots, \bar{y}_m) = A\nabla f(x) + \eta$
- Solve $\min_{z \in \mathbb{R}^n} \|z\|_1$ s.t. $\|y - Az\|_2 \leq \eta$
- Output $\widehat{\nabla f}(x)$

3.4 Trade-offs

(this comment is not there in the paper [Borkar et al., 2015].) In (3), it may seem that we can increase the accuracy of our gradient estimate greatly by choosing τ to be as large as possible. The fact that m grows as τ^2 does not seem to be a hindrance as the vector $A\nabla f(x)$ can be calculated with only two function evaluations, irrespective of its dimension. However, the noise in the observations increases as the dimensions of $A\nabla f(x)$ increases. This can be inferred by the growing number of terms in the noise expression in (1) with n. Thus, as m increases, η increases in (3).

4 Application to reinforcement learning

Consider the following setting in reinforcement learning, which is identical to that of a Markov Decision Process, except that the state-transition probabilities and rewards are not known. Consider the case of infinite horizon. The aim of reinforcement learning is to choose the best stationary policy, *i.e.*, the best decision at each state that will lead to the lowest long-run discounted cost. Every policy can be represented by a vector which is the size of the state-space (which is assumed to be finite but large). Assume that the set of available actions in each state is a continuous interval. Thus, the long run cost as a function of policy can be viewed as a function $C : \mathbb{R}^n \to \mathbb{R}$, which is unknown, but can be simulated or obtained via experimentation. An arbitrary initial policy can be chosen, and the optimal policy can be chosen via policy gradient descent. This application is mentioned in [Borkar et al., 2015], with additional references given as [Sutton et al., 1999] and [Zhao et al., 2011].

Note: *Due to my incomplete understanding, I could not summarise the idea behind manifold learning/non-linear regression*

References

Vivek S Borkar, Vikranth R Dwaracherla, and Neeraja Sahasrabudhe. Gradient estimation with simultaneous perturbation and compressive sensing. *arXiv preprint arXiv:1511.08768*, 2015.

Léon Bottou, Frank E Curtis, and Jorge Nocedal. Optimization methods for large-scale machine learning. *arXiv preprint arXiv:1606.04838*, 2016.

Emmanuel J Candes and Terence Tao. Near-optimal signal recovery from random projections: Universal encoding strategies? *IEEE transactions on information theory*, 52(12):5406–5425, 2006.

Simon Foucart and Holger Rauhut. *A mathematical introduction to compressive sensing*, volume 1. Birkhäuser Basel, 2013.

James C Spall. Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. *IEEE transactions on automatic control*, 37(3): 332–341, 1992.

Richard S Sutton, David A McAllester, Satinder P Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In *NIPS*, volume 99, pages 1057–1063, 1999.

Tingting Zhao, Hirotaka Hachiya, Gang Niu, and Masashi Sugiyama. Analysis and improvement of policy gradient estimation. In *Advances in Neural Information Processing Systems*, pages 262–270, 2011.