490 Project: Performance of Perturbed Gradient in Escaping Saddle Points

Boya Hou, Yiqing Xie, Chuyuan Tao

I. Introduction

The stopping criterion of gradient descent is usually of the form $\|\nabla f(x)\| \leq \epsilon$, where ϵ is small and positive. However, for non-convex functions, gradient descent might converge to local maxima or saddle points which are also first- order stationary points. It has been shown in several literature that one method to escape ”strict” saddle point in the non-convex setting is to add noise. Nevertheless, [1] did not consider degenerate saddle points. Our project focus on investigating how the interval between the random jumps, and the size of the random jumps impact the probability of stalling near the saddle point even when there are degenerate saddle points.

II. Related work

Jin et al [1] shows a perturbed gradient descent algorithm that converges to a ϵ second order stationary point in essentially the same amount of time as gradient descent finds first order stationary point. They proves that ”stuck region” around a saddle point only consists of a very small proportion of the volume of perturbation ball (the region where we generate perturbation). However, it requires ”strict saddle condition”, where the Hessian matrix at the saddle point has a strictly negative eigenvalue. On the other hand, non-convex problems can involve degenerate saddle points such as monkey saddle point whose second order local minimum but not third order local minimum. Such degenerate structure is not considered in this paper. For problems without the strict saddle property, their perturbed gradient descent techniques can converge to a saddle point. We attempt to apply perturbed gradient method to investigate how parameters affect the performance of perturbed gradient descent even when there are degenerate saddle points .

III. Our Algorithm

A. *Define Perturbation Condition*

1) Gradient is small, which indicates that current point is probably near a saddle point
2) No perturbation in last several iterations.

B. *Algorithm*

1) For t=0,1,... do
2) If perturbation condition holds, then random jump:
3) $x_t = x_t + \rho d$, where ρd is a random point generated from unit ball.(ρ is the size of random jump, d is the direction.
4) $x_{t+1} = x_t - \alpha \nabla f(x)$

C. *Choice of Parameters*

Since we want to escape saddle point both successfully and as efficiently as possible, we have to choose jump interval and jump size along with initial point and step size judiciously. Jump interval will affect efficiency and a proper jump size will guarantee the ability to escape a saddle point. In the following sections, we will discuss how jump interval T an jump size ρ contribute to performace of our algorithm.

IV. Algorithm Performance

V. Conclusion

References

[1] Chi Jin et al. “How to Escape Saddle Points Efficiently”. In: (Mar. 2017). arXiv: 1703.00887. URL: http://arxiv.org/abs/1703.00887.