IFT 6085 - Lecture 27
SGD Escapes Saddle Points

Scribe(s): [Riashat Islam] **Instructor:** Ioannis Mitliagkas

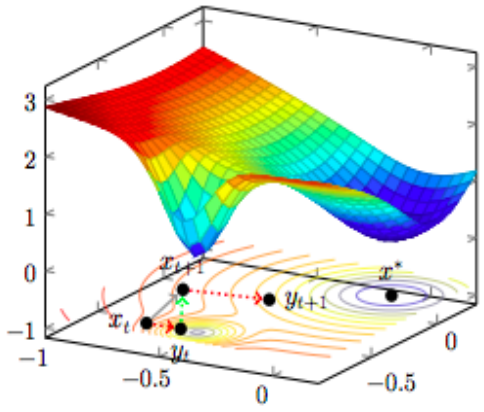
1 Introduction

Stochastic gradient descent (SGD) is widely used in machine learning, and is known to find better solutuons than gradient descent for modern neural networks. SGD is usually computed with a mini-batch of the dataset and have become the de facto algorithms for training neural networks. In the regime of convex optimization, SGD is proved to be a nice tradeoff between accuracy and efficiency: it requires more iterations to converge, but fewer gradient evaluations per iteration.

In this report, we explain that SGD is working on a smoothed version of the loss function, and even if the loss function may have many poor local minima or saddle points (which may make SGD prone to getting stuck), SGD will not get stuck at a sharp local minima with small diameters, conditional that the neighbourhoods of these regions contain enough gradient information. The size of this neighbourhood is controlled by the step size and gradient noise. In other words, SGD wll get close to the local minima, stay around the minima with constant probability but eventually will escape the local minima.

We demonstrate further that for training neural networks, noise is a crucial component for non-convex optimization problems, and with the help of these noisy gradients SGD converges faster and to a better solution. Most importantly, the noise helpds SGD escape saddle points and can give better generalization, while also guaranteeing polynomial hitting time of googd local minima under some given assumptions [1], [2], [3]

This can be further illustrated using the figure below, which shows that SGD could escape a local minimum within one step. We argue that SGD converges to a good local minima since the sharp local minima are often eliminated by the convolution operator that transforms f into another function, and this convolution has the effect of smoothing out short-range fluctuations.



2 Background

In the previous lecture, we have shown that SGD converges to a critic point at a rate of $O(\frac{1}{T})$ where the idea was to show that the gradient $\nabla f(w_k) = 0$ closer to a critical point. Our goal was to find a minimum, and control sub-optimality $f(w_k) - f* < \epsilon$. However, getting close t a global optima is equivalent to a NP-hard problem. This is mainly because a non-convex function may have many local minima and it might be hard to find the best one (global minimum) among them. Furthermore, even finding a local minimum might be hard as there can be many saddle points which have 0 gradient, but are not local minima. In general, there are no known algorithm that guarantees to find a local minimum in a polynomial number of steps.

In the case of deep neural networks, the main bottleneck in optomization is not due to local minima, but due to the existence of saddle points. Gradient based algorithms are in particulr susceptible to saddle point problems as they only rely on the gradient information.

We made some simplifying assumptions previosuly.

- If the sub-optimal $f(w_k) - f*$ is big, then we could conclude that we are not close to an optimal and it is a strong gradinet. In other words, it is only the critical points which are a global minima. We showed in our theorem before that if β-smoothed and it satisfies the PL condition, then SGD converges to a minimum at a rate of $O(\frac{1}{T})$.

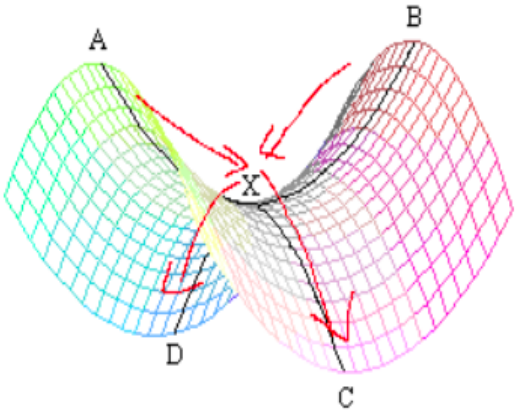
3 SGD escapes saddle points

Definition of Strict Saddle Points : Given a function f(w) that is twice differentiable, we call w a stationary point if $\nabla f(w) = 0$. A stationary point can either be a local minimum, a local maximum or a saddle point. We identify an interesting class of non-convex functions which we call strict saddle. For these functions the Hessian of every saddle point has a negative eigenvalue. In particular, this means that local second-order algorithms which are similar to the ones in (Dauphin et al., 2014) can always make some progress. It may seem counter-intuitive why stochastic gradient can work in these cases: in particular if we run the basic gradient descent starting from a stationary point then it will not move. However, we show that the saddle points are not stable and that the randomness in stochastic gradient helps the algorithm to escape from the saddle points.

Some properties of stationary points $\nabla f(w) = 0$:

- It is a local minimum if $\nabla^2 f(w) \geq 0$
- It is a local maximim if $\nabla^2 f(w) \leq 0$
- It is a saddle point if $\nabla^2 f(w)$ has both positive and negative eigenvalues

Figure below for example, shows a saddle point with negative eigenvalue



In this lecture, we proceed under the hypothesis that almost all minima are almost global and as neural networks get wider an increasing fraction of minima becomes better approximations to a global minima. In other words, almost all minima of overparameterized neural networks are almost global, and if all minima are almost the same, then we are good to converging to any one of them.

- In this talk, we will further show how to guarantee that we will not get stuck to saddle points. We will see hw many steps are typically needed to escape saddle points and why does adding noise help in this.

Two of the possible solutions why we don't get stuck at saddle points if we are using SGD are as follows.

- Pushed around by stochastic noise. Our analysis shows that we need a little bit of injected noise to escape saddle points.
- For mini-bathces to train neural nets, the objective changes

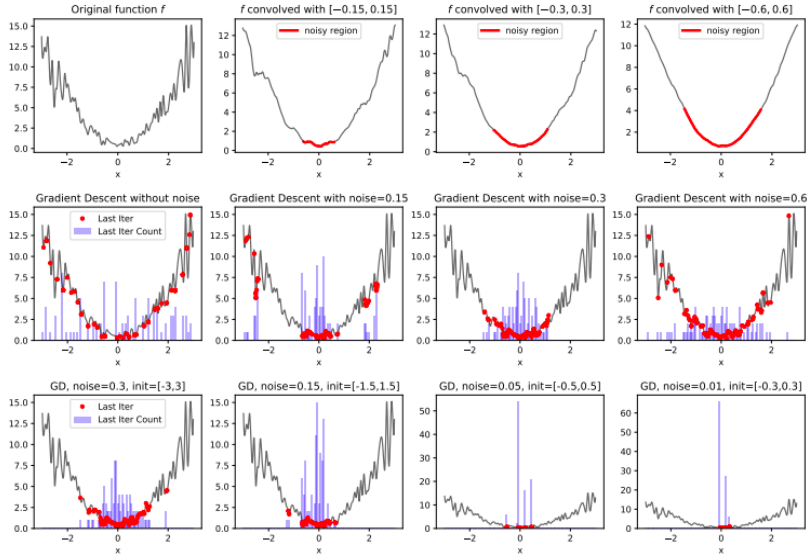


Figure above demonstrates SGD on a spiky function f. **Row 1**: f gets smoother after convolving with uniform random noise. **Row 2**: Run SGD with different noise levels. Every figure is obtained with 100 trials with different random initializations. Red dots represent the last iterates of these trials, while blue bars represent the cumulative counts. GD without noise easily gets stuck at various local minima, while SGD with appropriate noise level converges to a local region. **Row 3**: In order to get closer to x*, one may run SGD in multiple stages with shrinking learning rates.

Adding noise to SGD

$$g_t = \nabla f(x_t) + \epsilon_t \tag{1}$$

where $\epsilon_t \sim N(0, I)$. This is an isotropic noise (ie, same variance in all directions) and ϵ_t encompases all the variation in a big isotropic ball. We argued that for small enough ϵ noise ball, the smallest term in the Taylor expansion for $f(w + \epsilon v)$ would dominate if we write out the Taylor expansion for $f(w + \epsilon v)$

$$f(w + \epsilon v) = f(w) + \epsilon \nabla f(w)^T v + \epsilon^2 v^T \nabla^2 f(w) v \tag{2}$$

Therefore, $f(w + \epsilon v) \geq f(w)$, ie there exists $\epsilon > 0$ such that for any direction around w, there would be local minima. A strict saddle property is that, if the gradient is high, $||f(w)||_2 \geq \epsilon$, assuming any stationary point is a local minima/maximima or saddle point. The noise assumptions therefore help explain why SGD will escape all saddle points and converge to a local minimum. We therefore draw the conclusions that, all minima can be considered as equally good, and SGD arrives at one local minimum which can be considered the global minimum.

4 Perturbed Gradient Descent

We discussed perturbed SGD (PGD) [2] which is a variant of SGD that also helps to escape all saddle points and converge to a local minimum much faster. We discussed that PSGD, a perturbed form of SGD, converges to a second order stationary point, and the convergence is less dependent on the dimension. (Jin et al 2017) showed that when all saddle points are non-degenerate, all second-order stationary points are local minima and their result thus shows that perturbed gradient descent can escape saddle points almost for free. In short, the algorithm is as follows :

- If $||\nabla f(x_t)|| \leq g_{thresh}$ and last perturbed time is $> t_{thresh}$ steps before, then do a random perturbation (ball)
- If perturbation happend t_{thresh} ago, but f is decreased for less than f_{thresh}, return the value before last perturbation
- Do a gradient descent step $x_{t+1} = x_t - \eta \nabla f(x_t)$

Intuitively, what PGD does is : it adds Gaussian noise once in a while when you sense that the gradient is tiny, since a tiny gradient could be a saddle point. It will add noise if $||\nabla f(w)|| \leq g_{thresh}$ if the last time perturbed is greater than t_{thresh}. In other words, it will make progress when there is large gradient, and look to escape a saddle point if gradient is small.

Lemma : If we are at a saddle point, and we perturb in PGD, then the function value can have a significant reduction. Just by looking at the gradient or hessian, we can tell whether we are at a minimum or saddle point.

Few of the contributions from PGD are as follows :

- Convergence with PGD will result in finding a local minima, which also means that gradient descent can escape all saddle points with only logarithmic overhead in runtime
- Convergence to local minima and escaping saddle points highly depends on the characterization of the geometry around saddle points - ie, points from where gradient descent gets stuck t a saddle point constitute a thin band. After a random perturbation, the point is very unlikely to be in the same band and hence can efficiently escape from the saddle point

References

[1] R. Ge, F. Huang, C. Jin, and Y. Yuan. Escaping from saddle points - online stochastic gradient for tensor decomposition. In *Proceedings of The 28th Conference on Learning Theory, COLT 2015, Paris, France, July 3-6, 2015*, pages 797–842, 2015. URL `http://jmlr.org/proceedings/papers/v40/Ge15.html`.

[2] C. Jin, R. Ge, P. Netrapalli, S. M. Kakade, and M. I. Jordan. How to escape saddle points efficiently. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, pages 1724–1732, 2017. URL `http://proceedings.mlr.press/v70/jin17a.html`.

[3] R. Kleinberg, Y. Li, and Y. Yuan. An alternative view: When does SGD escape local minima? *CoRR*, abs/1802.06175, 2018. URL `http://arxiv.org/abs/1802.06175`.