On the difficulty of high-dimensional non-convex optimization

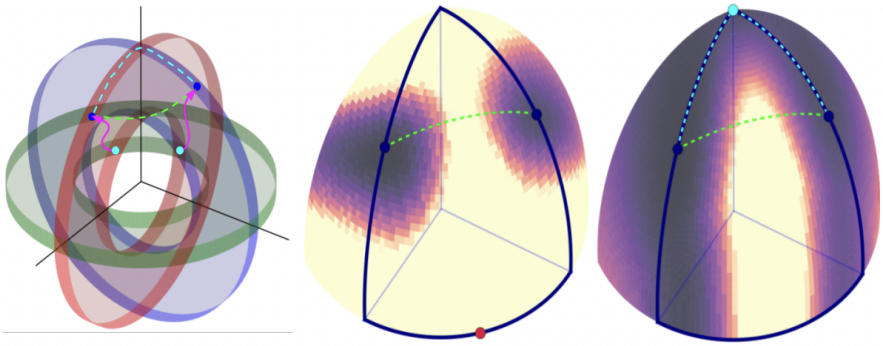

Figure 1: A model of the low-loss manifold comprising two wedges in 3-dimensional space.

Deep learning involves the optimization of non-convex error functions over continuous, high-dimensional spaces. Gradient descent, quasi-Newton, and BFGS are commonly used to perform such minimizations and find the optimal solution (global minima). Despite the high dimension of the weight space loss landscape, the optimization behavior shows many surprisingly simple features. Li et al. 2018a show that despite the typically very high number of trainable parameters, constraining optimization to a small number of randomly chosen directions often suffices to reach a comparable accuracy. Goodfellow et al.[2015] show that there is a smooth path connecting the initialization and the final minima. This suggests the existence of possibly long directions of the solution manifold- directions in which the loss changes slowly. Some of the surprising observations reported in the Stanislav [2019] paper are as follows:-

1. No significant obstacles along the optimization path indicate the existence of long directions.
2. Constraining optimization to a random, low-dimensional section of the weight space is sufficient for good optimization (comparable to full optimization in terms of performance). The performance depends on the hyperplane's dimension and also on the radius at which it is positioned in the weight space. Thus, an important result is that good solutions are everywhere and are evenly distributed densely enough that even a random, low-dimensional hyperplane hits them consistently.

With non-convex functions, such as neural nets, it is possible to have many local minima. Indeed, nearly any deep model is essentially guaranteed to have a large number of local minima. However, this is not necessarily a significant problem. It is empirically confirmed that as the dimensionality N increases, local minima with high error relative to the global minimum occur with a probability that is exponentially small in N. This is mainly because neural nets exhibit model identifiability problems which results in a large number of local minima but all are equivalent in their cost function value. They do not have very high error values relative to the global minima and are thus good solutions that stochastic gradient descent (SGD) often finds themselves during the training process. Critical points with high costs are far more likely to be saddle points. We are not exactly concerned with finding the exact minimum of a function but seek only to reduce its value significantly to obtain a good generalization error.

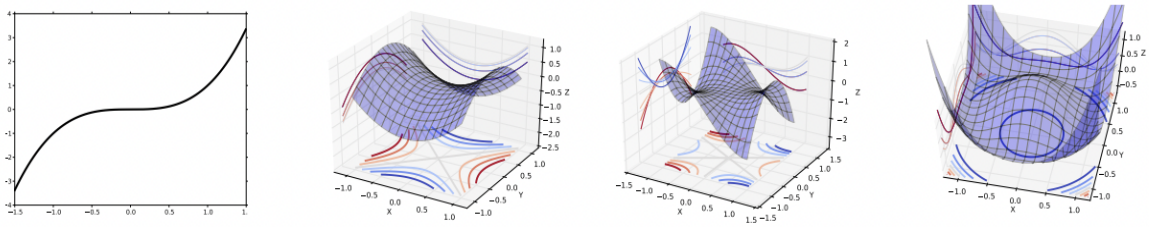

Figure 2: Illustration of three different types of saddle points.

Consider an error function f(θ) where θ is an N-dimensional continuous variable. A critical point is by definition a point θ where the gradient of f(θ) vanishes. All critical points of f() can be further characterized by the curvature of the function as described by the eigenvalues of the Hessian. They are as follows:-
1. If all eigenvalues are non-zero and positive, then the critical point is a local minimum.
2. If all eigenvalues are non-zero and negative, then the critical point is a local maximum.
3. If all the eigenvalues are non-zero and we have both positive and negative eigenvalues, then the critical point is a saddle point with a min-max structure (see Figure 2).
4. If the Hessian matrix is singular, the degenerate critical point can be a saddle point. If it is a saddle, then if we restrict θ to only change along the direction of singularity. The restricted function does not exhibit a minimum nor a maximum; it indicates, in second order, a plateau. When moving from one side to the other of the plateau, the eigenvalue corresponding to this picked direction generically changes sign, being exactly zero at the critical point.

A plateau is an almost flat region in some direction. This structure is given by having the eigenvalues corresponding to the directions of the plateau be close to 0, but not exactly 0. In these flat regions, the gradient and the Hessian are all zero. In a convex problem, a wide, flat region must consist entirely of the global minima. Still, in a general optimization problem, such a region could correspond to a high objective function value. This structure is given by having the eigenvalues corresponding to the directions of the plateau be close to 0, but not exactly 0. Saddle points are a significant hindrance for rapid high-dimensional non-convex optimization. The Hessian matrix has both positive and negative eigenvalues at a saddle point. Points lying along eigenvectors associated with positive eigenvalues have a greater cost than the saddle point, while points lying along eigenvectors with negative eigenvalues have lower values. We can think of a saddle point as being a local minimum along one cross-section of the cost function and a local maxima along another cross-section. Many classes of random functions, such as the Gaussian process, exhibit the following behavior: local minima are common in low-dimensional space. In higher-dimensional spaces, local minima are rare, saddle points are more common. In n-dimensional space, the expected ratio of the number of saddle points to local minima grows exponentially with n.

Dauphin et al. [2014] experimentally showed that real neural networks have loss functions that contain many high-cost saddle points. Saddle points are unstable under gradient descent dynamics on the error surface, as the dynamics are repelled away from the saddle by directions of negative curvature. However, this repulsion can occur due to the plateaus of small negative eigenvalues. A similar slow-down can occur for local minima with small positive eigenvalues. Second order methods were designed to rapidly descend local minima by rescaling gradient steps by inverse eigenvalues. However, the Newton method does not treat saddle points appropriately; they become attracted to these points and make learning halt under these settings. While gradient descent is designed to move "downhill" and is not explicitly optimized to seek a critical point, Newton's method is designed to seek a point where the gradient is zero and can easily get stuck in the saddle point.
Also the trajectory of SGD from the initialization point to the local minima can be highly complicated in high-dimensional spaces. For example, when a learning curve bounces up and down repeatedly, we do not know whether the objective function is highly bumpy or whether SGD is rapidly changing direction due to noise in the stochastic, minibatch-based estimate of the gradient. When the objective function remains constant for long periods, we do not know whether the parameters are stuck in a flat region, oscillating around a local minimum, or tracing their way around the perimeter of a larger obstacle.

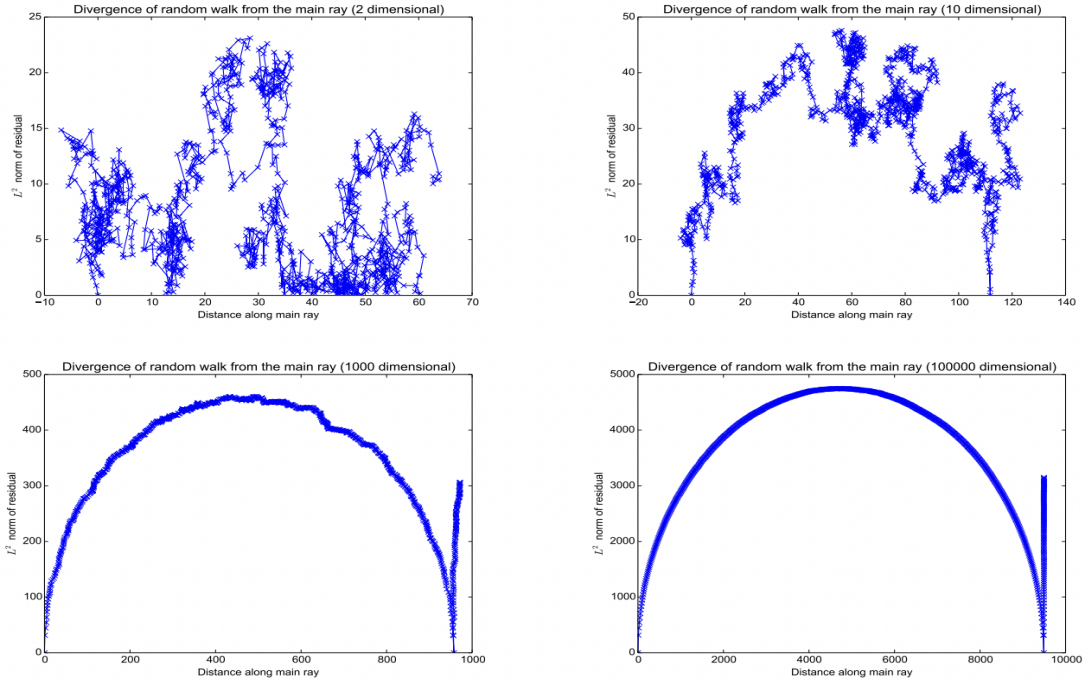

Figure3: Plots of the projection along the axis from the initialization to solution versus the norm of the residual of this projection for random walks of varying dimension. Each plot is formed by using 1,000 steps. The step 900 is designated as being the "solution" and continue to plot 100 more steps. Each step is made by incrementing the current coordinate by a sample from a Gaussian distribution with zero mean and unit covariance. Figure adapted from Goodfellow et al.[2015].

In Goodfellow et al.[2015] they show that there exists a linear subspace in which a neural network could proceed by descending a single smooth slope with no barriers. They also show that in some cases SGD does encounter obstacles, such as a ravine that shapes its path and the local minima or saddle points never really slowed the SGD trajectory. This suggests that less exotic problems such as poor conditioning and variance in the gradient estimate are the primary difficulties in training neural networks.

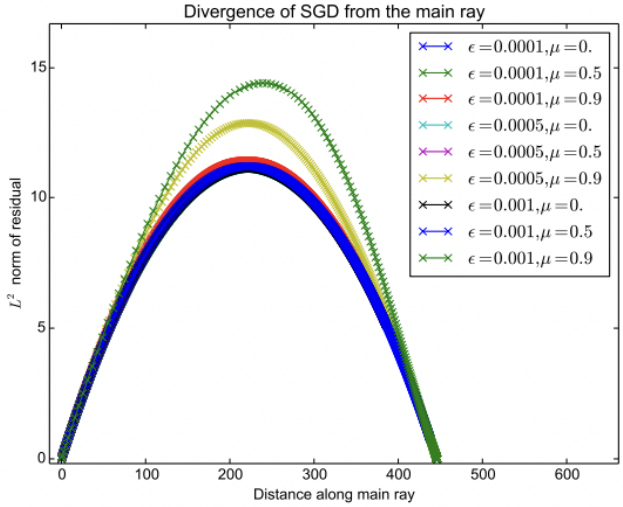

Figure 4: To show the effect of different learning rates and momentum coefficients μ. To make the plots comparable, they use the true solution to a synthetic, convex problem as the endpoint for all trajectories (In the non-convex case, different trajectories could go to different trajectories). Because this problem is 100,000 dimensional, the curves all have a very simple shape, and the primary quantity distinguishing them is the maximum norm of the residual. Figure adapted from Goodfellow et al.[2015].

Ill-conditioning of the Hessian can also cause problems during neural net training which can make SGD to get "stuck" in the sense that even small steps increase the cost function. Newton's method is an excellent tool for minimizing convex functions with poorly conditioned Hessian matrices, but that is not the case for non-convex optimization of deep neural networks. This article was an attempt to summarize some of the difficulties and obstacles during neural network training from various perspectives.

References

1. Li, Chunyuan, et al. "Measuring the intrinsic dimension of objective landscapes." *arXiv preprint arXiv:1804.08838* (2018).
2. Goodfellow, Ian J., Oriol Vinyals, and Andrew M. Saxe. "Qualitatively characterizing neural network optimization problems." *arXiv preprint arXiv:1412.6544* (2014).
3. Fort, Stanislav, and Stanislaw Jastrzebski. "Large scale structure of neural network loss landscapes." *Advances in Neural Information Processing Systems* 32 *arXiv:1906.04724* (2019).
4. Dauphin, Yann N., et al. "Identifying and attacking the saddle point problem in high-dimensional non-convex optimization." *Advances in neural information processing systems* 27 *arXiv:1406.2572* (2014).