On the Study of the Topology, Geometry, and Second Order Properties of the Loss Surface of Deep Neural Networks

Yvonne B. Alama
yvonne.alama@mail.mcgill.ca

Anna Brandenberger
anna.brandenberger@mail.mcgill.ca

School of Computer Science, McGill University
For Prof. Prakash Panangaden

1 Introduction

Our adventure began by discovering, thanks to the list of recommended papers given by Dr. Panangaden, the paper by Daxler et al with the exciting title: *Essentially No Energy Barriers in Neural Network Energy Landscape*, [1]. This paper, being of empirical nature, provided us with insight on the numerical construction of continuous paths along the loss function connecting two arbitrary minima, which was done using an algorithm from molecular statistical mechanics. Their experiments showed that as the architecture gets wider and especially deeper, the loss at the saddle points remains close to the loss of the original minima, yielding paths on which the value of the loss function remains low. In the hope of adding theoretical results to our present empirical knowledge of the loss landscape of neural networks, we referred to the paper by Freeman and Bruna entitled *Topology and Geometry of Half-Rectified Network Optimization*, [2]. This fruitful paper provided us with rigorous (as well as empirical) results relating the connectedness of the landscape and the size of the hidden layers through the study of the topology and geometry of the level sets of the loss function. Thanks to these two papers we witnessed, mostly through empirical results, the important interplay between the connectedness of the landscape and the model over-parametrization, which was referred as common knowledge amongst deep learning practitioners. However, for us this relationship was a surprising one, and we were in search of a theoretical explanation which would justify the intuition of deep learning practitioners behind this vital interplay. This goal was fulfilled by analyzing the paper by Sagun et al entitled *Empirical Analysis of the Hessian of over-parametrized Neural Networks*, [3], which offers a theoretical intuition of the empirical results found in [1] and [2] of the connected structure of the landscape through the study of the second order properties of the Hessian of the loss function.

This paper is divided as follows: we present the main ideas of the papers [2] and [1] in section 2 and 3 respectively. These two sections offer the motivation for section 4, in which we analyze the theoretical intuition given by [3], and provide a possible explanation of the connectedness of the landscape through over-parametrization.

2 Connectedness of level sets of the loss function

The paper [2] first characterizes poor local minima using topological connectedness. Poor local minima are defined in this paper as local minima that are not global minima, at which gradient descent can get stuck. [2] offers a theoretical analysis of the conditions on the data distribution and model architecture that ensure connectedness. Connectedness is indeed an essential characteristic of network loss surfaces, since connected level sets imply that one can always find a descent direction at each energy level. We will see in Proposition 2.1 that when all energy levels are connected, there can be no poor minima i.e. all local minima are also global. This theoretical analysis is performed for linear networks and single layer non-linear ReLU networks. The authors then experimentally study the connectedness of loss landscapes for a variety of datasets and more complicated non-linear deep models.

The paper first deals with the linear case, in which it has previously been proven that deep (linear) networks without regularization term have asymptotically connected level sets ([5]). The paper [2] provides an alternative proof of this fact, which in fact generalizes the two-layer case to also include linear networks with a ridge regression term in the loss function (which consists of regularization based on the ℓ_2 norm).

Then, they move to the half-rectified ReLU case and prove that these networks are also asymptotically connected, providing explicit bounds which link connectedness with network over-parametrization (Section 2.2).

Before presenting the results of the paper, we must define the basic notation, which we will also re-use in Section 4. Let $x \in \mathbb{R}^d$, and denote the true label by $y \in \mathbb{R}$. Define the sample data by $D = \{(x^i, y^i)\}_{i=1}^N$, generated randomly according to an unknown distribution $\mathcal{D}$. Let our model be parametrized by $w \in \mathbb{R}^M$, hence the number of parameters in the system is M.

Definition 2.1 (Network)**.** A network is defined by a model function

$$f(\cdot, w) : \mathbb{R}^d \longrightarrow \mathbb{R}$$

i.e. the output of the network for input $x \in \mathbb{R}^d$, using parameters $w \in \mathbb{R}^M$. For example, for a deep neural network, w contains the weights and biases used for all layers.

Definition 2.2 (Optimization problem)**.** We can then define the empirical and oracle loss functions which respectively have forms

$$L_e(w) = \frac{1}{N}\sum_{i=1}^{N} \|f(x_i, w) - y_i\|^2 + \kappa\mathcal{R}(w) \text{ and } L_o(w) = \mathbb{E}_{(X,Y)\sim\mathcal{D}}\|f(X, w) - Y\|^2 + \kappa\mathcal{R}(w)$$

where $\mathcal{R}(w)$ is the regularization term. The paper considers sparse and ridge regularization which consist respectively of the ℓ_1 and ℓ_2 norms of w.

Definition 2.3 (Level set)**.** We define the level set of the loss surface $L(w)$ as

$$\Omega_L(\lambda) = \{w \in \mathbb{R}^M \mid L(w) \leq \lambda\}$$

the set of parameters w yielding a loss smaller or equal to λ, for each energy level λ.[1]

Definition 2.4 (Strict local minima)**.** $w \in \mathbb{R}^M$ is a strict local minima of $L(w)$ if there exists $\epsilon > 0$ such that $\forall\ w' \in B(w, \epsilon),\ w' \neq w$, we have $L(w') > L(w)$.

An important question in learning is whether there exist poor local minima. We have the following proposition which links this question to the connectedness of level sets Ω_L defined above. We offer a proof, another is given in Appendix B of [2].

Proposition 2.1. *If $\Omega_L(\lambda)$ is connected for all λ, then every local minima of $L(w)$ is a global minima.*

Proof. First consider the case where our local minima, w, is a strict one, satisfying Definition 2.4. We now argue by contradiction: say $\Omega_L(\lambda)$ is connected $\forall\lambda$, but there exists a strict local minima of $L(w)$ which is not a global one. Let w be this strict local min and let $L(w) = \lambda_0$.

[1] This definition applies for both the oracle $L_o(w)$ and empirical $L_e(w)$ loss functions.

Consider $\Omega_L(\lambda_0)$, by definition of strict local minima, $\Omega_L(\lambda_0)$ must contain w and cannot contain $B(w,\epsilon) \setminus w$. Furthermore, since w is not a global min there must exist $\tilde{w}$ s.t $L(\tilde{w}) < L(w)$ (in fact there is a basin of attraction of a global min). Hence $\tilde{w} \in \Omega_L(\lambda_0)$. By construction since one of the values in $\Omega_L(\lambda_0)$; w, is isolated from the other values in $\Omega_L(\lambda_0)$, $\Omega_L(\lambda_0)$ is a disconnected set. This contradicts the hypothesis. Therefore, we needed w to be a global min. A similar proof can be done for non-strict minima (allowing for $L(w') \geq L(w)$ in Definition 2.4). □

2.1 Linear Network

We now wish to examine the connectedness of $L(w)$ for specific network architectures. As shown in Proposition 2.1, this will also tell us about the existence of poor local minima.

Definition 2.5 (Linear network)**.** The multilayer ($K-1$ layer) linear network considered in this subsection is defined by the following model function, with parametrization split up into W_i for some $i \in [1, K]$, the weight matrices for each layer:

$$f(x,w) = W_K \dots W_1 x \ , \ w = (W_1, \dots, W_K).$$

For such a linear network, in the absence of a regularization term, i.e. $\kappa = 0$, [2] prove that the level sets for both the empirical and oracle loss are connected at each loss level λ. This implies by Proposition 2.1 that there are no poor local minima, as every local minima is a global minima. The proof offered by [2] also allows for a ridge regularization term ($\kappa > 0$, $R(w) = \|w\|^2$) in the single layer $K = 2$ case.

2.2 Half-Rectified Network

Now, most modern architectures use the ReLU and not linear activation, due to its acceleration of the convergence of stochastic gradient descent and sparsity effects ([6], [7]). We would like to examine this non-linear case, which turns out to be a lot more difficult – [2] were only able to get a theoretical result about level set connectedness for the single layer case.

Definition 2.6 (Half-rectified network)**.** The nonlinear ReLU network is defined by

$$f(x,w) = W_K \rho \left(W_{K-1} \rho \left(\dots \rho(W_1 x) \right) \right) \ , \ w = (W_1, \dots, W_K)$$

where $\rho(z) = \max(0, z)$.[2]

Recall our specification of the network in Definition 2.1 where M is the total number of parameters in the network, and N the size of the sample drawn: we consider the network over-parametrized when $M \gg N$.

We now present the main theorem of [2] (Thm 2.4 pg. 6). To obtain connectedness results for such a non-linear network, they considered connecting any two parameters w_A and w_B from parameter space, both with loss L_o less than some $\lambda \in \mathbb{R}$ (i.e. w_A and w_B belonging to the same level set), via a continuous path in the space of parametrizations. They denote this path connecting w_A to w_B by $\gamma_{A,B} : [0,1] \to \mathbb{R}^M$. They then allow for the loss along γ to have some deviation above the level of the desired loss, λ. This deviation amount was shown to depend on how over-parametrized the network is. Indeed, they show that γ can be set to have loss uniformly bounded by $\max(\lambda, \epsilon)$, where ϵ is a parameter that decreases with model over-parametrization, and which has the following asymptotic behavior: $\epsilon = O(M^{\frac{1}{d}})$. Hence, they proved that as

[2] We can implement biases b_i for each layer, as in $y_i(x) = W_i x + b_i$, by simply replacing input vector x with $\overline{x} = (x, 1)$ and replacing each W_i with $\overline{W}_i = \left(\begin{array}{c|c} W_i & b_i \\ \hline 0 & 1 \end{array} \right)$. We will continue using x and W_i for simplicity.

the number of parameters increases the level sets become asymptotically connected. Thus, the main result in [2] is that a single layer half-rectified network is asymptotically connected through over-parametrization.

2.3 Dynamic String Sampling Algorithm to find Paths

The main theorem in Section 2.2 was proven for only the single layer case, but [2] extend its intuition to more general architectures. They conjecture that it should also be "easy" in deeper nets to connect two parametrizations w_A and w_B which lie in the same level set, i.e. both with loss less than some λ.

Their goal was to obtain a numerical estimation of this "ease-of-connection". The authors argue that a good measure is the normalized length of the geodesic $\gamma_{A,B}(t)$:

$$g(w_A, w_B) = \frac{|\gamma_{A,B}(t)|}{|w_A - w_B|}$$

This length represents approximately how much one must alter a linear path between a pair of parametrizations $w_A, w_B \in \mathbb{R}^M$. Convex model functions satisfy $g(w_A, w_B) = 1 \;\forall\; w_A, w_B$, since the geodesic between any points will be a linear interpolation. However, non-convex models will have geodesic length strictly greater than 1. This is justified in the proof of Thm 2.4 (their main theorem) where they construct γ (see appendix B.4 in [2]).

The authors use a dynamic programming based algorithm called Dynamic String Sampling to estimate this geodesic path (finding the exact path is hard, especially given complicated models).

Given two different parametrizations w_A and w_B, each with loss L_o less than some given λ, the algorithm's goal is to search for a path $\gamma_{A,B}(t) \subset \Omega_L(\lambda)$. We present a sketch of their greedy algorithm, the Dynamic String Sampling Algorithm, in the Appendix (Section 5.2).

In [2] the Dynamic String Sampling algorithm is run on multiple regression and classification tasks to connect pairs of parametrizations (gotten from randomly initializing pair of parametrizations which were trained to the same loss value, as explained in the algorithm description in the appendix). The tasks were fairly classic: quadratic and cubic regression tasks using a fully connected multilayer network; MNIST digit recognition and CIFAR10 image recognition tasks using a convolutional neural network; and a word prediction task on the PTB dataset using a reccurent neural network (LSTM) architecture.

For all of these experiments, using their greedy algorithm, the authors were able to connect pairs of parametrizations. Furthermore, they showed that as the loss value at the parametrizations diminish the normalized length of the geodesic grew, hence also providing insight on the geometric regularity of the level sets (see [2] pg. 9).

We will now turn our attention to a more recent paper [1], which also studies the connected structure of the loss surface. The authors focus on connectedness between arbitrary local minima, and extend the experimental work done by [2] by applying a different minima-connecting method called the Automated Nudged Elastic Band (AutoNEB) algorithm.

3 The Automated Nudged Elastic Band: an algorithm for connecting minima

The paper [1] show the construction of continuous paths between arbitrary minima of the non-convex loss functions of state of the art neural network architectures on the dataset CIFAR10 and CIFAR100. This construction is done by the use of a model from molecular statistical mechanics: The Automated Nudged Elastic Band (AutoNEB) algorithm.

Recalling Definition 2.2 of the loss function $L_e(w)$, which depends on network parametrization w (keeping architecture and training set fixed). The goal of AutoNEB is to find the continuous path p^* from parametrizations w_A to w_B through $\mathbb{R}^M$ with the lowest maximum loss:

$$p^*(w_A, w_B) = \underset{p \text{ from } w_A \text{ to } w_B}{\operatorname{argmin}} \{\max_{w \in p} L(w)\}$$

They refer to the parametrization w^* with maximum loss on a path as the "saddle point" of the path (as it is a saddle point of $L(w)$). Their goal is to get a good estimate of this loss to obtain an upper bound on the loss along the entier path.

The AutoNEB algorithm is based on the Nudged Elastic Band (NEB) algorithm ([4]), which essentially finds an approximation to the minimum loss path p^* by bending a straight line segment using gradient 'forces' until there are no more gradients perpendicular to the path. Since this is an approximation of p^*, the point with maximum loss on the returned path may not be the saddle point w^* that we were searching for, but is an upper bounds of the loss at w^*.

We will first present the **mechanical model** behind the idea of the NEB, which is quite interesting, and then move on to AutoNEB.

Consider connecting the two parametrizations $w_1, w_2 \in \mathbb{R}^M$ with a chain of $N+2$ pivots $p_i \in \mathbb{R}^M,\ i = 0, \dots, N+1$ which are each connected via springs of stiffness k. We fix $p_0 = w_1$ and $p_{N+1} = w_2$. We can consider the loss for each pivot $L(p_i)$ as a kind of potential energy, and recall that the potential energy for a spring is $V(x) = kx^2/2$.

We thus consider the total potential energy for the path

$$E(p) = \sum_{i=1}^{N} L(p_i) + \sum_{i=0}^{N} \frac{1}{2} k \|p_{i+1} - p_i\|^2$$

and the goal is to find the path that minimizes $E(p)$ using gradient descent.

The main problem with this formulation is the choice of k, which can be broken down into two cases:

- If k is chosen too small, the first term will dominate and, in areas with high loss, the distances between pivots will become large, (we would actually want them to be small to be able to identify the local maximum clearly) i.e. we have here that the pivots at high loss get pushed away from the saddle point through the action of $\sum_{i=1}^{N} L(p_i)$.
- However, if k is chosen to be too large, the second term will dominate and it will be advantageous to shorten each $p_{i+1} - p_i$, whose norm contributes quadratically, i.e., the spring term $\sum_{i=0}^{N} \frac{1}{2} k \|p_{i+1} - p_i\|^2$ will try to straighten the path as much as possible.

To counter these problems, the NEB algorithm considers the force due to $E(p)$ instead of directly minimizing $E(p)$. This force can be split up into a component coming from the loss, F^L and one from the springs, F^S:

$$F_i = -\nabla_{p_i} E(p) = F_i^L + F_i^S$$

We then modify (*nudge*) these forces for the NEB, such that the loss force only acts perpendicularly to the path (can no longer redistribute pivots to slide down from the saddle point), and the spring force only acts parallel to the path (can only redistribute pivots, and no longer straighten the path). Let $\hat{\tau}_i$ be the local tangent to the path. The NEB force is thus:

$$
\begin{aligned}
F_i^{\text{NEB}} &= F_i^L|_\perp + F_i^S|_\parallel \text{ , where } F_i^L|_\perp \text{ and } F_i^S|_\parallel \text{ are defined as follows:}\\
F_i^L|_\perp &= -\left(\nabla L(p_i) - (\nabla L(p_i)\cdot\hat{\tau}_i)\hat{\tau}_i\right)\\
F_i^S|_\parallel &= \left(F_i^S\cdot\hat{\tau}_i\right)\hat{\tau}_i \text{ where } F_i^S = -k\left(\|p_i - p_{i-1}\| - \|p_{i+1} - p_i\|\right)
\end{aligned}
$$

(spring force opposes unequal distances along the path).

Given this model, the **NEB** algorithm is very simple. We initialize the path $p^{(0)}$ with $N+2$ pivots which satisfy $p_0^{(0)} = w_1$, $p_{N+1}^{(0)} = w_2$ as previously described. We then compute for $t = 1, \dots, T$, where T is a predetermined (large enough) number of iterations, the projected force $F_i = F_i^L|_\perp$ for each pivot $i = 1, \dots, N$ (excluding the fixed ones at $i = 0$ and $i = N+1$ of course), and update all the pivots at the next time step according to $p_i^{(t)} = p_i^{(t-1)} + \gamma F_i$. At the beginning of each iteration, we also use $F_i^S|_\parallel$ to redistribute all pivots. After T iterations, we output the final path $p^{(T)}$.

Note that the computation of the forces for all the pivots can be easily parallelised.

The **AutoNEB** (Automated Nudged Elastic Band Algorithm) is essentially a wrapper for the NEB algorithm: it runs NEB for a small number of iterations T and small number of pivots N. It then checks if the pivots accurately sample the path. If not, new pivots are added at locations where it is estimated that the path requires more accuracy. This is run several times, $t' = 1, \dots, T'$.

While the AutoNEB algorithm is not guaranteed to find the absolute minimum loss path, it is an approximation so it may get stuck in local minimum loss paths. However, it is easy to remove a bad path between w_i and w_j by computing paths between other pairs of minima. As soon as a lower path between w_i and w_j is found by concatenating paths, the bad one can be removed.

Using this algorithm, [1] showed for various CNNs, ResNets and DenseNets on the image classification tasks CIFAR10 and CIFAR100 that the constructed minimum loss path between the minima had constant low loss. This was done by computing the loss at the maximum on the connection paths generated by AutoNEB. Knowing that this maximum is an upper bound to the true path saddle point, they found that the empirical upper bounds were astonishingly close to the loss at the minima themselves. Based on their experiments [1] conjectured the following:

- Neural network loss minima form a connected manifold in parameter space. That is, the part of parameter space with loss beneath some low value forms a connected component.

- As the architecture gets wider and especially deeper, the loss at the saddle points remains close to the loss of the original minima, yielding paths on which the value of the loss function remains low.

Wishing to provide an intuitive explanation of their observations, [1] briefly introduces the notion of resilience: locally, one can slightly perturb a parameter without it leading to significant increase in the loss value. If one can show that locally the majority of directions in the parameter space are flat then this would explain the ability to construct flat paths between arbitrary minima.

This lead to the following grand questions needing to be answered: Do current-day deep architectures have this resilience property? And, what is the relationship between over-parametrization and resilience (i.e. the flatness of the landscape)?

Through the numerous papers providing practical results with qualitative justifications, ([13], [14], [15], [16], [17]) we found one paper which answered our questions while providing some theoretical background to support it ([3]). This paper provides a phenomenological study on the local geometry at a minima by studying the eigenvalues and eigenvectors of the Hessian of the loss function. Moreover, they discuss how flatness of the landscape can be measured by the singularity of the Hessian (ie. the number of trivial eigenvalues). And finally, how the level of the singularity of the Hessian is dependent on the relationship between the number of parameters and the number of samples. They hence provide a possible explanation of the connectedness of the landscape through over-parametrization. Throughout the rest of this paper we will analyze the theoretical approach in [3], and provide additional details on the steps undertaken to arrive to their conclusion.

4 The interplay between the connected structure of the parameter space and the model over-parametrization

We first recall some definitions from Section 2 and establish some new notation:

We denote the loss function by $l(f(w,x),y)$ where $f(w,x)$ is the predictor (model function), and we take l to be a convex function. Hence, the empirical loss is as in Definition 2.2, given by:

$$L(w) := \frac{1}{N}\sum_{i=1}^{N} l(f(w,x_i),y_i).$$

Our goal is to study the spectrum of the Hessian since near its critical points; by Taylor's theorem, the second order approximation provides the best approximation of the function. We recall that a local minimum occurs when all eigenvalues are positive, and if there are negative and positive eigenvalues then we are at a saddle point.

At a critical point, that is when $\|\nabla L(w)\| = 0$, the eigenvectors indicate the directions in which the value of the function locally changes. Further, the magnitude of the corresponding eigenvalues indicates the size of fluctuations in the associated direction. We will next present a useful decomposition of the Hessian.

4.1 The Gauss-Newton decomposition of the Hessian

We will decompose the Hessian as the sum of two matrices: the sample covariance matrix of the gradients of model outputs and the Hessian of the model (i.e. the function that describes the model's outputs) (see [9]).

For ease of notation, for each sample data point $i \in \{1,\dots,N\}$, define the loss function as the composition $l_i \circ f_i : \mathbb{R}^M \to \mathbb{R}^+$ where $l_i : \mathbb{R} \to \mathbb{R}^+$ is *convex*, and where the model function is $f_i : \mathbb{R}^M \to \mathbb{R}$.

Calculating the gradiant and the Hessian of the loss function for a given $i \in \{1,\dots,N\}$ yields:

$$\nabla l_i(f_i(w)) = l_i'(f_i(w))\nabla f_i(w) \tag{1}$$

$$\nabla^2 l_i(f_i(w)) = l_i''(f_i(w))\nabla f_i(w)\nabla f_i(w)^T + l_i'(f(w))\nabla^2 f_i(w) \tag{2}$$

In order to write (2) in the desired form we must first show that a convex function l, has a non-negative second derivative. We note that [3] does not impose any regularity conditions on this convex function. Without assuming regularity this result is subtle and we will provide a proof that an arbitrary measurable convex function on $\mathbb{R}$ has the property that $l''(s) \geq 0$ a.e. on $\mathbb{R}$. (See the Appendix Section 5.1)

Using equation (2) and the fact that we can take the square root of $l_i'' \geq 0$, we can write the Hessian of the loss function in the desired decomposition:

$$\nabla^2 L(w) = \frac{1}{N}\sum_{i=1}^{N}\left[\sqrt{l_i''(f_i(w))}\nabla f_i(w)\right]\left[\sqrt{l_i''(f_i(w))}\nabla f_i(w)\right]^T + \frac{1}{N}\sum_{i=1}^{N} l_i'(f(w))\nabla^2 f_i(w) \quad (3)$$

We have the information that at a point $\bar{w}$ near a critical point, the *average* over the N samples of the gradient is close to zero by equation (1). Therefore, as stated in [3], by making the hypothesis that $l_i'(f(\bar{w}))$ and $\nabla^2 f_i(\bar{w})$ are not correlated we have that the second term in (3) is approximately equal to zero. We can then approximate the Hessian by the $M \times M$ sample covariance matrix:

$$\nabla^2 L(\bar{w}) \approx \frac{1}{N}\sum_{i=1}^{N}\left[\sqrt{l_i''(f_i(\bar{w}))}\nabla f_i(\bar{w})\right]\left[\sqrt{l_i''(f_i(\bar{w}))}\nabla f_i(\bar{w})\right]^T \quad (4)$$

Note that in the case where the model function is linear in w we have that $\nabla^2 f_i(w) = 0 \ \forall i \in \{1, \ldots, N\}$, in which case we do have an equality in equation (4). An example of this will be given at the end.

Our goal is now to prove that when there are more parameters than samples the covariance matrix in (4) leads to degeneracy. We explain the use of this result below.

The first step is to prove that the covariance matrix is a weighted sum of N rank one matrices.

Definition 4.1. A rank one matrix A has the property that $Rank(A) = 1$, i.e. the range of A is one-dimensional.

Lemma 4.1. *An $m \times n$ matrix A is a rank-one matrix if and only if $A = vw^T$ for some $v \in \mathbb{R}^m$ and $w \in \mathbb{R}^n$.*

Proof. Suppose that $A = vw^T$ and let $u \in \mathbb{R}^n$, then $Au = vw^T u = (u \cdot w)v$. This means that A maps every vector $u \in \mathbb{R}^n$ to a scalar multiple of v, hence its range is one-dimensional, i.e. A is of rank one.

On the other hand, if the range of A is one-dimensional, this means that A maps every vector $u \in \mathbb{R}^n$ to a multiple of a vector, say $v \in \mathbb{R}^m$. In particular, $A\vec{e}_1 = \nu_1 v$, $A\vec{e}_2 = \nu_2 v, \ldots, A\vec{e}_n = \nu_n v$ where $\vec{e}_i$ is the unit vector with value 1 at the ith component and 0 elsewhere and ν_i is some constant in $\mathbb{R}$. Hence the $i-th$ column of the matrix A can be written as $\nu_i v, i = 1, \ldots, n$, which means that A can be written as

$$A = vw^T, \text{ with } w^T = (\nu_1, \ldots, \nu_n) \qquad \square$$

We apply this lemma to the right hand side of (4) with $v = \left[\sqrt{l_i''(f_i(\bar{w}))}\nabla f_i(\bar{w})\right]$ and $w^T = v^T$. Hence, we have proven that each matrix in the sum in (4) is a $M \times M$ rank one matrix. It then follows that each of these rank one matrices have $M - 1$ trivial eigenvalues. Indeed, this is the case since their null space is of dimension $M - 1$, hence the geometric multiplicity of the eigenvalue $\lambda = 0$ is $M - 1$.

We are now ready to prove the crucial claim given in [3] which asserts that there are at least $M - N$ zero eigenvalues of the right hand side of (4).

Theorem 4.2. *Let $A_i, i = 1, \ldots, N$ be N rank-one $M \times M$ matrices. Then the matrix $\sum_{i=1}^N A_i$ has at least $M - N$ zero eigenvalues.*

Proof. We will prove this result by induction. We do the case $N = 2$ first. We will be using the formula for the sum of subspaces (see [11] thm 4.8, or a GRE math textbook):

$$\dim(A \cap B) + \dim(A + B) = \dim(A) + \dim(B),$$

with $A = \ker A_1$ and $B = \ker A_2$. Since A_i are rank-one matrices, their null-space has dimension $M - 1$ (by the rank-nullity theorem), and since $\dim(A + B) \leq M$ (as each null space has dimension at most M,) we conclude

$$\dim(A \cap B) \geq 2(M - 1) - M = M - 2.$$

Now assume by induction that

$$\dim\left(\bigcap_{i=1}^{N-1} \ker A_i\right) \geq M - (N - 1).$$

Using again that $\dim\left(\sum_{i=1}^N \ker A_i\right) \leq M$, we have:

$$\begin{aligned}\dim\left(\left(\bigcap_{i=1}^{N-1} \ker A_i\right) \cap \ker A_N\right) &\geq \dim\left(\bigcap_{i=1}^{N-1} \ker A_i\right) + \dim(\ker A_N) - M \\ &\geq M - (N - 1) + M - 1 - M = M - N.\end{aligned}$$

□

Using this theorem with the rank-one matrices:

$$A_i = \frac{1}{N}\left(\left[\sqrt{l_i''(f_i(w))}\nabla f_i(w)\right]\left[\sqrt{l_i''(f_i(w))}\nabla f_i(w)\right]^T\right),$$

we conclude that the right-hand side of equation (4) has at least $M - N$ trivial eigenvalues.

We now see that as M exceeds N, the RHS of (4) becomes 'more singular'. If we can obtain an equality in (4), this would imply that through over-parametrization the level of the singularity of the Hessian increases. Indeed, the eigenvalues of the Hessian determine the size of local changes in the loss function, hence this cluster of trivial eigenvalues would explain the flatness of the landscape near a critical point. Hence, providing an explanation of the results on the connected structure of the landscape found in [1] and [2], since the landscape would be flat at all critical points. Moreover, it could explain the empirical results found by [1] of saddle points of low loss found on the paths connecting arbitrary minima.

Next, we provide an example for which we have equality in (4). (See Appendix B in [3]) We let $M = d$ and take the following linear model function: $f(w, x) = w \cdot x$. Let the loss function be given by $l(s, y) = -y \log \frac{1}{1+e^{-s}} - (1 - y) \log(1 - \frac{1}{1+e^{-s}})$. This is a convex function in s since l_s is increasing. Finally, we take a single neuron and choose the sigmoid function as our activation function. Then by equation (3), the Hessian of the loss function is equal to the covariance matrix of the gradients of model outputs. We note that in this case we have a good understanding of the spectrum of the covariance matrix: its eigenvalues are distributed according to Marchenko-Pastur law. (See [3] Appendix B).

We lastly mention that most of the paper [3] focuses on experimental results of the spectrum of the Hessian, and use the generalized Gauss-Newton decomposition of the Hessian to suggest that in practical applications one can expect to have a cluster of trivial eigenvalues.

5 Appendix

5.1 Rigorous results on convex functions

We will prove that a measurable convex function on $\mathbb{R}$ has a non-negative second derivative except possibly on a set of measure zero. This fact will be obtain in several steps. The first step is to prove that an increasing function on $\mathbb{R}$ is continuous except possibly on a countable set.

Lemma 5.1. *Let h be an increasing function on $\mathbb{R}$, then h is continuous except possibly on a countable set.*

Proof. One can characterize the set of discontinuities of h by $D = \{x \in \mathbb{R} \mid h(x^-) < h(x^+)\}$ (where $h(x^-) = \lim_{y \to x, y<x} h(y)$) since an increasing function can only have jump discontinuities. By the density of $\mathbb{Q}$ in $\mathbb{R}$, $\forall x \in D, \exists q_x \in \mathbb{Q}$ such that $q_x \in (h(x^-), h(x^+))$. Since h is increasing, taking $x_1, x_2 \in D$ with $x_1 < x_2$, we have $h(x_1^-) < h(x_1^+) \leq h(x_2^-) < h(x_2^+)$. Hence by construction, $\{q_x\}_{x \in D}$ is a disjoint collection in $\mathbb{Q}$. It follows that we have found an injective map from D to $\mathbb{Q}$, and hence that $D \subseteq \mathbb{Q}$, namely that the set of discontinuities of h is at most countable. □

Next we prove that if a measurable function l is convex then l is differentiable except on a countable set and that on the complement of that set, l' is increasing.

Proposition 5.2. *Let l be a measurable function that is convex on $\mathbb{R}$, then l is differentiable except on a countable set C_1 and l' is an increasing function on $\mathbb{R} \setminus C_1$.*

Proof. First we note that since l is convex (See [10] pg. 130), we have:

$$\frac{l(x_1) - l(x)}{x_1 - x} \leq \frac{l(x_2) - l(x_1)}{x_2 - x_1} \leq \frac{l(x_2) - l(x)}{x_2 - x}, \text{ for } x_1 < x < x_2. \tag{5}$$

It follows in particular, and using also the monotone convergence theorem for sequences, that the one-sided derivatives of $l(s)$ exist, i.e. $l'(s^+), l'(s^-)$ exists $\forall s \in \mathbb{R}$. Using again equation (5) where $x_1 < x$ yields the following useful inequalities:

$$l'(x_1^-) \leq l'(x_1^+) \leq \frac{l(x) - l(x_1)}{x - x_1} \leq l'(x^-) \leq l'(x^+). \tag{6}$$

In particular $l'(x^+)$ and $l'(x^-)$ as functions of x are increasing functions. From the above Lemma, this means that they are continuous except on a countable set that we will call C_1. Now let $x_1 \in \mathbb{R} \setminus C_1$, using (6) and the continuity of $l'(x^-)$ at x_1, we obtain that $\lim_{x \to x_1} l'(x^-) = l'(x_1^-)$. Therefore by (6), we have

$$l'(x_1^-) \leq l'(x_1^+) \leq l'(x_1^-),$$

so that l' is differentiable at x_1 and since x_1 was arbitrary, we conclude that l' is differentiable in $\mathbb{R} \setminus C_1$. Finally, it follows from (6) that l' is increasing on $\mathbb{R} \setminus C_1$. □

Finally, using Lebesgue's Theorem on increasing functions (see pg. 112 in [10]), we conclude that $l'(s)$ being an increasing function on $\mathbb{R} \setminus C_1$ is in fact differentiable except on a set of measure zero. Hence the second derivative of l exists almost everywhere on $\mathbb{R}$ (since the finite unions of sets of measure zero is zero). By the definition of the derivative it then follows that since l' is increasing a.e. on $\mathbb{R}$ that $l''(s) \geq 0$ a.e. on $\mathbb{R}$.

5.2 Sketch of the Dynamic String Sampling Algorithm

Starting by initializing parameters $w_1 \neq w_2$ randomly, we separately train the networks $f(x_i, w_1)$ and $f(x_i, w_2)$ using stochastic gradient descent to λ. The algorithm then recursively builds a string of parametrizations $\{w_k\}$ which continuously connect each w_1 to w_2 in the following manner:

Let $\tilde{\gamma}_{a,b}(t) = t \cdot a + (1-t) \cdot b,\ t \in (0,1)$ be the linear interpolation between a and b.

- Pick a w_3 on the linear interpolation between w_1 and w_2, i.e. a $t^* \in (0,1)$ either by taking $t^* = 0.5$ (midpoint) or a local maximum of the interpolated loss curve: t^* such that $\frac{d}{dt}L(\tilde{\gamma}_{w_1,w_2}(t^*)) = 0$ (the authors used this option).[3]
- The new parametrization $w_3 = \tilde{\gamma}_{w_1,w_2}(t^*)$ is added to the string, and stochastic gradient descent is performed on $f(x_i, w_3)$ until its loss is below λ.
- For each pair $[w_1, w_3]$ and $[w_2, w_3]$, compute the maximum value of the interpolated loss curve $\max_t[L(\tilde{\gamma}_{w_i,w_3}(t))]$, $i = 1, 2$. If this value is greater than λ, the string building recursively runs for that pair, adding to the global string of parametrizations.

In the end, the algorithm outputs a string $\{w_k\}$ that continuously connects w_1 to w_2 such that the linearly interpolated loss $\max_t[L(tw_i + (1-t)w_j)]$ for each pair of neighbouring parametrizations $w_{\{i,j\}}$ is below λ.

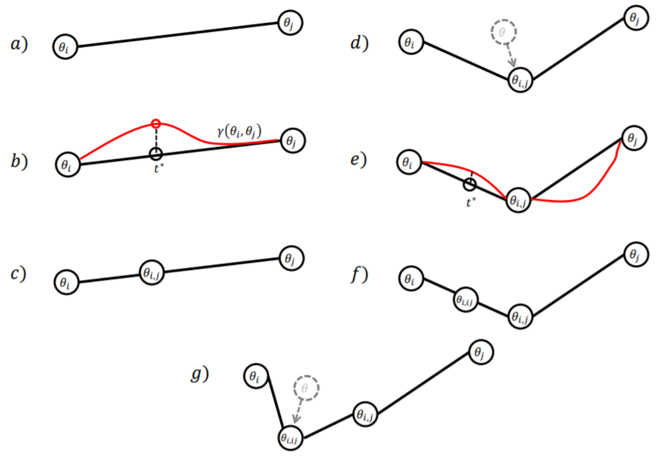

Figure 1: Cartoon of the Dynamic String Sampling algorithm. Starting at *a*) with the two parametrizations $\theta_{\{i,j\}}$ (both with $L(\theta_{i,j}) < \lambda$) to be connected, at *b*), the interpolated loss function (red curve) is computed and we find the local maximum at t^*. *c*), The local max $\theta_{ij} = \theta_3$ is added to the string of parametrizations and then, *d*), trained using stochastic gradient descent to have loss value less than λ. *e*), new interpolated loss curves are computed between the parametrizations, and steps *f*) and *g*) show the recursion of the algorithm: the steps the same as *c*) and *d*) but on the interval $[\theta_i, \theta_{ij}]$.

It is important to note that this algorithm can only confirm whether w_A and w_B are connected, but cannot guarantee that they are disconnected if the algorithm fails to converge. So for cases that are not easily convergent, one has to rely on heuristic arguments to choose when exactly to stop the algorithm. In practice, for the problems the authors chose to examine, convergence was not a problem.

References

[1] Draxler, F., Veschgini K., Salmhofer, M. and Hamprechtet F. A. *Essentially No Barriers in Neural Network Energy Landscape.* PMLR 80, 2018.

[3] Choosing t^* at a local maximum provides a small increase in algorithm runtime, but can be unstable, which is why [2] noted down the two options.

[2] Freeman, D. C. and Bruna J. *Topology and Geometry of Half-Rectified Network Optimization.* arXiv preprint arXiv:1611.01540, 2016.

[3] Sagun, L., Evci, U., Guney V.U., Dauphin Y. and Bottou L. *Empirical Analysis of the Hessian of Over-Parametrized Neural Networks.* ICLR, arXiv:1706.04454v3, 2018.

[4] Jónsson G., Mills G., and Jacobsen K. W. *Nudged elastic band method for finding minimum energy paths of transitions.* Classical and quantum dynamics in condensed phase simulations, pp. 385-404. World Scientific, 1998.

[5] Kawaguchi, K. *Deep Learning without Poor Local Minima.* arXiv preprint arXiv:1502.03167, 2016.

[6] Krizhevsky A., Sutskever I., and Hinton, G. E. *ImageNet classification with deep convolutional neural networks.* Proceedings of the 25th International Conference on Neural Information Processing Systems (NIPS'12), F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger (Eds.), Vol. 1. 1097-1105. 2012.

[7] Glorot, X., Bordes, A. and Bengio, Y. *Deep Sparse Rectifier Neural Networks.* Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics (PMLR). 15:315-323, 2011.

[8] Hochreiter, S. and Schmidhuber, J. *Flat minima.* Neural Computation, 9(1):142, 1997.

[9] LeCun, Y., Bottou, L., Orr, G., and Muller, K. *Efficient backprop.* Neural Networks: Tricks of the trade, Orr, G. and K., Muller (Eds.). Springer, 1998.

[10] Royden, H.L. and Fitzpatrick P.M. Real Analysis, Fourth Edition. ISBN 978-0-13-143747-0, 2010.

[11] Nering, E.D. Linear Algebra and Matrix Theory, 2nd edition. Wiley and Sons, 1970.

[12] Bloemendal, A., Knowles, A., Yau H.-T. and Yin J. *On the principal components of sample covariance matrices. Probability Theory and Related Fields*, 164(1-2):459-552, 2016.

[13] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. *Dropout: A simple way to prevent neural networks from overfitting.* The Journal of Machine Learning Research, 15(1):19291958, 2014.

[14] Hansen, L. K. and Salamon, P. *Neural network ensembles.* IEEE transactions on pattern analysis and machine intelligence, 12(10):9931001, 1990.

[15] Liu, Z., Li, J., Shen, Z., Huang, G., Yan, S., and Zhang, C. *Learning efficient convolutional networks through network slimming.* Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 27362744, 2017.

[16] Garipov, T., Izmailov, P., Podoprikhin, D., Vetrov, D. P., and Wilson, A. G. *Loss Surfaces, Mode Connectivity, and Fast Ensembling of DNNs.* arXiv Eprint arXiv:1802.10026, 2018.

[17] Li, H., Xu, Z., Taylor, G., and Goldstein, T. Visualizing the loss landscape of neural nets. arXiv preprint arXiv:1712.09913, 2017.