# Sharp Minima Can Generalize For Deep Nets

**Laurent Dinh** [1] **Razvan Pascanu** [2] **Samy Bengio** [3] **Yoshua Bengio** [1 4]

## Abstract

Despite their overwhelming capacity to overfit, deep learning architectures tend to generalize relatively well to unseen data, allowing them to be deployed in practice. However, explaining why this is the case is still an open area of research. One standing hypothesis that is gaining popularity, e.g. Hochreiter & Schmidhuber (1997); Keskar et al. (2017), is that the flatness of minima of the loss function found by stochastic gradient based methods results in good generalization. This paper argues that most notions of flatness are problematic for deep models and can not be directly applied to explain generalization. Specifically, when focusing on deep networks with rectifier units, we can exploit the particular geometry of parameter space induced by the inherent symmetries that these architectures exhibit to build equivalent models corresponding to arbitrarily sharper minima. Furthermore, if we allow to reparametrize a function, the geometry of its parameters can change drastically without affecting its generalization properties.

## 1 Introduction

Deep learning techniques have been very successful in several domains, like *object recognition in images* (e.g Krizhevsky et al., 2012; Simonyan & Zisserman, 2015; Szegedy et al., 2015; He et al., 2016), *machine translation* (e.g. Cho et al., 2014; Sutskever et al., 2014; Bahdanau et al., 2015; Wu et al., 2016; Gehring et al., 2016) and *speech recognition* (e.g. Graves et al., 2013; Hannun et al., 2014; Chorowski et al., 2015; Chan et al., 2016; Collobert et al., 2016). Several arguments have been brought forward to justify these empirical results. From a representational point of view, it has been argued that deep networks can efficiently approximate certain functions (e.g. Montufar et al., 2014; Raghu et al., 2016). Other works (e.g Dauphin et al., 2014; Sagun et al., 2014; Choromanska et al., 2015) have looked at the structure of the error surface to analyze how trainable these models are. Finally, another point of discussion is how well these models can generalize (Nesterov & Vial, 2008; Keskar et al., 2017; Zhang et al., 2017). These correspond, respectively, to low *approximation*, *optimization* and *estimation* error as described by Bottou (2010).

Our work focuses on the analysis of the estimation error. In particular, different approaches had been used to look at the question of why *stochastic gradient descent* results in solutions that generalize well (Bottou & LeCun, 2005; Bottou & Bousquet, 2008). For example, Duchi et al. (2011); Nesterov & Vial (2008); Hardt et al. (2016); Bottou et al. (2016); Gonen & Shalev-Shwartz (2017) rely on the concept of *stochastic approximation* or *uniform stability* (Bousquet & Elisseeff, 2002). Another conjecture that was recently (Keskar et al., 2017) explored, but that could be traced back to Hochreiter & Schmidhuber (1997), relies on the geometry of the loss function around a given solution. It argues that flat minima, for some definition of flatness, lead to better generalization. Our work focuses on this particular conjecture, arguing that there are critical issues when applying the concept of flat minima to deep neural networks, which require rethinking what flatness actually means.

While the concept of flat minima is not well defined, having slightly different meanings in different works, the intuition is relatively simple. If one imagines the error as a one-dimensional curve, a minimum is flat if there is a wide region around it with roughly the same error, otherwise the minimum is sharp. When moving to higher dimensional spaces, defining flatness becomes more complicated. In Hochreiter & Schmidhuber (1997) it is defined as the size of the connected region around the minimum where the training loss is relatively similar. Chaudhari et al. (2017) relies, in contrast, on the curvature of the second order structure around the minimum, while Keskar et al. (2017) looks at the maximum loss in a bounded neighbourhood of the minimum. All these works rely on the fact that flatness results in robustness to low precision arithmetic or noise in the parameter space, which, using an *minimum description length*-based argument, suggests a low expected overfitting.

[1]Université of Montréal, Montréal, Canada [2]DeepMind, London, United Kingdom [3]Google Brain, Mountain View, United States [4]CIFAR Senior Fellow. Correspondence to: Laurent Dinh <laurent.dinh@umontreal.ca>.

*Proceedings of the $34^{th}$ International Conference on Machine Learning*, Sydney, Australia, PMLR 70, 2017. 

However, several common architectures and parametrizations in deep learning are already at odds with this conjecture, requiring at least some degree of refinement in the statements made. In particular, we show how the geometry of the associated parameter space can alter the ranking between prediction functions when considering several measures of *flatness/sharpness*. We believe the reason for this contradiction stems from the Bayesian arguments about KL-divergence made to justify the generalization ability of flat minima (Hinton & Van Camp, 1993). Indeed, *Kullback-Liebler divergence is invariant to change of parameters whereas the notion of "flatness" is not*. The demonstrations of Hochreiter & Schmidhuber (1997) are approximately based on a Gibbs formalism and rely on strong assumptions and approximations that can compromise the applicability of the argument, including the assumption of a discrete function space.

## 2 Definitions of flatness/sharpness

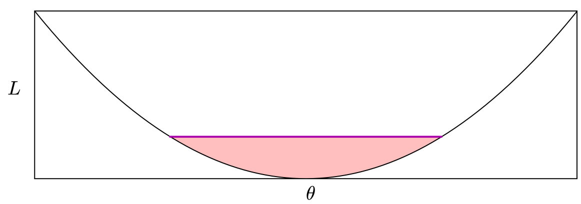



Figure 1: An illustration of the notion of flatness. The loss $L$ as a function of $\theta$ is plotted in black. If the height of the red area is $\epsilon$, the width will represent the volume $\epsilon$-flatness. If the width is $2\epsilon$, the height will then represent the $\epsilon$-sharpness. Best seen with colors.

For conciseness, we will restrict ourselves to supervised scalar output problems, but several conclusions in this paper can apply to other problems as well. We will consider a function $f$ that takes as input an element $x$ from an input space $\mathcal{X}$ and outputs a scalar $y$. We will denote by $f_\theta$ the prediction function. This prediction function will be parametrized by a parameter vector $\theta$ in a parameter space $\Theta$. Often, this prediction function will be over-parametrized and two parameters $(\theta, \theta') \in \Theta^2$ that yield the same prediction function everywhere, $\forall x \in \mathcal{X}, f_\theta(x) = f_{\theta'}(x)$, are called *observationally equivalent*. The model is trained to minimize a continuous loss function $L$ which takes as argument the prediction function $f_\theta$. We will often think of the loss $L$ as a function of $\theta$ and adopt the notation $L(\theta)$.

The notion of flatness/sharpness of a minimum is relative, therefore we will discuss metrics that can be used to compare the relative flatness between two minima. In this section we will formalize three used definitions of flatness in the literature.

Hochreiter & Schmidhuber (1997) defines a flat minimum as "a large connected region in weight space where the error remains approximately constant". We interpret this formulation as follows:

**Definition 1.** *Given $\epsilon > 0$, a minimum $\theta$, and a loss $L$, we define $C(L, \theta, \epsilon)$ as the largest (using inclusion as the partial order over the subsets of $\Theta$) connected set containing $\theta$ such that $\forall \theta' \in C(L, \theta, \epsilon), L(\theta') < L(\theta) + \epsilon$. The $\epsilon$-flatness will be defined as the volume of $C(L, \theta, \epsilon)$. We will call this measure the volume $\epsilon$-flatness.*

In Figure 1, $C(L, \theta, \epsilon)$ will be the purple line at the top of the red area if the height is $\epsilon$ and its volume will simply be the length of the purple line.

Flatness can also be defined using the local curvature of the loss function around the minimum if it is a critical point [1]. Chaudhari et al. (2017); Keskar et al. (2017) suggest that this information is encoded in the eigenvalues of the Hessian. However, in order to compare how flat one minimum versus another, the eigenvalues need to be reduced to a single number. Here we consider the *spectral norm and trace of the Hessian*, two typical measurements of the eigenvalues of a matrix.

Additionally Keskar et al. (2017) defines the notion of $\epsilon$-sharpness. In order to make proofs more readable, we will slightly modify their definition. However, because of norm equivalence in finite dimensional space, our results will transfer to the original definition in full space as well. Our modified definition is the following:

**Definition 2.** *Let $B_2(\epsilon, \theta)$ be an Euclidean ball centered on a minimum $\theta$ with radius $\epsilon$. Then, for a non-negative valued loss function $L$, the $\epsilon$-sharpness will be defined as proportional to*

$$\frac{\max_{\theta' \in B_2(\epsilon, \theta)} \big(L(\theta') - L(\theta)\big)}{1 + L(\theta)}.$$

In Figure 1, if the width of the red area is $2\epsilon$ then the height of the red area is $\max_{\theta' \in B_2(\epsilon, \theta)} \big(L(\theta') - L(\theta)\big)$.

$\epsilon$-sharpness can be related to the spectral norm of the Hessian. Indeed, a second-order Taylor expansion of $L$ around a critical point minimum is written

$$\begin{aligned} L(\theta') = \; & L(\theta) + \frac{1}{2}\,(\theta' - \theta)\,(\nabla^2 L)(\theta)(\theta' - \theta)^T \\ & + o(\|\theta' - \theta\|_2^2). \end{aligned}$$

In this second order approximation, the $\epsilon$-sharpness at $\theta$

[1]In this paper, we will often assume that is the case when dealing with Hessian-based measures in order to have them well-defined.

would be

$$\frac{\left|\!\left|\!\left|(\nabla^2 L)(\theta)\right|\!\right|\!\right|_2 \epsilon^2}{2\big(1 + L(\theta)\big)}.$$

# 3 Properties of Deep Rectified Networks

Before moving forward to our results, in this section we first introduce the notation used in the rest of paper. Most of our results, for clarity, will be on the deep rectified feedforward networks with a linear output layer that we describe below, though they can easily be extended to other architectures (e.g. convolutional, etc.).

**Definition 3.** *Given $K$ weight matrices $(\theta_k)_{k\leq K}$ with $n_k = dim\big(vec(\theta_k)\big)$ and $n = \sum_{k=1}^{K} n_k$, the output $y$ of a deep rectified feedforward networks with a linear output layer is:*

$$y = \phi_{rect}\Big(\phi_{rect}\big(\cdots \phi_{rect}(x\cdot\theta_1)\cdots\big)\cdot\theta_{K-1}\Big)\cdot\theta_K,$$

*where*

- *$x$ is the input to the model, a high-dimensional vector*
- *$\phi_{rect}$ is the rectified elementwise activation function (Jarrett et al., 2009; Nair & Hinton, 2010; Glorot et al., 2011), which is the positive part $(z_i)_i \mapsto (\max(z_i, 0))_i$.*
- *vec reshapes a matrix into a vector.*

Note that in our definition we excluded the bias terms, usually found in any neural architecture. This is done mainly for convenience, to simplify the rendition of our arguments. However, the arguments can be extended to the case that includes biases (see Appendix B) . Another choice is that of the linear output layer. Having an output activation function does not affect our argument either: since the loss is a function of the output activation, it can be rephrased as a function of linear pre-activation.

Deep rectifier models have certain properties that allows us in section 4 to arbitrary manipulate the flatness of a minimum.

An important topic for optimization of neural networks is understanding the non-Euclidean geometry of the parameter space as imposed by the neural architecture (see, for example Amari, 1998). In principle, when we take a step in parameter space what we expect to control is the change in the behavior of the model (i.e. the mapping of the input $x$ to the output $y$). In principle we are not interested in the parameters per se, but rather only in the mapping they represent.

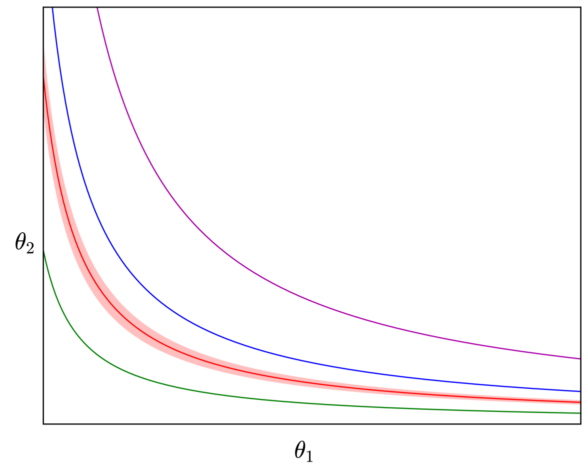



Figure 2: An illustration of the effects of non-negative homogeneity. The graph depicts level curves of the behavior of the loss $L$ embedded into the two dimensional parameter space with the axis given by $\theta_1$ and $\theta_2$. Specifically, each line of a given color corresponds to the parameter assignments $(\theta_1, \theta_2)$ that result observationally in the same prediction function $f_\theta$. Best seen with colors.

If one defines a measure for the change in the behavior of the model, which can be done under some assumptions, then, it can be used to define, at any point in the parameter space, a metric that says what is the equivalent change in the parameters for a unit of change in the behavior of the model. As it turns out, for neural networks, this metric is not constant over $\Theta$. Intuitively, the metric is related to the curvature, and since neural networks can be highly nonlinear, the curvature will not be constant. See Amari (1998); Pascanu & Bengio (2014) for more details. Coming back to the concept of flatness or sharpness of a minimum, this metric should define the flatness.

However, the geometry of the parameter space is more complicated. Regardless of the measure chosen to compare two instantiations of a neural network, because of the structure of the model, it also exhibits a large number of symmetric configurations that result in exactly the same behavior. Because the rectifier activation has the non-negative homogeneity property, as we will see shortly, one can construct a continuum of points that lead to the same behavior, hence the metric is singular. Which means that one can exploit these directions in which the model stays unchanged to shape the neighbourhood around a minimum in such a way that, by most definitions of flatness, this property can be controlled. See Figure 2 for a visual depiction, where the flatness (given here as the distance between the different level curves) can be changed by moving along the curve.

Let us redefine, for convenience, the *non-negative homogeneity* property (Neyshabur et al., 2015; Lafond et al., 2016) below. Note that beside this property, the reason for studying the rectified linear activation is for its widespread adoption (Krizhevsky et al., 2012; Simonyan & Zisserman, 2015; Szegedy et al., 2015; He et al., 2016).

**Definition 4.** *A given a function $\phi$ is* non-negative homogeneous *if*

$$\forall (z, \alpha) \in \mathbb{R} \times \mathbb{R}^+, \phi(\alpha z) = \alpha \phi(z)$$

.

**Theorem 1.** *The rectified function $\phi_{rect}(x) = \max(x, 0)$ is non-negative homogeneous.*

*Proof.* Follows trivially from the constraint that $\alpha > 0$, given that $x > 0 \Rightarrow \alpha x > 0$, iff $\alpha > 0$. □

For a deep rectified neural network it means that:

$$\phi_{rect}\big(x \cdot (\alpha\theta_1)\big) \cdot \theta_2 = \phi_{rect}(x \cdot \theta_1) \cdot (\alpha\theta_2),$$

meaning that for this one (hidden) layer neural network, the parameters $(\alpha\theta_1, \theta_2)$ is observationally equivalent to $(\theta_1, \alpha\theta_2)$. This observational equivalence similarly holds for convolutional layers.

Given this non-negative homogeneity, if $(\theta_1, \theta_2) \neq (0, 0)$ then $\big\{(\alpha\theta_1, \alpha^{-1}\theta_2), \alpha > 0\big\}$ is an infinite set of observationally equivalent parameters, inducing a strong non-identifiability in this learning scenario. Other models like *deep linear networks* (Saxe et al., 2013), *leaky rectifiers* (He et al., 2015) or *maxout networks* (Goodfellow et al., 2013) also have this non-negative homogeneity property.

In what follows we will rely on such transformations, in particular we will rely on the following definition:

**Definition 5.** *For a single hidden layer rectifier feedforward network we define the family of transformations*

$$T_\alpha : (\theta_1, \theta_2) \mapsto (\alpha\theta_1, \alpha^{-1}\theta_2)$$

*which we refer to as a $\alpha$-scale transformation.*

Note that a $\alpha$-scale transformation will not affect the generalization, as the behavior of the function is identical. Also while the transformation is only defined for a single layer rectified feedforward network, it can trivially be extended to any architecture having a single rectified network as a submodule, e.g. a deep rectified feedforward network. For simplicity and readability we will rely on this definition.

# 4 Deep Rectified networks and flat minima

In this section we exploit the resulting strong non-identifiability to showcase a few shortcomings of some definitions of flatness. Although $\alpha$-scale transformation does not affect the function represented, it allows us to significantly decrease several measures of flatness. For another definition of flatness, $\alpha$-scale transformation show that all minima are equally flat.

## 4.1 Volume $\epsilon$-flatness

**Theorem 2.** *For a one-hidden layer rectified neural network of the form*

$$y = \phi_{rect}(x \cdot \theta_1) \cdot \theta_2,$$

*and a minimum $\theta = (\theta_1, \theta_2)$, such that $\theta_1 \neq 0$ and $\theta_2 \neq 0$, $\forall \epsilon > 0$ $C(L, \theta, \epsilon)$ has an infinite volume.*

We will not consider the solution $\theta$ where any of the weight matrices $\theta_1, \theta_2$ is zero, $\theta_1 = 0$ or $\theta_2 = 0$, as it results in a constant function which we will assume to give poor training performance. For $\alpha > 0$, the $\alpha$-scale transformation $T_\alpha : (\theta_1, \theta_2) \mapsto (\alpha\theta_1, \alpha^{-1}\theta_2)$ has Jacobian determinant $\alpha^{n_1 - n_2}$, where once again $n_1 = dim\big(\text{vec}(\theta_1)\big)$ and $n_2 = dim\big(\text{vec}(\theta_2)\big)$. Note that the Jacobian determinant of this linear transformation is the change in the volume induced by $T_\alpha$ and $T_\alpha \circ T_\beta = T_{\alpha\beta}$. We show below that there is a connected region containing $\theta$ with infinite volume and where the error remains approximately constant.

*Proof.* We will first introduce a small region with approximately constant error around $\theta$ with non-zero volume. Given $\epsilon > 0$ and if we consider the loss function continuous with respect to the parameter, $C(L, \theta, \epsilon)$ is an open set containing $\theta$. Since we also have $\theta_1 \neq 0$ and $\theta_2 \neq 0$, let $r > 0$ such that the $\mathcal{L}_\infty$ ball $B_\infty(r, \theta)$ is in $C(L, \theta, \epsilon)$ and has empty intersection with $\{\theta', \theta'_1 = 0\}$. Let $v = (2r)^{n_1 + n_2} > 0$ the volume of $B_\infty(r, \theta)$.

Since the Jacobian determinant of $T_\alpha$ is the multiplicative change of induced by $T_\alpha$, the volume of $T_\alpha\big(B_\infty(r, \theta)\big)$ is $v\alpha^{n_1 - n_2}$. If $n_1 \neq n_2$, we can arbitrarily grow the volume of $T_\alpha\big(B_\infty(r, \theta)\big)$, with error within an $\epsilon$-interval of $L(\theta)$, by having $\alpha$ tends to $+\infty$ if $n_1 > n_2$ or to 0 otherwise.

If $n_1 = n_2, \forall \alpha' > 0, T_{\alpha'}\Big(B_\infty(r, \theta)\Big)$ has volume $v$. Let $C' = \bigcup_{\alpha' > 0} T_{\alpha'}\Big(B_\infty(r, \theta)\Big)$. $C'$ is a connected region where the error remains approximately constant, i.e. within an $\epsilon$-interval of $L(\theta)$.

Let $\alpha = 2\frac{\|\theta_1\|_\infty + r}{\|\theta_1\|_\infty - r}$. Since

$$B_\infty(r, \theta) = B_\infty(r, \theta_1) \times B_\infty(r, \theta_2),$$

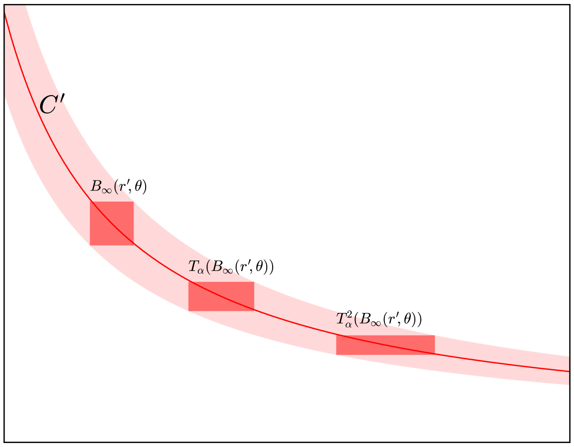



Figure 3: An illustration of how we build different disjoint volumes using $T_\alpha$. In this two-dimensional example, $T_\alpha\big(B_\infty(r',\theta)\big)$ and $B_\infty(r',\theta)$ have the same volume. $B_\infty(r',\theta), T_\alpha\big(B_\infty(r',\theta)\big), T_\alpha^2\big(B_\infty(r',\theta)\big), \ldots$ will therefore be a sequence of disjoint constant volumes. $C'$ will therefore have an infinite volume. Best seen with colors.

where $\times$ is the Cartesian set product, we have

$$T_\alpha\big(B_\infty(r,\theta)\big) = B_\infty(\alpha r, \alpha\theta_1) \times B_\infty(\alpha^{-1}r, \alpha^{-1}\theta_2).$$

Therefore, $T_\alpha\big(B_\infty(r,\theta)\big) \cap B_\infty(r,\theta) = \emptyset$ (see Figure 3).

Similarly, $B_\infty(r,\theta), T_\alpha\big(B_\infty(r,\theta)\big), T_\alpha^2\big(B_\infty(r,\theta)\big), \ldots$ are disjoint and have volume $v$. We have also $T_\alpha^k\big(B_\infty(r',\theta)\big) = T_{\alpha^k}\big(B_\infty(r',\theta)\big) \in C'$. The volume of $C'$ is then lower bounded by $0 < v + v + v + \cdots$ and is therefore infinite. $C(L,\theta,\epsilon)$ has then infinite volume too, making the volume $\epsilon$-flatness of $\theta$ infinite. $\square$

This theorem can generalize to rectified neural networks in general with a similar proof. Given that every minimum has an infinitely large region (volume-wise) in which the error remains approximately constant, that means that every minimum would be infinitely flat according to the volume $\epsilon$-flatness. Since all minima are equally flat, it is not possible to use volume $\epsilon$-flatness to gauge the generalization property of a minimum.

## 4.2 Hessian-based measures

The non-Euclidean geometry of the parameter space, coupled with the manifolds of observationally equal behavior of the model, allows one to move from one region of the parameter space to another, changing the curvature of the model without actually changing the function. This approach has been used with success to improve optimization, by moving from a region of high curvature to a region of well behaved curvature (e.g. Desjardins et al., 2015; Salimans & Kingma, 2016). In this section we look at two widely used measures of the Hessian, the spectral radius and trace, showing that either of these values can be manipulated without actually changing the behavior of the function. If the flatness of a minimum is defined by any of these quantities, then it could also be easily manipulated.

**Theorem 3.** *The gradient and Hessian of the loss $L$ with respect to $\theta$ can be modified by $T_\alpha$.*

*Proof.*

$$L(\theta_1,\theta_2) = L(\alpha\theta_1, \alpha^{-1}\theta_2),$$

we have then by differentiation

$$(\nabla L)(\theta_1,\theta_2) = (\nabla L)(\alpha\theta_1,\alpha^{-1}\theta_2)\begin{bmatrix}\alpha\mathbb{I}_{n_1} & 0\\ 0 & \alpha^{-1}\mathbb{I}_{n_2}\end{bmatrix}$$
$$\Leftrightarrow (\nabla L)(\alpha\theta_1,\alpha^{-1}\theta_2) = (\nabla L)(\theta_1,\theta_2)\begin{bmatrix}\alpha^{-1}\mathbb{I}_{n_1} & 0\\ 0 & \alpha\mathbb{I}_{n_2}\end{bmatrix}$$

and

$$(\nabla^2 L)(\alpha\theta_1,\alpha^{-1}\theta_2)$$
$$= \begin{bmatrix}\alpha^{-1}\mathbb{I}_{n_1} & 0\\ 0 & \alpha\mathbb{I}_{n_2}\end{bmatrix}(\nabla^2 L)(\theta_1,\theta_2)\begin{bmatrix}\alpha^{-1}\mathbb{I}_{n_1} & 0\\ 0 & \alpha\mathbb{I}_{n_2}\end{bmatrix}.$$

$\square$

**Sharpest direction** Through these transformations we can easily find, for any critical point which is a minimum with non-zero Hessian, an observationally equivalent parameter whose Hessian has an arbitrarily large spectral norm.

**Theorem 4.** *For a one-hidden layer rectified neural network of the form*

$$y = \phi_{rect}(x\cdot\theta_1)\cdot\theta_2,$$

*and critical point $\theta = (\theta_1,\theta_2)$ being a minimum for $L$, such that $(\nabla^2 L)(\theta) \neq 0$, $\forall M > 0, \exists\alpha > 0, \left|\!\left|\!\left|(\nabla^2 L)\big(T_\alpha(\theta)\big)\right|\!\right|\!\right|_2 \geq M$ where $\left|\!\left|\!\left|(\nabla^2 L)\big(T_\alpha(\theta)\big)\right|\!\right|\!\right|_2$ is the spectral norm of $(\nabla^2 L)\big(T_\alpha(\theta)\big)$.*

*Proof.* The trace of a symmetric matrix is the sum of its eigenvalues and a real symmetric matrix can be diagonalized in $\mathbb{R}$, therefore if the Hessian is non-zero, there is one non-zero positive diagonal element. Without loss of generality, we will assume that this non-zero element of value $\gamma > 0$ corresponds to an element in $\theta_1$. Therefore the Frobenius norm $\left|\!\left|\!\left|(\nabla^2 L)\big(T_\alpha(\theta)\big)\right|\!\right|\!\right|_F$ of

$$(\nabla^2 L)(\alpha\theta_1,\alpha^{-1}\theta_2)$$
$$= \begin{bmatrix}\alpha^{-1}\mathbb{I}_{n_1} & 0\\ 0 & \alpha\mathbb{I}_{n_2}\end{bmatrix}(\nabla^2 L)(\theta_1,\theta_2)\begin{bmatrix}\alpha^{-1}\mathbb{I}_{n_1} & 0\\ 0 & \alpha\mathbb{I}_{n_2}\end{bmatrix}.$$

is lower bounded by $\alpha^{-2}\gamma$.

Since all norms are equivalent in finite dimension, there exists a constant $r > 0$ such that $r\|\|A\|\|_F \leq \|\|A\|\|_2$ for all symmetric matrices $A$. So by picking $\alpha < \sqrt{\frac{r\gamma}{M}}$, we are guaranteed that $\big\|\big\|(\nabla^2 L)\big(T_\alpha(\theta)\big)\big\|\big\|_2 \geq M$. $\square$

Any minimum with non-zero Hessian will be observationally equivalent to a minimum whose Hessian has an arbitrarily large spectral norm. Therefore for any minimum in the loss function, if there exists another minimum that generalizes better then there exists another minimum that generalizes better and is also sharper according the spectral norm of the Hessian. The spectral norm of critical points' Hessian becomes as a result less relevant as a measure of potential generalization error. Moreover, since the spectral norm lower bounds the trace for a positive semi-definite symmetric matrix, the same conclusion can be drawn for the trace.

Further properties of the Hessian are analyzed in Appendix E.

### 4.3 $\epsilon$-sharpness

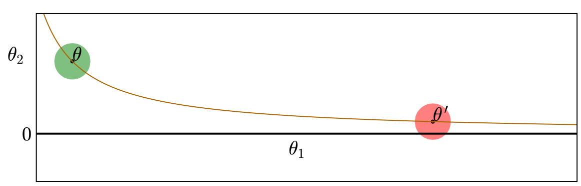



Figure 4: An illustration of how we exploit non-identifiability and its particular geometry to obtain sharper minima: although $\theta$ is far from the $\theta_2 = 0$ line, the observationally equivalent parameter $\theta'$ is closer. The green and red circle centered on each of these points have the same radius. Best seen with colors.

We have redefined for $\epsilon > 0$ the $\epsilon$-sharpness of Keskar et al. (2017) as follow

$$\frac{\max_{\theta' \in B_2(\epsilon, \theta)} \big(L(\theta') - L(\theta)\big)}{1 + L(\theta)}$$

where $B_2(\epsilon, \theta)$ is the Euclidean ball of radius $\epsilon$ centered on $\theta$. This modification will demonstrate more clearly the issues of that metric as a measure of probable generalization. If we use $K = 2$ and $(\theta_1, \theta_2)$ corresponding to a non-constant function, i.e. $\theta_1 \neq 0$ and $\theta_2 \neq 0$, then we can define $\alpha = \frac{\epsilon}{\|\theta_1\|_2}$. We will now consider the observationally equivalent parameter $T_\alpha(\theta_1, \theta_2) = (\epsilon\frac{\theta_1}{\|\theta_1\|_2}, \alpha^{-1}\theta_2)$. Given that $\|\theta_1\|_2 \leq \|\theta\|_2$, we have that $(0, \alpha^{-1}\theta_2) \in B_2\big(\epsilon, T_\alpha(\theta)\big)$, making the maximum loss in this neighborhood at least as high as the best constant-valued function, incurring relatively high sharpness. Figure 4 provides a visualization of the proof.

For rectified neural network every minimum is observationally equivalent to a minimum that generalizes as well but with high $\epsilon$-sharpness. This also applies when using the *full-space* $\epsilon$-sharpness used by Keskar et al. (2017). We can prove this similarly using the equivalence of norms in finite dimensional vector spaces and the fact that for $c > 0, \epsilon > 0, \epsilon \leq \epsilon(c+1)$ (see Keskar et al. (2017)). We have not been able to show a similar problem with *random subspace* $\epsilon$-sharpness used by Keskar et al. (2017), i.e. a restriction of the maximization to a random subspace, which could relate to the notion of *wide valleys* described by Chaudhari et al. (2017).

By exploiting the non-Euclidean geometry and non-identifiability of rectified neural networks, we were able to demonstrate some of the limits of using typical definitions of minimum's flatness as core explanation for generalization.

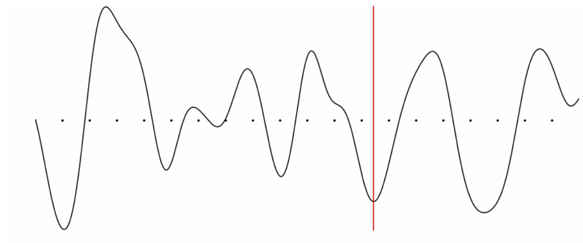
## 5 Allowing reparametrizations

In the previous section 4 we explored the case of a fixed parametrization, that of deep rectifier models. In this section we demonstrate a simple observation. If we are allowed to change the parametrization of some function $f$, we can obtain arbitrarily different geometries without affecting how the function evaluates on unseen data. The same holds for reparametrization of the input space. The implication is that the correlation between the geometry of the parameter space (and hence the error surface) and the behavior of a given function is meaningless if not preconditioned on the specific parametrization of the model.

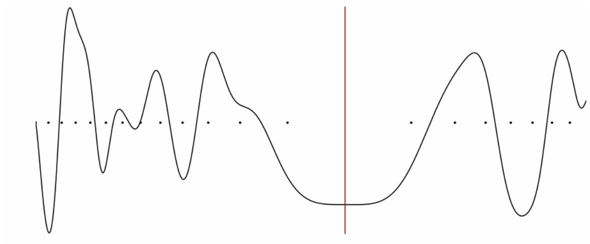
### 5.1 Model reparametrization

One thing that needs to be considered when relating flatness of minima to their probable generalization is that the choice of parametrization and its associated geometry are arbitrary. Since we are interested in finding a prediction function in a given family of functions, no reparametrization of this family should influence generalization of any of these functions. Given a bijection $g$ onto $\theta$, we can define new transformed parameter $\eta = g^{-1}(\theta)$. Since $\theta$ and $\eta$ represent in different space the same prediction function, they should generalize as well.

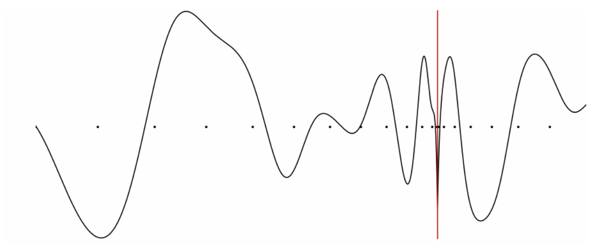
Let's call $L_\eta = L \circ g$ the loss function with respect to the new parameter $\eta$. We generalize the derivation of Subsec-



(a) Loss function with default parametrization



(b) Loss function with reparametrization



(c) Loss function with another reparametrization

Figure 5: A one-dimensional example on how much the geometry of the loss function depends on the parameter space chosen. The $x$-axis is the parameter value and the $y$-axis is the loss. The points correspond to a regular grid in the default parametrization. In the default parametrization, all minima have roughly the same curvature but with a careful choice of reparametrization, it is possible to turn a minimum significantly flatter or sharper than the others. Reparametrizations in this figure are of the form $\eta = (|\theta - \hat{\theta}|^2 + b)^a(\theta - \hat{\theta})$ where $b \geq 0, a > -\frac{1}{2}$ and $\hat{\theta}$ is shown with the red vertical line.

tion 4.2:

$$\begin{aligned} L_\eta(\eta) &= L\big(g(\eta)\big) \\ \Rightarrow (\nabla L_\eta)(\eta) &= (\nabla L)\big(g(\eta)\big)(\nabla g)(\eta) \\ \Rightarrow (\nabla^2 L_\eta)(\eta) &= (\nabla g)(\eta)^T(\nabla^2 L)\big(g(\eta)\big)(\nabla g)(\eta) \\ &\quad + (\nabla L)\big(g(\eta)\big)(\nabla^2 g)(\eta). \end{aligned}$$

At a differentiable critical point, we have by definition $(\nabla L)\big(g(\eta)\big) = 0$, therefore the transformed Hessian at a critical point becomes

$$(\nabla^2 L_\eta)(\eta) = (\nabla g)(\eta)^T(\nabla^2 L)\big(g(\eta)\big)(\nabla g)(\eta).$$

This means that by reparametrizing the problem we can modify to a large extent the geometry of the loss function so as to have sharp minima of $L$ in $\theta$ correspond to flat minima of $L_\eta$ in $\eta = g^{-1}(\theta)$ and conversely. Figure 5 illustrates that point in one dimension. Several practical (Dinh et al., 2014; Rezende & Mohamed, 2015; Kingma et al., 2016; Dinh et al., 2016) and theoretical works (Hyvärinen & Pajunen, 1999) show how powerful bijections can be. We can also note that the formula for the transformed Hessian at a critical point also applies if $g$ is not invertible, $g$ would just need to be surjective over $\Theta$ in order to cover exactly the same family of prediction functions

$$\{f_\theta, \theta \in \Theta\} = \{f_{g(\eta)}, \eta \in g^{-1}(\Theta)\}.$$

We show in Appendix A, bijections that allow us to perturb the relative flatness between a finite number of minima.

Instances of commonly used reparametrization are *batch normalization* (Ioffe & Szegedy, 2015), or the *virtual batch normalization* variant (Salimans et al., 2016), and *weight normalization* (Badrinarayanan et al., 2015; Salimans & Kingma, 2016; Arpit et al., 2016). Im et al. (2016) have plotted how the loss function landscape was affected by batch normalization. However, we will focus on weight normalization reparametrization as the analysis will be simpler, but the intuition with batch normalization will be similar. Weight normalization reparametrizes a nonzero weight $w$ as $w = s\frac{v}{\|v\|_2}$ with the new parameter being the scale $s$ and the unnormalized weight $v \neq 0$.

Since we can observe that $w$ is invariant to scaling of $v$, reasoning similar to Section 3 can be applied with the simpler transformations $T'_\alpha : v \mapsto \alpha v$ for $\alpha \neq 0$. Moreover, since this transformation is a simpler isotropic scaling, the conclusion that we can draw can be actually more powerful with respect to $v$:

- every minimum has infinite volume $\epsilon$-sharpness;
- every minimum is observationally equivalent to an infinitely sharp minimum and to an infinitely flat minimum when considering nonzero eigenvalues of the Hessian;
- every minimum is observationally equivalent to a minimum with arbitrarily low full-space and random subspace $\epsilon$-sharpness and a minimum with high full-space $\epsilon$-sharpness.

This further weakens the link between the flatness of a minimum and the generalization property of the associated prediction function when a specific parameter space has not been specified and explained beforehand.

## 5.2 Input representation

As we conclude that the notion of flatness for a minimum in the loss function by itself is not sufficient to determine its generalization ability in the general case, we can choose to focus instead on properties of the prediction function instead. Motivated by some work in *adversarial examples* (Szegedy et al., 2014; Goodfellow et al., 2015) for deep neural networks, one could decide on its generalization property by analyzing the gradient of the prediction function on examples. Intuitively, if the gradient is small on typical points from the distribution or has a small Lipschitz constant, then a small change in the input should not incur a large change in the prediction.

But this infinitesimal reasoning is once again very dependent of the local geometry of the input space. For an invertible preprocessing $\xi^{-1}$, e.g. *feature standardization*, *whitening* or *gaussianization* (Chen & Gopinath, 2001), we will call $f_\xi = f \circ \xi$ the prediction function on the preprocessed input $u = \xi^{-1}(x)$. We can reproduce the derivation in Section 5 to obtain

$$\frac{\partial f_\xi}{\partial u^T}\big(\xi(u)\big) = \frac{\partial f}{\partial x^T}\big(\xi(u)\big)\frac{\partial \xi}{\partial u^T}(u).$$

As we can alter significantly the relative magnitude of the gradient at each point, analyzing the amplitude of the gradient of the prediction function might prove problematic if the choice of the input space have not been explained beforehand. This remark applies in applications involving images, sound or other signals with invariances (Larsen et al., 2015). For example, Theis et al. (2016) show for images how a small drift of one to four pixels can incur a large difference in terms of $\mathcal{L}_2$ norm.

# 6 Discussion

It has been observed empirically that minima found by standard deep learning algorithms that generalize well tend to be flatter than found minima that did not generalize well (Chaudhari et al., 2017; Keskar et al., 2017). However, when following several definitions of flatness, we have shown that the conclusion that flat minima should generalize better than sharp ones cannot be applied as is without further context. Previously used definitions fail to account for the complex geometry of some commonly used deep architectures. In particular, the non-identifiability of the model induced by symmetries, allows one to alter the flatness of a minimum without affecting the function it represents. Additionally the whole geometry of the error surface with respect to the parameters can be changed arbitrarily under different parametrizations. In the spirit of (Swirszcz et al., 2016), our work indicates that more care is needed to define flatness to avoid degeneracies of the geometry of the model under study. Also such a concept can not be divorced from the particular parametrization of the model or input space.

# Acknowledgements

The authors would like to thank Grzegorz Świrszcz for an insightful discussion on the paper, Harm De Vries, Yann Dauphin, Jascha Sohl-Dickstein and Csar Laurent for useful discussions about optimization, Danilo Rezende for explaining universal approximation using normalizing flows and Kyle Kastner, Adriana Romero, Junyoung Chung, Nicolas Ballas, Aaron Courville, George Dahl, Yaroslav Ganin, Prajit Ramachandran, alar Glehre, Ahmed Touati and the ICML reviewers for useful feedback.

# References

Amari, Shun-Ichi. Natural gradient works efficiently in learning. *Neural Comput.*, 10(2), 1998.

Arpit, Devansh, Zhou, Yingbo, Kota, Bhargava U, and Govindaraju, Venu. Normalization propagation: A parametric technique for removing internal covariate shift in deep networks. *arXiv preprint arXiv:1603.01431*, 2016.

Bach, Francis R. and Blei, David M. (eds.). *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, volume 37 of *JMLR Workshop and Conference Proceedings*, 2015. JMLR.org. URL http://jmlr.org/proceedings/papers/v37/.

Badrinarayanan, Vijay, Mishra, Bamdev, and Cipolla, Roberto. Understanding symmetries in deep networks. *arXiv preprint arXiv:1511.01029*, 2015.

Bahdanau, Dzmitry, Cho, Kyunghyun, and Bengio, Yoshua. Neural machine translation by jointly learning to align and translate. In *ICLR'2015, arXiv:1409.0473*, 2015.

Bottou, Léon. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*, pp. 177–186. Springer, 2010.

Bottou, Léon and Bousquet, Olivier. The tradeoffs of large scale learning. In Platt, J.C., Koller, D., Singer, Y., and Roweis, S. (eds.), *Advances in Neural Information Processing Systems*, volume 20, pp. 161–168. NIPS Foundation (http://books.nips.cc), 2008. URL http://leon.bottou.org/papers/bottou-bousquet-2008.

Bottou, Léon and LeCun, Yann. On-line learning for very large datasets. *Applied Stochastic Models in Business and Industry*, 21(2):137–151, 2005. URL http://leon.bottou.org/papers/bottou-lecun-2004a.

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

Bousquet, Olivier and Elisseeff, André. Stability and generalization. *Journal of Machine Learning Research*, 2(Mar):499–526, 2002.

Chan, William, Jaitly, Navdeep, Le, Quoc V., and Vinyals, Oriol. Listen, attend and spell: A neural network for large vocabulary conversational speech recognition. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2016, Shanghai, China, March 20-25, 2016*, pp. 4960–4964. IEEE, 2016. ISBN 978-1-4799-9988-0. doi: 10.1109/ICASSP.2016.7472621. URL http://dx.doi.org/10.1109/ICASSP.2016.7472621.

Chaudhari, Pratik, Choromanska, Anna, Soatto, Stefano, LeCun, Yann, Baldassi, Carlo, Borgs, Christian, Chayes, Jennifer, Sagun, Levent, and Zecchina, Riccardo. Entropy-sgd: Biasing gradient descent into wide valleys. In *ICLR'2017, arXiv:1611.01838*, 2017.

Chen, Scott Saobing and Gopinath, Ramesh A. Gaussianization. In Leen, T. K., Dietterich, T. G., and Tresp, V. (eds.), *Advances in Neural Information Processing Systems 13*, pp. 423–429. MIT Press, 2001. URL http://papers.nips.cc/paper/1856-gaussianization.pdf.

Cho, Kyunghyun, van Merrienboer, Bart, Gülçehre, Çaglar, Bahdanau, Dzmitry, Bougares, Fethi, Schwenk, Holger, and Bengio, Yoshua. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In Moschitti, Alessandro, Pang, Bo, and Daelemans, Walter (eds.), *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pp. 1724–1734. ACL, 2014. ISBN 978-1-937284-96-1. URL http://aclweb.org/anthology/D/D14/D14-1179.pdf.

Choromanska, Anna, Henaff, Mikael, Mathieu, Michaël, Arous, Gérard Ben, and LeCun, Yann. The loss surfaces of multilayer networks. In *AISTATS*, 2015.

Chorowski, Jan K, Bahdanau, Dzmitry, Serdyuk, Dmitriy, Cho, Kyunghyun, and Bengio, Yoshua. Attention-based models for speech recognition. In *Advances in Neural Information Processing Systems*, pp. 577–585, 2015.

Collobert, Ronan, Puhrsch, Christian, and Synnaeve, Gabriel. Wav2letter: an end-to-end convnet-based speech recognition system. *arXiv preprint arXiv:1609.03193*, 2016.

Dauphin, Yann N., Pascanu, Razvan, Gülçehre, Çaglar, Cho, KyungHyun, Ganguli, Surya, and Bengio, Yoshua. Identifying and attacking the saddle point problem in high-dimensional non-convex optimization. *NIPS*, 2014.

Desjardins, Guillaume, Simonyan, Karen, Pascanu, Razvan, and Kavukcuoglu, Koray. Natural neural networks. *NIPS*, 2015.

Dinh, Laurent, Krueger, David, and Bengio, Yoshua. Nice: Non-linear independent components estimation. *arXiv preprint arXiv:1410.8516*, 2014.

Dinh, Laurent, Sohl-Dickstein, Jascha, and Bengio, Samy. Density estimation using real nvp. In *ICLR'2017, arXiv:1605.08803*, 2016.

Duchi, John, Hazan, Elad, and Singer, Yoram. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul):2121–2159, 2011.

Gehring, Jonas, Auli, Michael, Grangier, David, and Dauphin, Yann N. A convolutional encoder model for neural machine translation. *arXiv preprint arXiv:1611.02344*, 2016.

Glorot, Xavier, Bordes, Antoine, and Bengio, Yoshua. Deep sparse rectifier neural networks. In *Aistats*, volume 15, pp. 275, 2011.

Gonen, Alon and Shalev-Shwartz, Shai. Fast rates for empirical risk minimization of strict saddle problems. *arXiv preprint arXiv:1701.04271*, 2017.

Goodfellow, Ian J, Warde-Farley, David, Mirza, Mehdi, Courville, Aaron C, and Bengio, Yoshua. Maxout networks. *ICML (3)*, 28: 1319–1327, 2013.

Goodfellow, Ian J, Shlens, Jonathon, and Szegedy, Christian. Explaining and harnessing adversarial examples. In *ICLR'2015 arXiv:1412.6572*, 2015.

Graves, Alex, Mohamed, Abdel-rahman, and Hinton, Geoffrey. Speech recognition with deep recurrent neural networks. In *Acoustics, speech and signal processing (icassp), 2013 ieee international conference on*, pp. 6645–6649. IEEE, 2013.

Hannun, Awni Y., Case, Carl, Casper, Jared, Catanzaro, Bryan, Diamos, Greg, Elsen, Erich, Prenger, Ryan, Satheesh, Sanjeev, Sengupta, Shubho, Coates, Adam, and Ng, Andrew Y. Deep speech: Scaling up end-to-end speech recognition. *CoRR*, abs/1412.5567, 2014. URL http://arxiv.org/abs/1412.5567.

Hardt, Moritz, Recht, Ben, and Singer, Yoram. Train faster, generalize better: Stability of stochastic gradient descent. In Balcan, Maria-Florina and Weinberger, Kilian Q. (eds.), *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, pp. 1225–1234. JMLR.org, 2016. URL http://jmlr.org/proceedings/papers/v48/hardt16.html.

He, Kaiming, Zhang, Xiangyu, Ren, Shaoqing, and Sun, Jian. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pp. 1026–1034, 2015.

He, Kaiming, Zhang, Xiangyu, Ren, Shaoqing, and Sun, Jian. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.

Hinton, Geoffrey E and Van Camp, Drew. Keeping the neural networks simple by minimizing the description length of the weights. In *Proceedings of the sixth annual conference on Computational learning theory*, pp. 5–13. ACM, 1993.

Hochreiter, Sepp and Schmidhuber, Jürgen. Flat minima. *Neural Computation*, 9(1):1–42, 1997.

Hyvärinen, Aapo and Pajunen, Petteri. Nonlinear independent component analysis: Existence and uniqueness results. *Neural Networks*, 12(3):429–439, 1999.

Im, Daniel Jiwoong, Tao, Michael, and Branson, Kristin. An empirical analysis of deep network loss surfaces. *arXiv preprint arXiv:1612.04010*, 2016.

Ioffe, Sergey and Szegedy, Christian. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Bach & Blei (2015), pp. 448–456. URL http://jmlr.org/proceedings/papers/v37/ioffe15.html.

Jarrett, Kevin, Kavukcuoglu, Koray, LeCun, Yann, et al. What is the best multi-stage architecture for object recognition? In *Computer Vision, 2009 IEEE 12th International Conference on*, pp. 2146–2153. IEEE, 2009.

Keskar, Nitish Shirish, Mudigere, Dheevatsa, Nocedal, Jorge, Smelyanskiy, Mikhail, and Tang, Ping Tak Peter. On large-batch training for deep learning: Generalization gap and sharp minima. In *ICLR'2017, arXiv:1609.04836*, 2017.

Kingma, Diederik P, Salimans, Tim, Jozefowicz, Rafal, Chen, Xi, Sutskever, Ilya, and Welling, Max. Improved variational inference with inverse autoregressive flow. In Lee, D. D., Sugiyama, M., Luxburg, U. V., Guyon, I., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 29*, pp. 4743–4751. Curran Associates, Inc., 2016.

Klyachko, Alexander A. Random walks on symmetric spaces and inequalities for matrix spectra. *Linear Algebra and its Applications*, 319(1-3):37–59, 2000.

Krizhevsky, Alex, Sutskever, Ilya, and Hinton, Geoffrey E. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pp. 1097–1105, 2012.

Lafond, Jean, Vasilache, Nicolas, and Bottou, Léon. About diagonal rescaling applied to neural nets. *ICML Workshop on Optimization Methods for the Next Generation of Machine Learning*, 2016.

Larsen, Anders Boesen Lindbo, Sønderby, Søren Kaae, and Winther, Ole. Autoencoding beyond pixels using a learned similarity metric. *CoRR*, abs/1512.09300, 2015. URL http://arxiv.org/abs/1512.09300.

Montufar, Guido F, Pascanu, Razvan, Cho, Kyunghyun, and Bengio, Yoshua. On the number of linear regions of deep neural networks. In *Advances in neural information processing systems*, pp. 2924–2932, 2014.

Nair, Vinod and Hinton, Geoffrey E. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pp. 807–814, 2010.

Nesterov, Yurii and Vial, Jean-Philippe. Confidence level solutions for stochastic programming. *Automatica*, 44(6):1559–1568, 2008.

Neyshabur, Behnam, Salakhutdinov, Ruslan R, and Srebro, Nati. Path-sgd: Path-normalized optimization in deep neural networks. In *Advances in Neural Information Processing Systems*, pp. 2422–2430, 2015.

Pascanu, Razvan and Bengio, Yoshua. Revisiting natural gradient for deep networks. *ICLR*, 2014.

Raghu, Maithra, Poole, Ben, Kleinberg, Jon, Ganguli, Surya, and Sohl-Dickstein, Jascha. On the expressive power of deep neural networks. *arXiv preprint arXiv:1606.05336*, 2016.

Rezende, Danilo Jimenez and Mohamed, Shakir. Variational inference with normalizing flows. In Bach & Blei (2015), pp. 1530–1538. URL http://jmlr.org/proceedings/papers/v37/rezende15.html.

Sagun, Levent, Güney, V Ugur, Arous, Gerard Ben, and LeCun, Yann. Explorations on high dimensional landscapes. *arXiv preprint arXiv:1412.6615*, 2014.

Sagun, Levent, Bottou, Léon, and LeCun, Yann. Singularity of the hessian in deep learning. *arXiv preprint arXiv:1611.07476*, 2016.

Salimans, Tim and Kingma, Diederik P. Weight normalization: A simple reparameterization to accelerate training of deep neural networks. In *Advances in Neural Information Processing Systems*, pp. 901–901, 2016.

Salimans, Tim, Goodfellow, Ian, Zaremba, Wojciech, Cheung, Vicki, Radford, Alec, and Chen, Xi. Improved techniques for training gans. In *Advances in Neural Information Processing Systems*, pp. 2226–2234, 2016.

Saxe, Andrew M., McClelland, James L., and Ganguli, Surya. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *CoRR*, abs/1312.6120, 2013. URL http://arxiv.org/abs/1312.6120.

Simonyan, Karen and Zisserman, Andrew. Very deep convolutional networks for large-scale image recognition. In *ICLR'2015, arXiv:1409.1556*, 2015.

Sutskever, Ilya, Vinyals, Oriol, and Le, Quoc V. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pp. 3104–3112, 2014.

Swirszcz, Grzegorz, Czarnecki, Wojciech Marian, and Pascanu, Razvan. Local minima in training of deep networks. *CoRR*, abs/1611.06310, 2016.

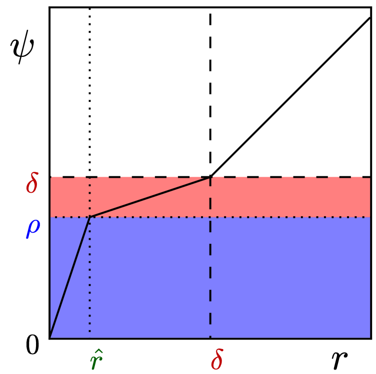
Szegedy, Christian, Zaremba, Wojciech, Sutskever, Ilya, Bruna, Joan, Erhan, Dumitru, Goodfellow, Ian, and Fergus, Rob. Intriguing properties of neural networks. In *ICLR'2014, arXiv:1312.6199*, 2014.

Szegedy, Christian, Liu, Wei, Jia, Yangqing, Sermanet, Pierre, Reed, Scott, Anguelov, Dragomir, Erhan, Dumitru, Vanhoucke, Vincent, and Rabinovich, Andrew. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–9, 2015.

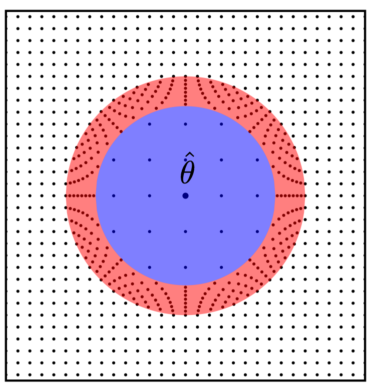
Theis, Lucas, Oord, Aäron van den, and Bethge, Matthias. A note on the evaluation of generative models. In *ICLR'2016 arXiv:1511.01844*, 2016.

Wu, Yonghui, Schuster, Mike, Chen, Zhifeng, Le, Quoc V, Norouzi, Mohammad, Macherey, Wolfgang, Krikun, Maxim, Cao, Yuan, Gao, Qin, Macherey, Klaus, et al. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*, 2016.

Zhang, Chiyuan, Bengio, Samy, Hardt, Moritz, Recht, Benjamin, and Vinyals, Oriol. Understanding deep learning requires rethinking generalization. In *ICLR'2017, arXiv:1611.03530*, 2017.




(a) $\psi(r, \hat{r}, \delta, \rho)$




(b) $g^{-1}(\theta)$

Figure 6: An example of a radial transformation on a 2-dimensional space. We can see that only the area in blue and red, i.e. inside $B_2(\hat{\theta}, \delta)$, are affected. Best seen with colors.

# A Radial transformations

We show an elementary transformation to locally perturb the geometry of a finite-dimensional vector space and therefore affect the relative flatness between a finite number minima, at least in terms of spectral norm of the Hessian. We define the function:

$$\begin{aligned}
&\forall \delta > 0, \forall \rho \in ]0, \delta[, \forall (r, \hat{r}) \in \mathbb{R}_+ \times ]0, \delta[, \\
&\quad \psi(r, \hat{r}, \delta, \rho) = \mathbb{1}\big(r \notin [0, \delta]\big)\, r + \mathbb{1}\big(r \in [0, \hat{r}]\big)\, \rho\, \frac{r}{\hat{r}} \\
&\qquad\qquad + \mathbb{1}\big(r \in ]\hat{r}, \delta]\big) \left( (\rho - \delta)\, \frac{r - \delta}{\hat{r} - \delta} + \delta \right) \\
&\quad \psi'(r, \hat{r}, \delta, \rho) = \mathbb{1}\big(r \notin [0, \delta]\big) + \mathbb{1}\big(r \in [0, \hat{r}]\big)\, \frac{\rho}{\hat{r}} \\
&\qquad\qquad + \mathbb{1}\big(r \in ]\hat{r}, \delta]\big)\, \frac{\rho - \delta}{\hat{r} - \delta}
\end{aligned}$$

For a parameter $\hat{\theta} \in \Theta$ and $\delta > 0, \rho \in ]0, \delta[, \hat{r} \in ]0, \delta[$, inspired by the *radial flows* (Rezende & Mohamed, 2015) in we can define the *radial transformations*

$$\forall \theta \in \Theta,\ g^{-1}(\theta) = \frac{\psi\Big(\|\theta - \hat{\theta}\|_2, \hat{r}, \delta, \rho\Big)}{\|\theta - \hat{\theta}\|_2} \left(\theta - \hat{\theta}\right) + \hat{\theta}$$

with Jacobian

$$\begin{aligned}
\forall \theta \in \Theta,\ (\nabla g^{-1})(\theta) = &\ \psi'(r, \hat{r}, \delta, \rho)\, \mathbb{I}_n \\
&- \mathbb{1}\big(r \in ]\hat{r}, \delta]\big)\, \frac{\delta(\hat{r} - \rho)}{r^3(\hat{r} - \delta)}\, (\theta - \hat{\theta})^T (\theta - \hat{\theta}) \\
&+ \mathbb{1}\big(r \in ]\hat{r}, \delta]\big)\, \frac{\delta(\hat{r} - \rho)}{r(\hat{r} - \delta)}\, \mathbb{I}_n,
\end{aligned}$$

with $r = \|\theta - \hat{\theta}\|_2$.

First, we can observe in Figure 6 that these transformations are purely local: they only have an effect inside the ball $B_2(\hat{\theta}, \delta)$. Through these transformations, you can arbitrarily perturb the ranking between several minima in terms of flatness as described in Subsection 5.1.

# B Considering the bias parameter

When we consider the bias parameter for a one (hidden) layer neural network, the non-negative homogeneity property translates into

$$\begin{aligned}
y &= \phi_{rect}(x \cdot \theta_1 + b_1) \cdot \theta_2 + b_2 \\
&= \phi_{rect}(x \cdot \alpha\theta_1 + \alpha b_1) \cdot \alpha^{-1}\theta_2 + b_2,
\end{aligned}$$

which results in conclusions similar to section 4.

For a deeper rectified neural network, this property results in

$$\begin{aligned}
y = &\ \phi_{rect}\Big(\phi_{rect}\big(\cdots \phi_{rect}(x \cdot \theta_1 + b_1) \cdots\big) \cdot \theta_{K-1} + b_{K-1}\Big) \\
&\cdot \theta_K + b_K \\
= &\ \phi_{rect}\Big(\phi_{rect}\big(\cdots \phi_{rect}(x \cdot \alpha_1\theta_1 + \alpha_1 b_1) \cdots\big) \\
&\cdot \alpha_{K-1}\theta_{K-1} + \prod_{k=1}^{K-1} \alpha_k b_{K-1}\Big) \cdot \alpha_K \theta_K + b_K
\end{aligned}$$

for $\prod_{k=1}^{K} \alpha_k = 1$. This can decrease the amount of eigenvalues of the Hessian that can be arbitrarily influenced.

# C Rectified neural network and Lipschitz continuity

Relative to recent works (Hardt et al., 2016; Gonen & Shalev-Shwartz, 2017) assuming *Lipschitz continuity* of the loss function to derive uniform stability bound, we make the following observation:

**Theorem 5.** *For a one-hidden layer rectified neural network of the form*

$$y = \phi_{rect}(x \cdot \theta_1) \cdot \theta_2,$$

*if $L$ is not constant, then it is not Lipschitz continuous.*

*Proof.* Since a Lipschitz function is necessarily absolutely continuous, we will consider the cases where $L$ is absolutely continuous. First, if $L$ has zero gradient almost everywhere, then $L$ is constant.

Now, if there is a point $\theta$ with non-zero gradient, then by writing

$$\begin{aligned}(\nabla L)(\theta_1, \theta_2) = [&(\nabla_{\theta_1} L)(\theta_1, \theta_2)\\ &(\nabla_{\theta_2} L)(\theta_1, \theta_2)],\end{aligned}$$

we have

$$\begin{aligned}(\nabla L)(\alpha\theta_1, \alpha^{-1}\theta_2) = [&\alpha^{-1}(\nabla_{\theta_1} L)(\theta_1, \theta_2)\\ &\alpha(\nabla_{\theta_2} L)(\theta_1, \theta_2)].\end{aligned}$$

Without loss of generality, we consider $(\nabla_{\theta_1} L)(\theta_1, \theta_2) \neq 0$. Then the limit of the norm

$$\begin{aligned}\|(\nabla L)(\alpha\theta_1, \alpha^{-1}\theta_2)\|_2^2 = \; &\alpha^{-2}\|(\nabla_{\theta_1} L)(\theta_1, \theta_2)\|_2^2\\ &+ \alpha^2\|(\nabla_{\theta_2} L)(\theta_1, \theta_2)\|_2^2\end{aligned}$$

of the gradient goes to $+\infty$ as $\alpha$ goes to 0. Therefore, $L$ is not Lipschitz continuous. □

This result can be generalized to several other models containing a one-hidden layer rectified neural network, including deeper rectified networks.

# D Euclidean distance and input representation

A natural consequence of Subsection 5.2 is that metrics relying on Euclidean metric like *mean square error* or *Earth-mover distance* will rank very differently models depending on the input representation chosen. Therefore, the choice of input representation is critical when ranking different models based on these metrics. Indeed, bijective transformations as simple as *feature standardization* or *whitening* can change the metric significantly.

On the contrary, ranking resulting from metrics like *f-divergence* and *log-likelihood* are not perturbed by bijective transformations because of the *change of variables formula*.

# E Eigenspectrum of Hessian

In Section 4.2, we show how to manipulate the spectral radius and trace of the Hessian as a notion of sharpness. In However, some notion of sharpness might take into account the entire eigenspectrum of the Hessian as opposed to its largest eigenvalue, for instance, Chaudhari et al. (2017) describe the notion of *wide valleys*, allowing the presence of very few large eigenvalues. We can generalize the transformations between observationally equivalent parameters to deeper neural networks with $K-1$ hidden layers: for $\alpha_k > 0, T_\alpha : (\theta_k)_{k\leq K} \mapsto (\alpha_k\theta_k)_{k\in K}$ with $\prod_{k=1}^K \alpha_k = 1$. If we define

$$D_\alpha = \begin{bmatrix} \alpha_1^{-1}\mathbb{I}_{n_1} & 0 & \cdots & 0\\ 0 & \alpha_2^{-1}\mathbb{I}_{n_2} & \cdots & 0\\ \vdots & \vdots & \ddots & \vdots\\ 0 & 0 & \cdots & \alpha_K^{-1}\mathbb{I}_{n_K}\end{bmatrix}$$

then the first and second derivatives at $T_\alpha(\theta)$ will be

$$\begin{aligned}(\nabla L)\big(T_\alpha(\theta)\big) =&(\nabla L)(\theta)D_\alpha\\ (\nabla^2 L)\big(T_\alpha(\theta)\big) =&D_\alpha(\nabla^2 L)(\theta)D_\alpha.\end{aligned}$$

We will show to which extent you can increase several eigenvalues of $(\nabla^2 L)\big(T_\alpha(\theta)\big)$ by varying $\alpha$.

**Definition 6.** *For each $n\times n$ matrix $A$, we define the vector $\lambda(A)$ of sorted singular values of $A$ with their multiplicity $\lambda_1(A) \geq \lambda_2(A) \geq \cdots \geq \lambda_n(A)$.*

If $A$ is symmetric positive semi-definite, $\lambda(A)$ is also the vector of its sorted eigenvalues.

**Theorem 6.** *For a $(K-1)$-hidden layer rectified neural network of the form*

$$y = \phi_{rect}(\phi_{rect}(\cdots\phi_{rect}(x\cdot\theta_1)\cdots)\cdot\theta_{K-1})\cdot\theta_K,$$

*and critical point $\theta = (\theta_k)_{k\leq K}$ being a minimum for $L$, such that $(\nabla^2 L)(\theta)$ has rank $r = rank\big((\nabla^2 L)(\theta)\big)$, $\forall M > 0, \exists\alpha > 0$ such that $\Big(r - \min_{k\leq K}(n_k)\Big)$ eigenvalues are greater than $M$.*

*Proof.* For simplicity, we will note $\sqrt{M}$ the *principal square root* of a symmetric positive-semidefinite matrix $M$. The eigenvalues of $\sqrt{M}$ are the square root of the eigenvalues of $M$ and are its *singular values*. By definition, the *singular values* of $\sqrt{(\nabla^2 L)(\theta)}D_\alpha$ are the square root of the eigenvalues of $D_\alpha(\nabla^2 L)(\theta)D_\alpha$. Without loss of generality, we consider $\min_{k\leq K}(n_k) = n_K$ and choose $\forall k < K, \alpha_k = \beta^{-1}$ and $\alpha_K = \beta^{K-1}$. Since $D_\alpha$ and $\sqrt{(\nabla^2 L)(\theta)}$ are positive symmetric semi-definite matrices, we can apply the multiplicative Horn inequalities (Klyachko, 2000) on singular values of the product $\sqrt{(\nabla^2 L)(\theta)}D_\alpha$:

$$\begin{aligned}&\forall i \leq n, j \leq (n - n_K),\\ &\qquad\lambda_{i+j-n}\big((\nabla^2 L)(\theta)D_\alpha^2\big) \geq \lambda_i\big((\nabla^2 L)(\theta)\big)\beta^2.\end{aligned}$$

By choosing $\beta > \sqrt{\frac{M}{\lambda_r\big((\nabla^2 L)(\theta)\big)}}$, since we have $\forall i \leq r, \lambda_i\big((\nabla^2 L)(\theta)\big) \geq \lambda_r\big((\nabla^2 L)(\theta)\big) > 0$ we can conclude that

$$\begin{aligned}
&\forall i \leq (r - n_K),\\
&\lambda_i\big((\nabla^2 L)(\theta) D_\alpha^2\big) \geq \lambda_{i+n_k}\big((\nabla^2 L)(\theta)\big)\beta^2\\
&\qquad\qquad\qquad\qquad \geq \lambda_r\big((\nabla^2 L)(\theta)\big)\beta^2 > M.
\end{aligned}$$

□

It means that there exists an observationally equivalent parameter with at least $\Big(r - \min_{k\leq K}(n_k)\Big)$ arbitrarily large eigenvalues. Since Sagun et al. (2016) seems to suggests that rank deficiency in the Hessian is due to over-parametrization of the model, one could conjecture that $\Big(r - \min_{k\leq K}(n_k)\Big)$ can be high for thin and deep neural networks, resulting in a majority of large eigenvalues. Therefore, it would still be possible to obtain an equivalent parameter with large Hessian eigenvalues, i.e. sharp in multiple directions.