BULLETIN OF THE POLISH ACADEMY OF SCIENCES
TECHNICAL SCIENCES, Vol. 66, No. 6, 2018
DOI: 10.24425/bpas.2018.125925

INVITED PAPER

Theory II: Deep learning and optimization

T. POGGIO* and Q. LIAO

Center for Brains, Minds, and Machines, McGovern Institute for Brain Research,
Massachusetts Institute of Technology, Cambridge, MA, 02139

Abstract. The landscape of the empirical risk of overparametrized deep convolutional neural networks (DCNNs) is characterized with a mix of theory and experiments. In part A we show the existence of a large number of global minimizers with zero empirical error (modulo inconsistent equations). The argument which relies on the use of Bezout theorem is rigorous when the RELUs are replaced by a polynomial nonlinearity. We show with simulations that the corresponding polynomial network is indistinguishable from the RELU network. According to Bezout theorem, the global minimizers are degenerate unlike the local minima which in general should be non-degenerate. Further we experimentally analyzed and visualized the landscape of empirical risk of DCNNs on CIFAR-10 dataset. Based on above theoretical and experimental observations, we propose a simple model of the landscape of empirical risk. In part B, we characterize the optimization properties of stochastic gradient descent applied to deep networks. The main claim here consists of theoretical and experimental evidence for the following property of SGD: SGD concentrates in probability – like the classical Langevin equation – on large volume, ”flat” minima, selecting with high probability degenerate minimizers which are typically global minimizers.

Key words: deep learning, convolutional neural networks, loss surface, optimization.

1. Introduction

In Part A of this review we characterize the landscape of the empirical risk, while in in part B we show how stochastic gradient descent (SGD) is able to find with high probability global minima instead of local minima[1].

1.1. Part A. We study the empirical risk from three perspectives:

- Theoretical analysis (Section 3): We study the nonlinear system of equations corresponding: a) to critical points of the gradient of the loss (for the L_2 loss function) and in particular; b) to zero minimizers, associated with interpolating solutions. The usual networks contain RELU nonlinearities. Here we use an ε-approximation of them in the sup norm using a polynomial approximation or the corresponding Legendre expansion. We can then invoke Bezout theorem to conclude that there are a very large number of local and global minima}, and that the global, zero-error minima are highly degenerate, whereas the local non-zero minima are – generically – not degenerate. In the case of classification, zero error implies the existence of large margin.

- Visualizations and experimental explorations (Section 4): The theoretical results above indicate that there are degenerate global minima in the loss surface of DCNN. However, it is unclear how the rest of the landscape looks like. To gain some insights into this question, we visualize the landscape of the entire training process using multidimensional scaling. We also probe the landscape at different locations by perturbation and interpolation experiments.

- A simple model of the landscape (Section 5). A simple model for the landscape of empirical risk, shown in Fig. 1 summarizes our theoretical and experimental results. At least in the case of overparametrized DCNNs, the loss surface might be simply a collection of (high-dimensional) basins, which have the following interesting properties: 1. Every basin reaches a flat global minima. 2. The basins are rugged in such a way that most small perturbations of the weights lead to a slightly different convergence path. 3. Despite being perhaps locally rugged, most basins have a relatively regular overall landscape, in the sense that the average of two model within a basin gives a model whose error is roughly the average of (or even lower than) the errors of the original two models. 4. Interpolation between basins, on the other hand, usually raises the error. 5. There may not be any local minima in a basin – we do not encounter any local minima in CIFAR, even when training with batch gradient descent (without noise).

1.2. Part B. Our main claim in Part B is that SGD finds with high probability global minima, because they are degenerate. Degenerate minima exist in general because of the results of Part A. They are preferred by SGD because they correspond to a large volume of the stationary Boltzman probability distribution.

[1] The material of this review is based on previous publications, in particular in [1–3].

*e-mail: tp@csail.mit.edu

Manuscript submitted 2018-05-05, initially accepted for publication 2018-05-05, published in December 2018.

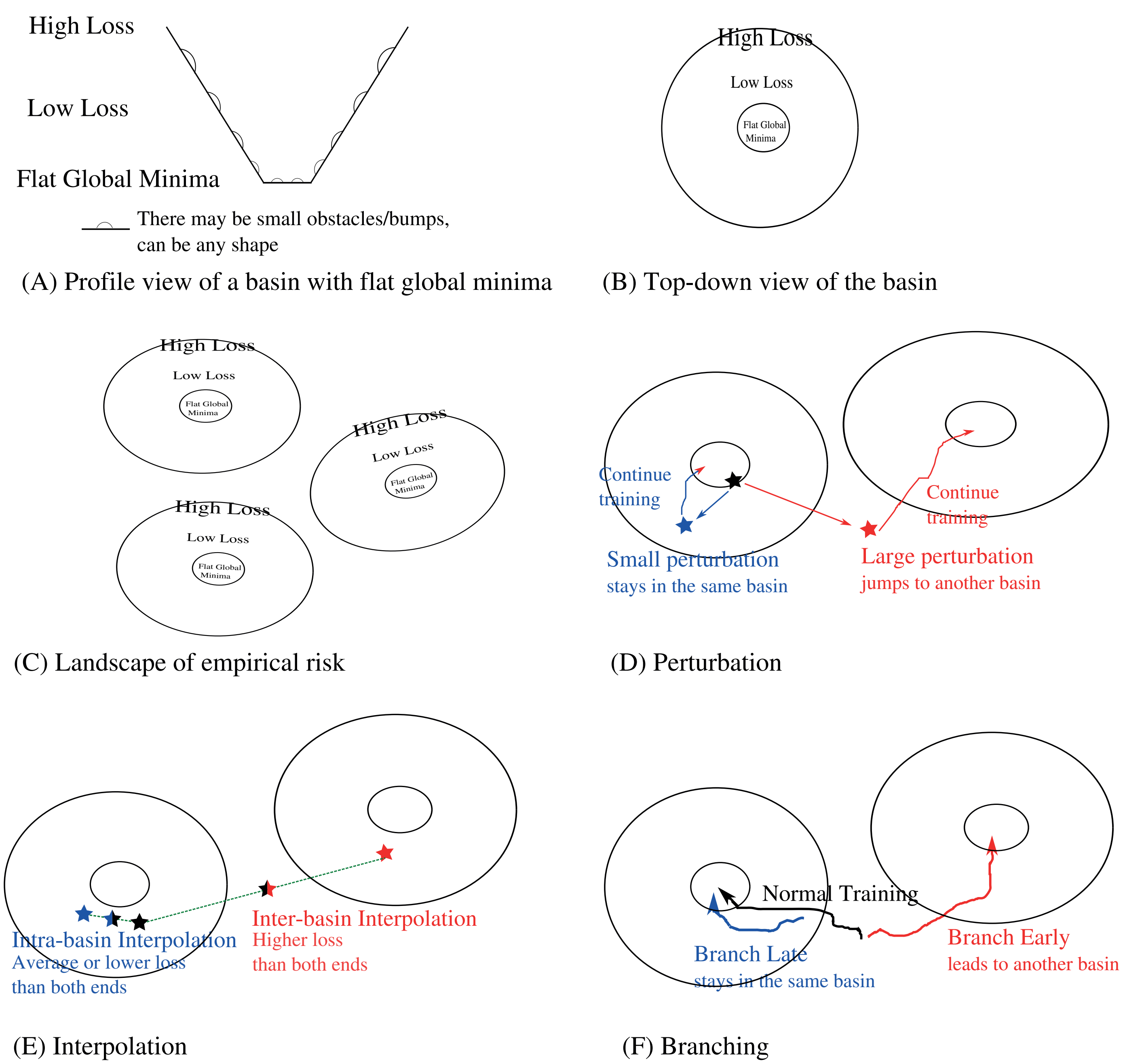


Fig. 1. The landscape of empirical risk of overparametrized DCNN may be simply a collection of (slightly rugged) basins. A) the profile view of a basin. B) the top-down view of a basin. C) an illustration of the landscape of empirical risk. D) perturbation experiment: a small perturbation does not move the model out of its current basin, so re-training converges back to the bottom of the same basin. If the perturbation is large, re-training converges to another basin. E) interpolation experiment: averaging two models within a basin tend to give a error that is the average of the two models (or less). Averaging two models between basins tend to give an error that is higher than both models. (F) branching experiment: one can create a ”branch” of a training trajectory by adding a small noise to model’s weights and continue training. As expected, the earlier the branch is created, the more different the final model becomes. Qualitatively, early branches reach different basins while later branches do not. See Fig. 4 for more details

Part A

2. Framework

We assume a deep polynomial network of the convolutional type with overparametrization, that is more weights than data points, since this is how successful deep networks have been used. Under these conditions, we will show that imposing zero empirical error provides a system of equations (at the zeros) that have a large number of degenerate solutions in the weights. The equations are polynomial in the weights, with coefficients reflecting components of the data vectors (one vector per data point). The system of equations is underdetermined (more unknowns than equations, e.g. data points) because of the assumed overparametrization. Because the global minima are degenerate, that is flat in many of the dimensions, they are more likely to be found by SGD than local minima which are less degenerate.

3. Landscape of the empirical risk: Theoretical analyses

The following theoretical analysis of the landscape of the empirical risk is based on a few assumptions: (1) We assume that the network is overparametrized, typically using several times more parameters (the weights of the network) than data points. In practice, even with data augmentation (in most of the experiments in this paper we do not use data augmentation), it is an empirical observation that it is usually possible to increase the number of parameters making training easier without sacrificing generalization performance in classification and while. (2) This section assumes a regression framework. We study how many solutions in weights lead to perfect prediction of training labels. For simplicity our analysis is focused on the square loss.

Among the critical points of the gradient of the empirical loss, we consider first the zeros of the loss function given by the set of equations $f(x_i) - y_i = 0$ for $i = 1, \cdots, N$, where N is the number of training examples.

The function *f* realized by a deep neural network is polynomial if each of RELU units is replaced by a univariate polynomial. Each RELU can be approximated within any desired ε in the sup norm by a polynomial. In the well-determined case (as many unknown weights as equations, that is data points), Bezout theorem provides an upper bound on the number of solutions. The number of distinct zeros (counting points at infinity, using projective space, assigning an appropriate multiplicity to each intersection point, and excluding degenerate cases) would be equal to Z – the product of the degrees of each of the equations. Since the system of equations is usually underdetermined – as many equations as data points but more unknowns (the weights) – we expect an infinite number of global minima, under the form of *Z* regions of zero empirical error. If the equations are inconsistent, there are still many global minima of the squared error that are solutions of systems of equations with a similar form. The degree of each approximating equation $\ell^d(\varepsilon)$ is determined by the desired accuracy ε for approximating the ReLU activation by a univariate polynomial P of degree $\ell(\varepsilon)$ and by the number of layers d.

The argument based on RELUs approximation for estimating the number of zeros is a qualitative one since good approximation of the $f(x_i)$ does not imply by itself good approximation – via Bezout theorem – of the number of its zeros. Notice, however, that we can abandon the approximation argument and just consider the number of zeros when the RELUs are replaced by a low order univariate polynomial. The argument then would not strictly apply to RELU networks but would still carry weight because the two types of networks – with polynomial activation and with RELUs – behave empirically (see Fig. 2) in a similar way.

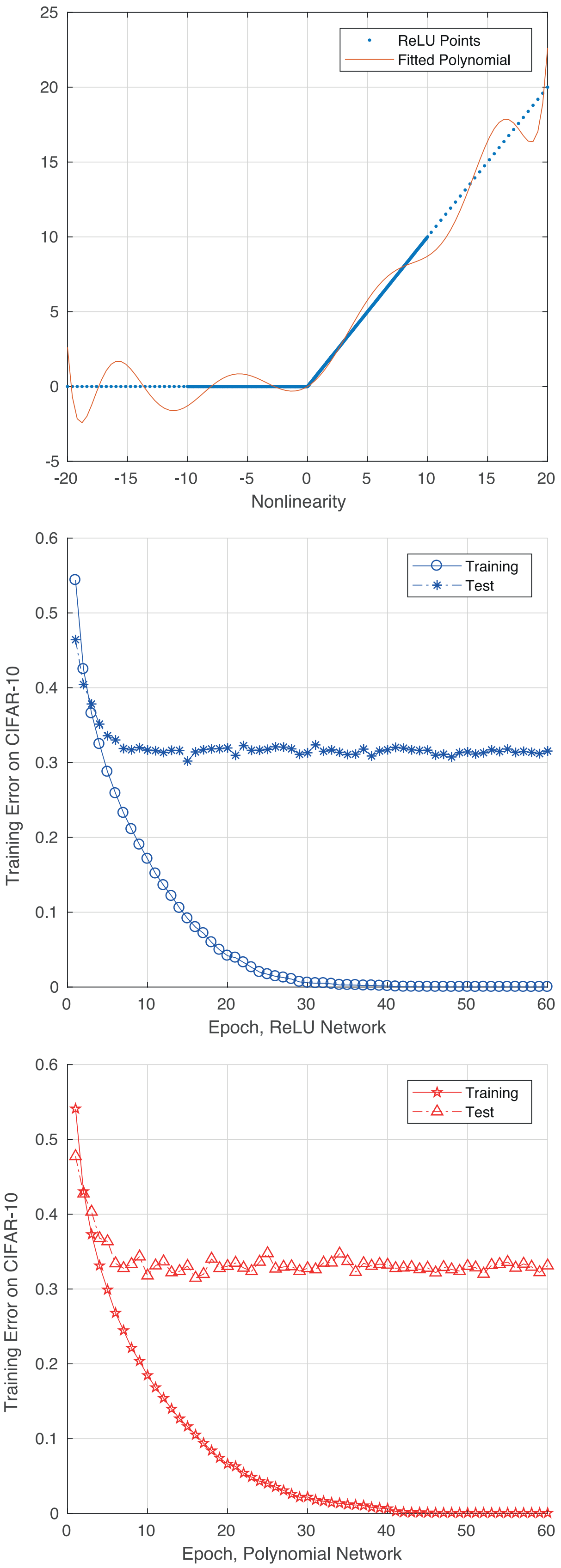


Fig. 2. One can convert a deep network into a polynomial function by using polynomial nonlinearity. As long as the nonlinearity approximates ReLU well (especially near 0), the ”polynomial net” performs similarly to a ReLU net. Our theory applies rigorously to a ”polynomial net”

Even in the non-degenerate case (as many data as parameters), Bezout theorem suggests that there are many solutions. With d layers the degree of the polynomial equations is ℓ^d. With N datapoints the Bezout upper bound in the zeros of the weights is ℓ^{Nd}. Even if the number of real zero – corresponding to zero empirical error – is much smaller $\big($Smale and Shub estimate [4] $l^{Nd}\!/_2\big)$, the number is still enormous: for a CIFAR situation this may be as high as 2^{10^5}.

As mentioned, in several cases we expect absolute zeros to exist with zero empirical error. If the equations are inconsistent it seems likely that global minima with similar properties exist.

It is interesting now to consider the critical points of the gradient. The critical points of the gradient are $\nabla_w \sum_{i=1}^N V\big(f(x_i), y_i\big) = 0$, which gives K equations: $\sum_{i=1}^N \nabla_w V\big(f(x_i), y_i\big) \nabla_w f(x_i) = 0$, where $V(.,.)$ is the loss function.

Approximating within ε in the sup norm each ReLU in $f(x_i)$ with a fixed polynomial $P(z)$ yields a system of K polynomial equations in the weights. They are of course satisfied by the degenerate zeros of the empirical error but also by additional non-degenerate (in the general case) solutions.

Thus, we have **Proposition 1:** There are a very large number of zero-error minima which are highly degenerate unlike the local non-zero minima which are usually not degenerate.

4. The Landscape of the empirical risk: visualizing and analysing the loss surface during the entire training process (on CIFAR-10)

4.1. Experimental Settings. In the empirical work described below we analyze a classification problem with cross entropy loss. Our theoretical analyses with the regression framework provide a lower bound of the number of solutions of the classification problem.

Unless mentioned otherwise, we trained a 6-layer (with the 1st layer being the input) Deep Convolutional Neural Network (DCNN) on CIFAR-10. All the layers are 3×3 convolutional layers with stride 2. No pooling nor shortcut connection is used. Batch Normalizations (BNs) [5] are used between hidden layers. The shifting and scaling parameters in BNs are not used. No data augmentation is performed, so that the training set is fixed (size = 50,000). There are 188,810 parameters in the DCNN.

Multidimensional Scaling. The core of our visualization approach is Multidimensional Scaling (MDS) [6]. We record a large number of intermediate models during the process of several training schemes. Each model is a high dimensional point with the number of dimensions being the number of parameters. The strain-based MDS algorithm is applied to such points and a corresponding set of 2D points are found such that the dissimilarity matrix between the 2D points are as similar to those of the high-dimensional points ascpossible. One minus the cosine distance is used as the dissimilaritycmetric. This is more robust to scaling of the weights, which is usually normalized out by BNs, though the euclidean distance gives qualitatively similar results.

4.2. Visualization of SGD training trajectories. We show in Fig. 3 the optimization trajectories of Stochastic Gradient Descent (SGD) throughout the entire optimization process of training a DCNN on CIFAR-10. The SGD trajectories follow the mini-batch approximations of the training loss surface. Although the trajectories are noisy, the collected points along the trajectories provide a good visualization of the landscape of the empirical risk. We show the visualization based on the weights of layer 2. The results from other layers (e.g., layer 5) are qualitatively similar.

4.3. Visualization of training loss surface with batch gradient descent. Next, we visualize in Fig. 4 the training loss surface by training the models using Batch Gradient Descent (BGD). In addition to training, we also performed perturbation and interpolation experiments as described in Fig. 4. The main trajectory, its branches and the interpolated models together provide another way of visualizing the landscape of the empirical risk.

5. The landscape of the empirical risk: towards an intuitive baseline model

In this section, we propose a simple baseline model for the landscape of empirical risk that is consistent with all of our theoretical and experimental findings. In the case of overparametrized DCNNs, here is a recapitulation of our main observations so far:

- Theoretically, we show that there are a large number of global minimizers with zero empirical error. The same minimizers are degenerate, that is they correspond to multidimensional valleys.
- Regardless of the use of Stochastic Gradient Descent (SGD) or Batch Gradient Descent (BGD), a small perturbation of the model almost always leads to a slightly different convergence path. The earlier the perturbation is in the training process, the more different the final model becomes.
- Interpolating two ”nearby” convergence paths lead to a convergence path with similar errors at every epoch. Interpolating two ”distant” models lead to raised errors.
- We do not observe local minima, even when training with BGD.

A simple model of the landscape of the empirical risk is consistent with the above observations: the landscape is a collection of (hyper) basins, each containing a flat global minima. Illustrations are provided in Fig. 1.

There are of course other variants of this model that can explain our experimental findings. In Fig. 5, we show an alternative model that we call ”basin-fractal”. This model is consistent with most of the above observations. The key difference between the simple basins model and the ”basin-fractal” model is that in the latter case, one should be able to find ”walls” (raised errors) between two models within the same basin. Since it is a fractal, these ”walls” should be present at any level of errors. So far, we only discovered ”walls” between two models when the trajectories leading to them are very different (obtained either by splitting very early in training, as shown in Fig. 4 branch

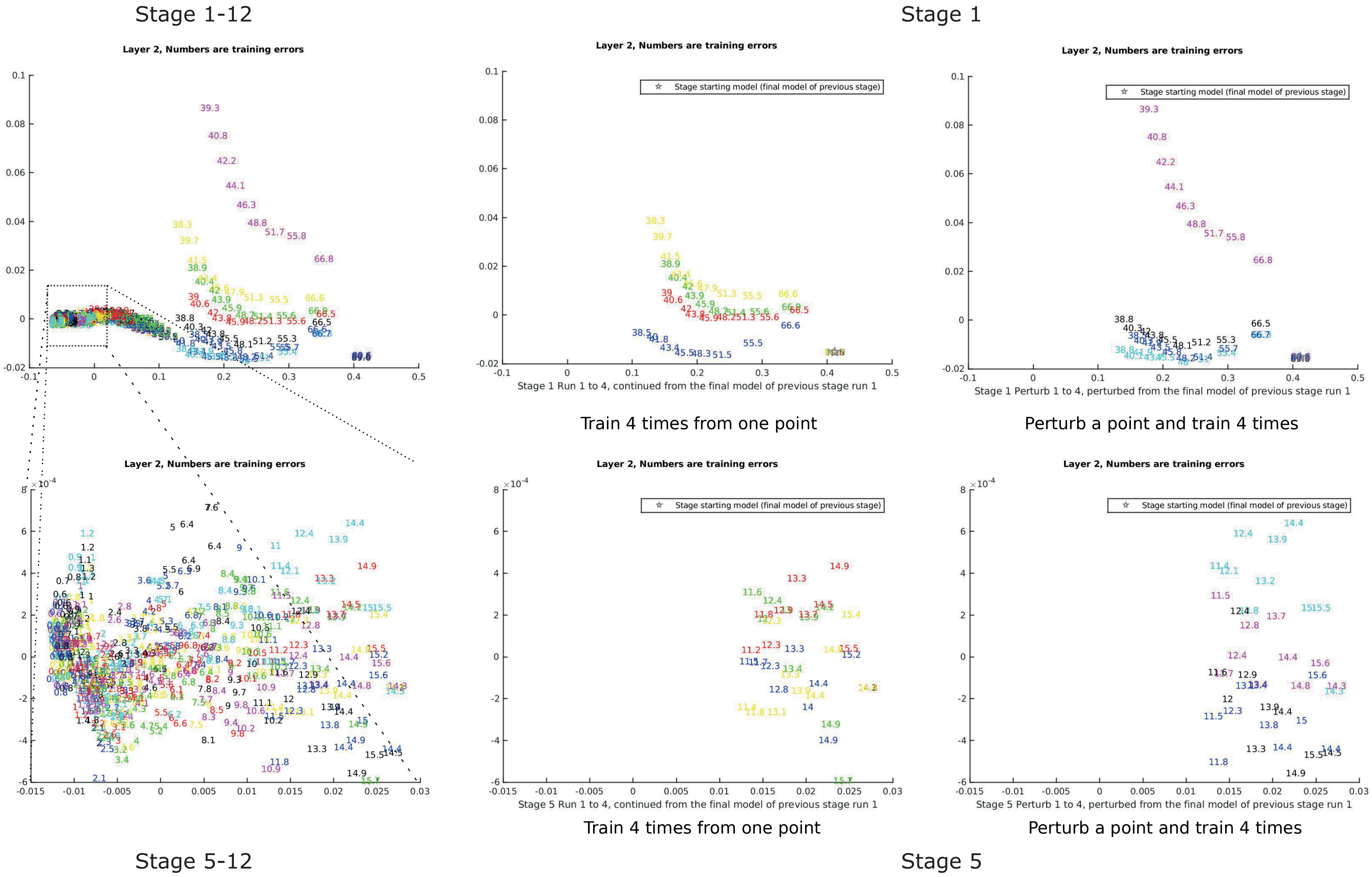


Fig. 3. We train a 6-layer (with the 1st layer being the input) plain convolutional network as described in Section 4.1} on CIFAR-10 with stochastic gradient descent (batch size = 100). We divide the training process into 12 stages. In each stage, we perform *8 parallel* SGDs with learning rate 0.01 for 10 epochs, resulting in 8 parallel trajectories shown in different colors. Trajectories 1 to 4 in each stage start from the final model (denoted by *P*) of trajectory 1 of the previous stage. Trajectories 5 to 8 in each stage start from a perturbed version of *P*. The perturbation is performed by adding gaussian noise to the weights of each layer with the standard deviation being 0.01 times the layer's standard deviation. To visualize the weights of the network at each training stage, we used Multidimensional Scaling (MDS) to project the weights into a 2D space where the pairwise 2D distances roughly reflect the distances in the original high-dimensional space (the absolute units of x and y axes are not important). Each numerical number plotted in above figures indicates the training error of the corresponding model. Every subfigure above shows the MDS visualization of the models at some particular training stages. To avoid clutter, stage 1 and stage 5 training trajectories are shown separately on the right. The results shown here are based on weights from layer 2. The results based on other layers are qualitatively similar. In general, we observe that running again any trajectory with SGD almost always leads to a slightly different convergence path. Also, as shown in the first subfigure, later stages of training make much smaller changes to the weights than early stages – the model seems to converge to a small basin. The "zoomed-in" view of the basin is shown in the second row

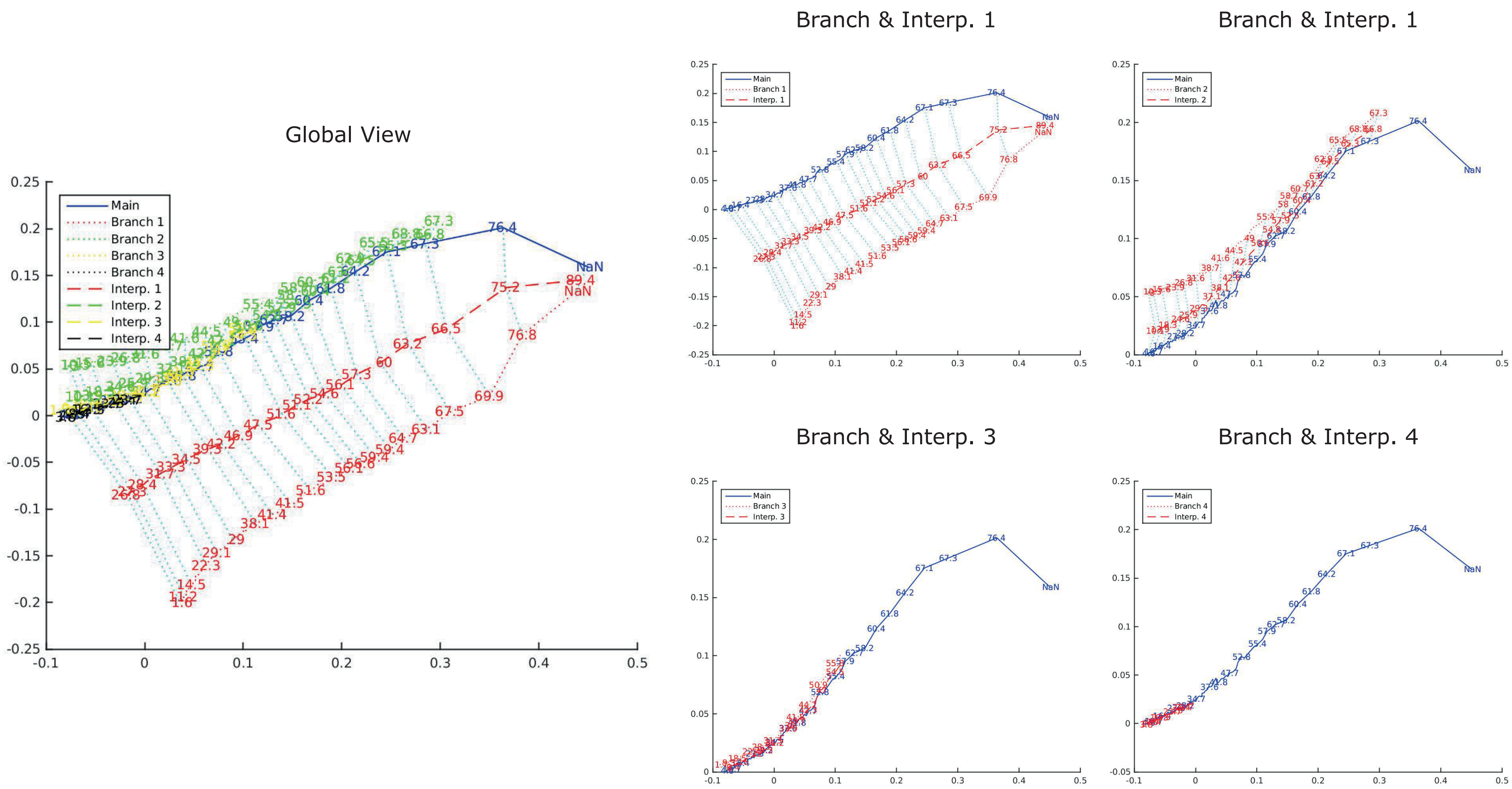


Fig. 4. Visualizing the exact training loss surface using Batch Gradient Descent (BGD). A plain Convolutional Neural Network as described in Section 4.1 is trained on CIFAR-10 from scratch using Batch Gradient Descent (BGD). Again, Multidimensional Scaling (MDS) is performed to project the weights of the models into a 2D space such that the pairwise 2D distances reflect the distances in the original high-dimensional parameter space (the absolute units of x and y axes are not important). Each numerical number in the figure represents a model and its training error. "NaN" corresponds to randomly initialized models (we did not evaluate them and assume they perform at chance). At epoch 0, 10, 50 and 200, we create a branch by perturbing the model by adding a Gaussian noise to all layers. The standard deviation of the Gaussian is 0.25*S, where S denotes the standard deviation of the weights in each layer, respectively. We also interpolate (by averaging) the models between the branches and the main trajectory, epoch by epoch. The interpolated models are evaluated on the entire training set to get a performance. First, surprisingly, BGD does not get stuck in any local minima, indicating that the landscape is "nicer" than expected. The test error of solutions found by BGD is somewhat worse than those found by SGD, but not much worse (BGD $\sim$ 40%, SGD $\sim$ 32%). Another interesting observation is that as training proceeds, the same amount of perturbation is less effective in yielding a drastically different trajectory. Nevertheless, a perturbation almost always leads to a different model. The local neighborhood of the main trajectory seems to be relatively flat and contain many good solutions, supporting our theoretical predictions. It is also intriguing to see interpolated models to have very reasonable performance. The results here are based on weights from layer 2. The results of other layers are similar

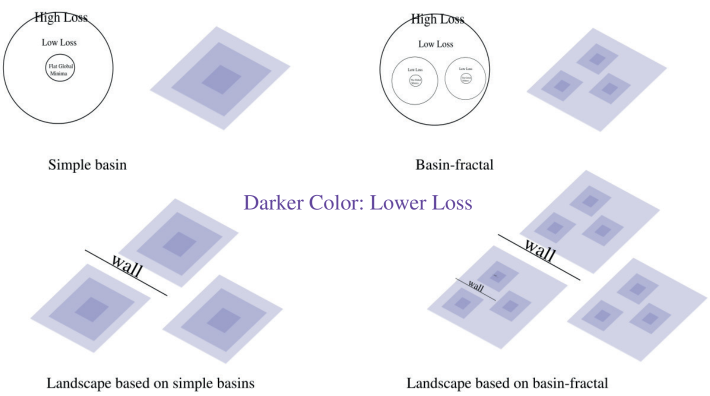


Fig. 5. Discussion: does the loss surface look like simple basins or a basin fractal? The main difference is that whether there are ”walls” (raised errors) within each basin. Experimentally we have not observed such ”walls”

1 or by performing a very large perturbation to the weights). We did not find significant “walls” in all other perturbation and interpolation experiments.

Another surprising finding about the basins is that they do not show any local minima, even when training with batch gradient descent. When the training is long enough with a small enough learning rate, we always achieve zero classification error and negligible cross entropy loss.

Part B

6. SGD: Basic setting

Let Z be a probability space with an unknown measure ρ. A training set S_n is a set of i.i.d. samples z_i, $i = 1, \cdots, n$ from ρ. Assumethat a hypothesis $\mathscr{H}$ is chosen in advance of training. Here we assume $\mathscr{H}$ is a p-dimensional Hilbert space, and identify elements of $\mathscr{H}$ with p-dimensional vectors in $\mathbb{R}^p$. A loss function is a map V: $\mathscr{H} \times Z \to \mathbb{R}_+$. Moreover, we assume the expected loss

$$I(f) = \mathbb{E}_z V(f, z) \tag{1}$$

exists for all $f \in \mathscr{H}$. We consider the problem of finding a minimizer of $I(f)$ in a closed subset $K \subset \mathscr{H}$. We denote this minimizer by f_K so that

$$I(f_K) = \min_{f \in K} I(f). \tag{2}$$

In general, the existence and uniqueness of a minimizer is not guaranteed unless some further assumptions are specified.

Since ρ is unknown, we are not able evaluate $I(f)$. Instead, we try to minimize the empirical loss

$$I_{S_n}(f) = \hat{\mathbb{E}}_{z \sim S_n} V(f, z) = \frac{1}{n} \sum_{i=1}^{n} V(f, z_i) \tag{3}$$

as a proxy. In deep learning, the most commonly used algorithm is SGD and its variants. The basic version of SGD is defined by the following iterations:

$$f_{t+1} = \Pi_K\big(f_t - \gamma_t \nabla V(f_t, z_t)\big) \tag{4}$$

where z_t is a sampled from the training set S_n uniformly at random, and $\nabla V(f_t, z_t)$ is an unbiased estimator of the full gradient of the empirical loss at f_t:

$$\hat{\mathbb{E}}_{z_t \sim S_n}[\nabla V(f_t, z_t)] = \nabla \hat{I}(f_t)$$

γ_t is a decreasing sequence of non-negative numbers, usually called the learning rates or step sizes. $\Pi_K : \mathscr{H} \to K$ is the projection map into K, and when $K = \mathscr{H}$, it becomes the identity map. It is interesting that the following equation, labeled SGDL, and studied by several authors, including [7], seem to work as well as or better than the usual repeat SGD used to train deep networks, as discussed in Section 5:

$$f_{t+1} = f_t - \gamma_n \nabla V(f_t, z_t) + \gamma_t' W_t. \tag{5}$$

Here W_t is a standard Gaussian vector in $\mathbb{R}^p$ and γ_t' is a sequence going to zero.

We consider a situation in which the expected cost function $I(f)$ can have, possibly multiple, global minima. As argued by [8] there are two ways to prove convergence of SGD. The first method consists of partitioning the parameter space into several attraction basins, assume that after a few iterations the algorithm confines the parameters in a single attraction basin, and proceed as in the convex case. A simpler method, instead of proving that the function f converges, proves that the cost function $I(f)$ and its gradient $\mathscr{E}_z \nabla V(f_t, z_t)$ converge.

Existing results show that when the learning rates decrease with an appropriate rate, and subject to relatively mild assumptions, stochastic gradient descent converges almost surely to a global minimum when the objective function is convex or pseudoconvex[2], and otherwise converges almost surely to a local minimum. This direct optimization shortcuts the usual discussion for batch ERM about differences between optimizing the empirical risk on S_n and the expected risk.

7. SGD implicit bias in the case of overparametrization

We conjecture that SGD, while minimizing the empirical loss, also implicitely maximizes the volume, that is ”flatness”, of the minima.

Our argument can be loosely described as follows. The zero minimizers are unique for $n \gg W$ and become degenerate , that

[2] In convex analysis, a pseudoconvex function is a function that behaves like a convex function with respect to finding its local minima, but need not actually be convex. Informally, a differentiable function is pseudoconvex if it is increasing in any direction where it has a positive directional derivative.

is flat, for $n \ll W$. Of course counting effective parameters is tricky in the case of deep net so the inequalities above should be considered just guidelines.

We consider the steps of our argument, starting with properties of SGD that have been mostly neglected from the machine learning point of view, to the best of our knowledge.

7.1. SGD as an approximate Langevin equation. We consider the usual SGD update defined by the recursion

$$f_{t+1} = f_t - \gamma_t \nabla V(f_t, z_t), \tag{6}$$

where z_t is fixed, $\nabla V(f_t, z_t)$ is the gradient of the loss with respect to f at z_t, and γ_t is a suitable decreasing sequence. When $z_t \subset [n]$ is a minibatch, we overload the notation and write $\nabla V(f_t, z_t) = \frac{1}{|z_t|}\sum_{z \in z_t} \nabla V(f_t, z)$.

We define an ”equivalent pseudo noise”

$$\xi_t = \nabla V(f_t, z_t) - \nabla I_{S_n}(f_t), \tag{7}$$

where it is clear that $\mathscr{E}\xi_t = 0$.

We then rewrite Equation 6 as

$$f_{t+1} = f_t - \gamma_t\big(\nabla I_{S_n}(f_t) + \xi_t\big). \tag{8}$$

With typical values used in minibatch (each minibatch corresponding to z_t) training, it turns out that the vector of gradient updates $\nabla V(f_t, z_t)$ empirically shows components with an approximate Gaussian distributions (see Fig. 6). This is expected because of the Central Limit Theorem (each minibatch involves sum over many random choices of datapoints).

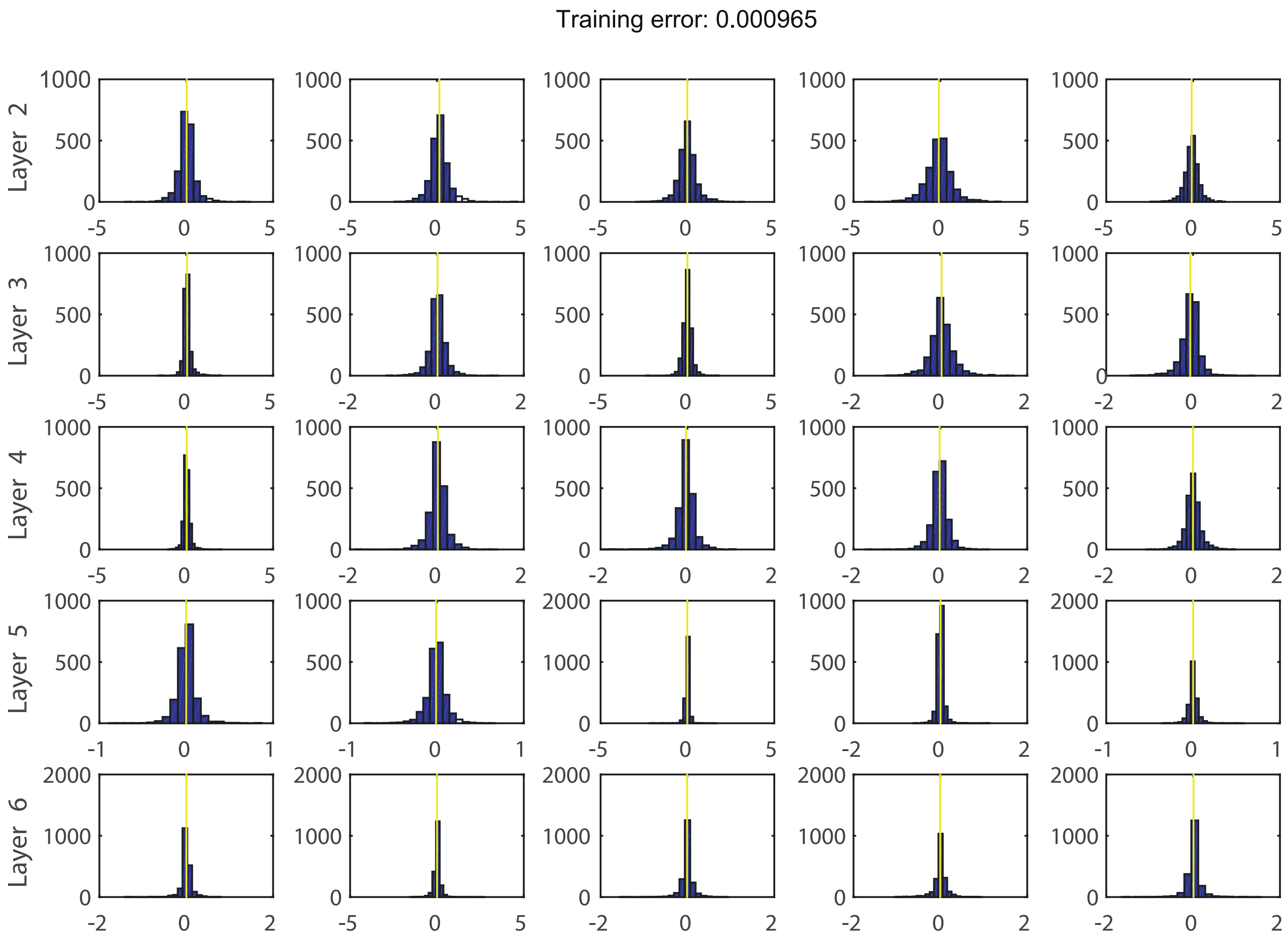


Fig. 6. Histograms of some of the components of $\nabla V(f_t, z_i)$ over i for fixed t in the asymptotic regime. Notice that the average corresponds to the gradient of the full loss, which is empirically very small. The histograms look approximatively Gaussian, as expected (see text) for minibatches that are not too small and not too large

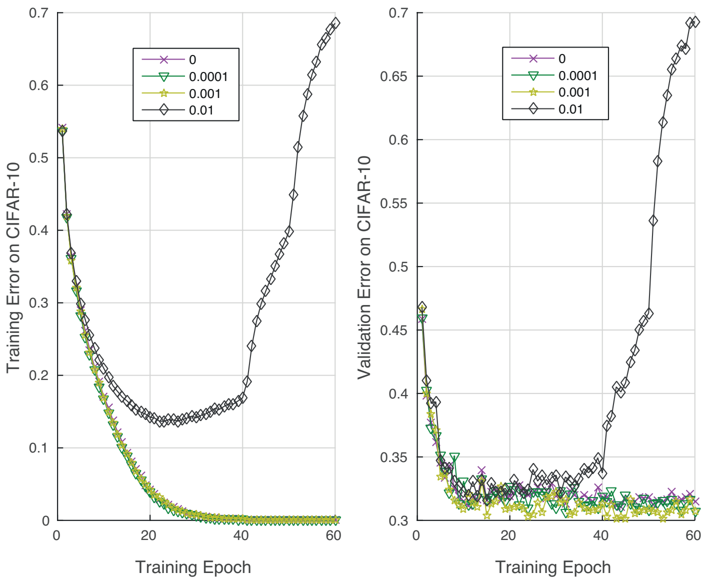


Fig. 7. Equation 5 – that is SGD with added Gaussian (with constant power) – behaves in a similar way to standard SGD. Notice that SGDL has slightly better validation performance than SGD

Now we observe that (8) is similar to a stochastic Langevin equation, with a noise scaled as γ_n rather than $\sqrt{\gamma_n}$. In fact, the continuous SGD dynamics corresponds to a stochastic gradient equation using a potential function defined by $U = I_{S_n}[f_t] = \frac{1}{n}\sum_{i=1}^{n} V(f, z_i)$ (see Proposition 3 and section 5 in [9]). If the noise were the derivative of the Brownian motion, this is a Langevin equation with an associated Fokker-Planck equation yielding the probability distributions of f_t. In particular, the stationary asympotic probability distribution is the Boltzman distribution given by $\approx e^{\frac{-U}{\gamma K}}$. For more details on stochastic dynamical systems, see for instance section 5 of [10]. Several proofs that adding a white noise term to equation (6) will make it converge to a global minimum are available (see [11]). Notice that the discrete version of the Langevin dynamics is equivalent to a Metropolis-Hastings algorithm for small learning rate (when the rejection step can be neglected).

7.2. SGDL concentrates at large volume, ”flat” minima. The argument about convergence of SGDL to large volume minima that we call ”flat”, is straighforward. The asymptotic distribution reached by a Langevin equation (GDL) –as well as by SGDL – is the Boltzman distribution that is

$$P(f) = \frac{1}{Z} e^{-\frac{U}{T}}, \tag{9}$$

where Z is a normalization constant, U is the loss and T reflects the noise power. The equation implies, and Fig. 9 shows, that SGD selects in probability degenerate minima rather than non-degenerate ones of the same depth. Among two minimum basins of equal depth, the one with a larger volume, is much more likely in high dimensions (Fig. 8). Taken together, these two facts suggest that SGD selects degenerate minimizers and, among those, the ones corresponding to larger flat valleys of the loss. If we assume that the landscape of the empirical minima is well-behaved in the sense that deeper minima have broader basin of attraction, we can then prove that SDGL shows concentration in probability – because of the high dimensionality – of its asymptotic distribution Equation 9 – to minima that are the most robust to perturbations of the weights. Notice that these assumptions are satisfied in the zero error case: among zero-minimizer, SGDL selects the ones that are flatter, i.e. have the larger volume.

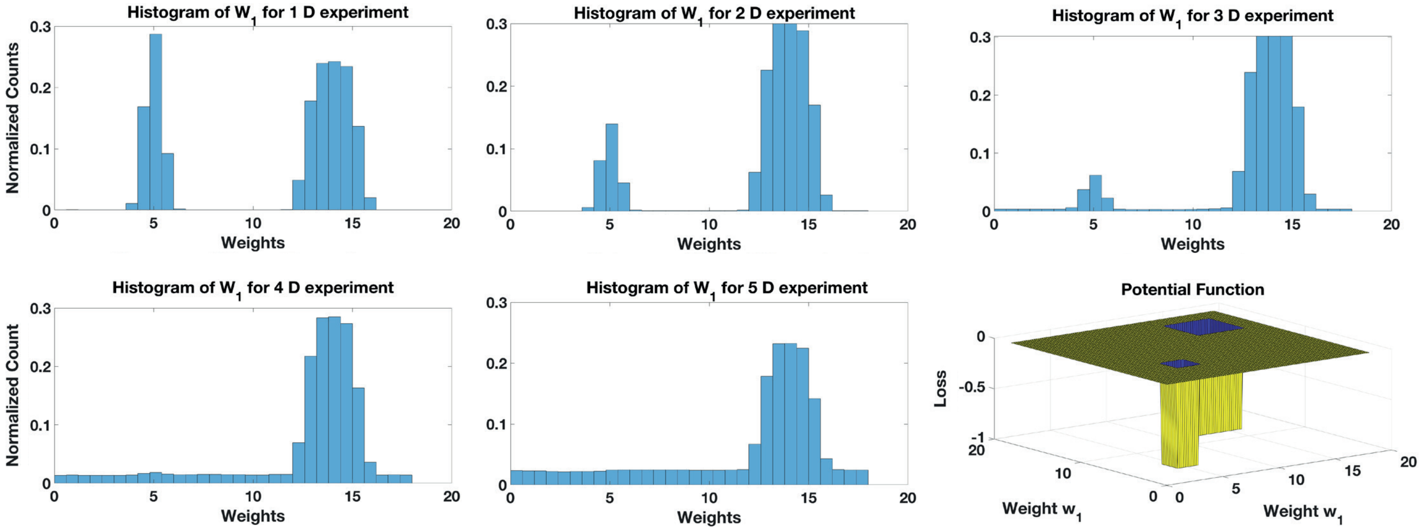


Fig. 8. The figure shows the histogram of a one-dimensional slice of the asymptotic distribution obtained by running Langevin Gradient Descent (GDL) on the potential surface on the right. The potential function has two minima with the same depth: one is wider (by a factor 2 in each dimension). The histogram for the first weight coordinate is shown here for dimensionality 1, 2, 3, 4 and 5. The figures show – as expected from the Boltzman distribution – that noisy gradient descent selects with high probability larger volume minimizers among minima of the same depth. As expected, higher dimensionality implies higher probability of selecting the flatter minimum

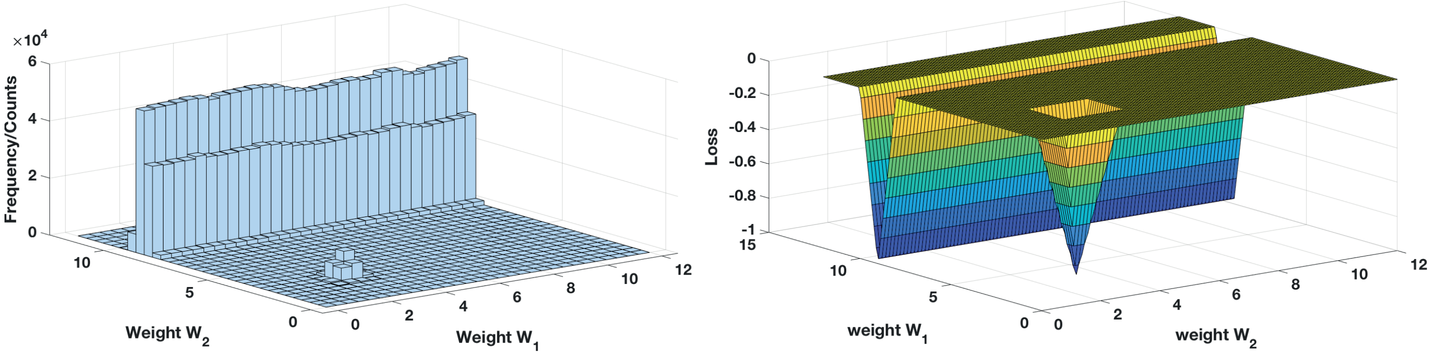


Fig. 9. Langevin Gradient Descent (GDL) on the 2D potential function shown above leads to an asymptotic distribution with the histogram shown on the left. As expected from the form of the Boltzman distribution, the Langevin dynamics prefers degenerate minima to non-degenrate minima of the same depth. In high dimensions we expect the asymptotic distribution to concentrate strongly around the degenerate minima as confirmed by Fig. 10

In [2] we review qualitative arguments of why flat minima"may" imply robust optimization and maximization of margin. Here we claim that SGDL and SGD maximize volume and "flatness" of the loss in weight space. Given a flat, degenerate minimum, one may ask where SGD will converge to. For situations such as in Fig. 10 and for a minimum such as in Fig. 11, Theory III suggests a locally minimum norm solution.

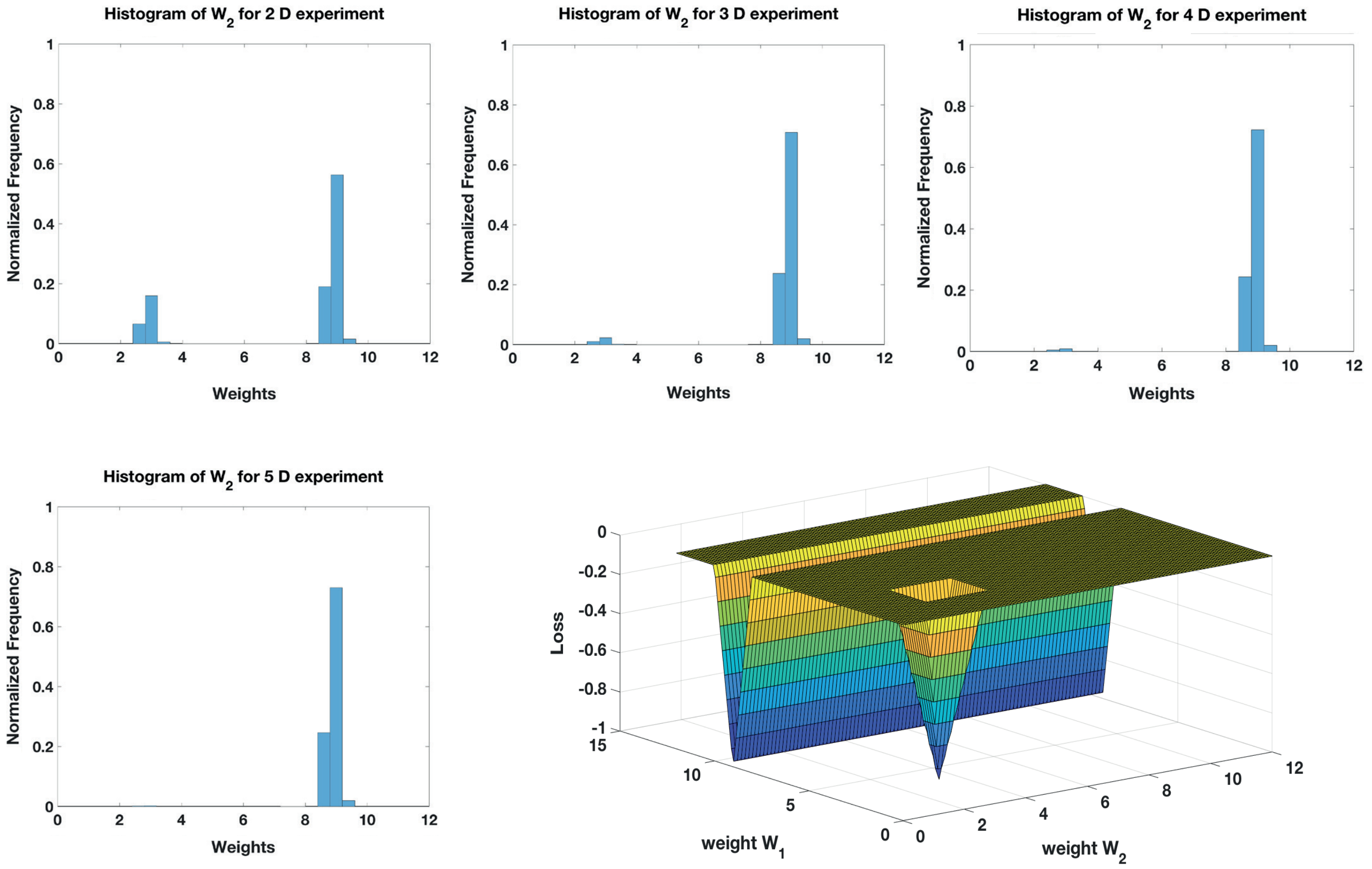


Fig. 10. The figure shows the histogram of a one-dimensional slice of the asymptotic distribution obtained by running Langevin Gradient Descent (GDL) on the potential surface on the right

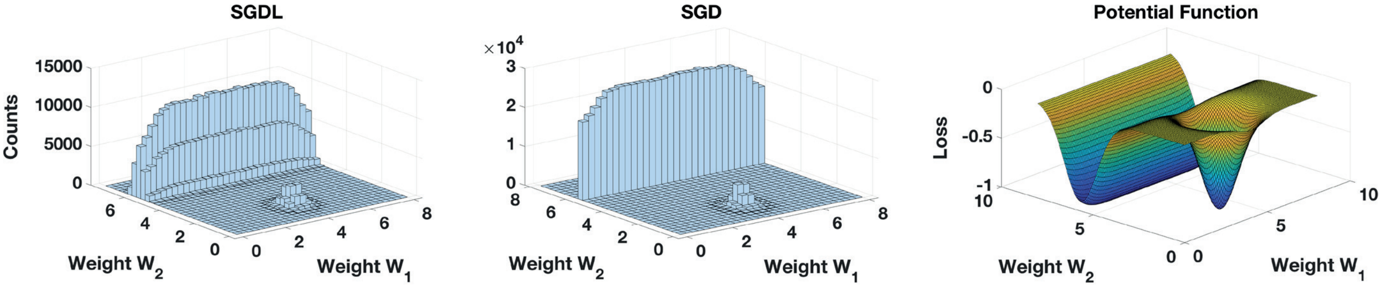


Fig. 11. Stochastic Gradient Descent and Langevin Stochastic Gradient Descent (SGDL) on the 2D potential function shown above leads to an asymptotic distribution with the histograms shown on the left. As expected from the form of the Boltzman distribution, both dynamics prefers degenerate minima to non-degenerate minima of the same depth

8. Random labels

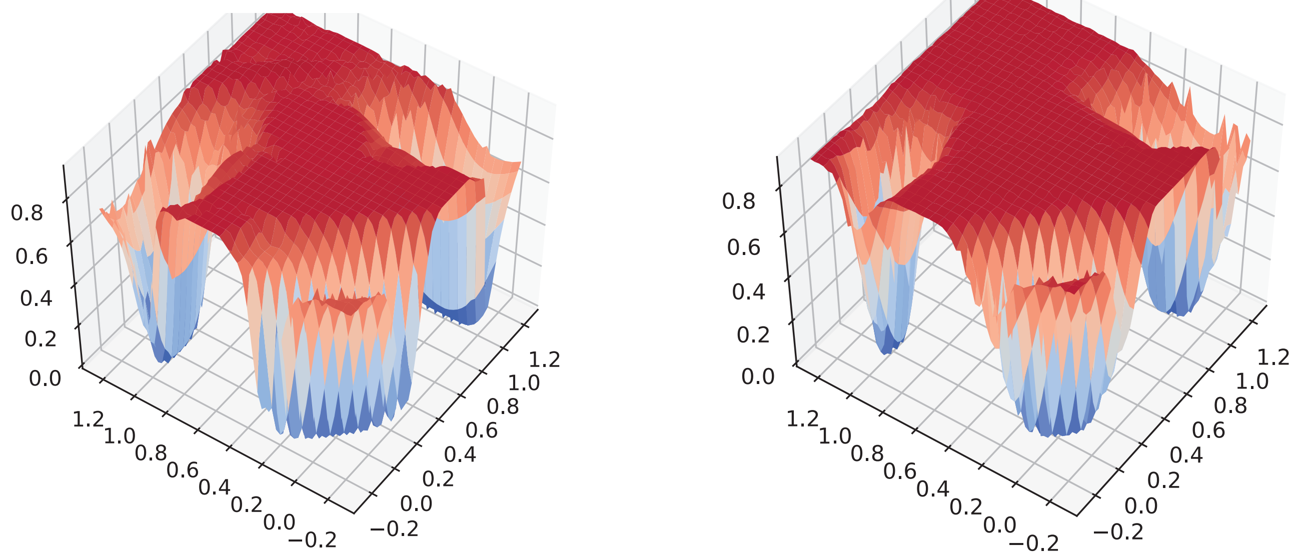
For this case, Part A predicts that it is in fact possible to interpolate the data on the training set, that is to achieve zero empirical error (because of overparametrization) and that this is in fact easy – because of the very high number of zeros of the polynomial approximation of the network– assuming that the target function is in the space of functions realized by the network. For n going to infinity we expect that the empirical error will converge to the expected (which is at chance level here), as shown in the figures. For finite $n < W$, the fact that the empirical error (which is zero) is so different from the expected seems puzzling, as observed by [12], especially because the algorithm is capable of low expected error with the same n for natural labels.

Table 1
The ”flatness test”: at the minimizer, we move the weights around in a random direction, and measure the furthest distance until the objective function is increased by ε (0.05), and then measure the average distance

	MNIST	CIFAR-10
all params	45.4 ± 2.7	17.0 ± 2.4
all params (random label)	6.9 ± 1.0	5.7 ± 1.0
top layer	15.0 ± 1.7	19.5 ± 4.0
top layer (random label)	3.0 ± 0.1	12.1 ± 2.6

A larger margin is found for natural labels than for random labels as shown in Table 1 and in Fig. 12 and Fig. 13. Figure 12 shows ”three-point interpolation” plots to illustrate the flatness of the landscape around global minima of the empirical loss found by SGD, on CIFAR-10, with natural labels and random labels, respectively. Specifically, let w_1, w_2, w_3 be three minimizers for the empirical loss found by SGD. For $\lambda = (\lambda_1, \lambda_2, \lambda_3)$ on the simplex Δ_3, let

$$w_\lambda = \lambda_1 w_1 + \lambda_2 w_2 + \lambda_3 w_3. \tag{10}$$


a) Natural label

b) Random label

Fig. 12. Illustration of the landscape of the empirical loss on CIFAR-10

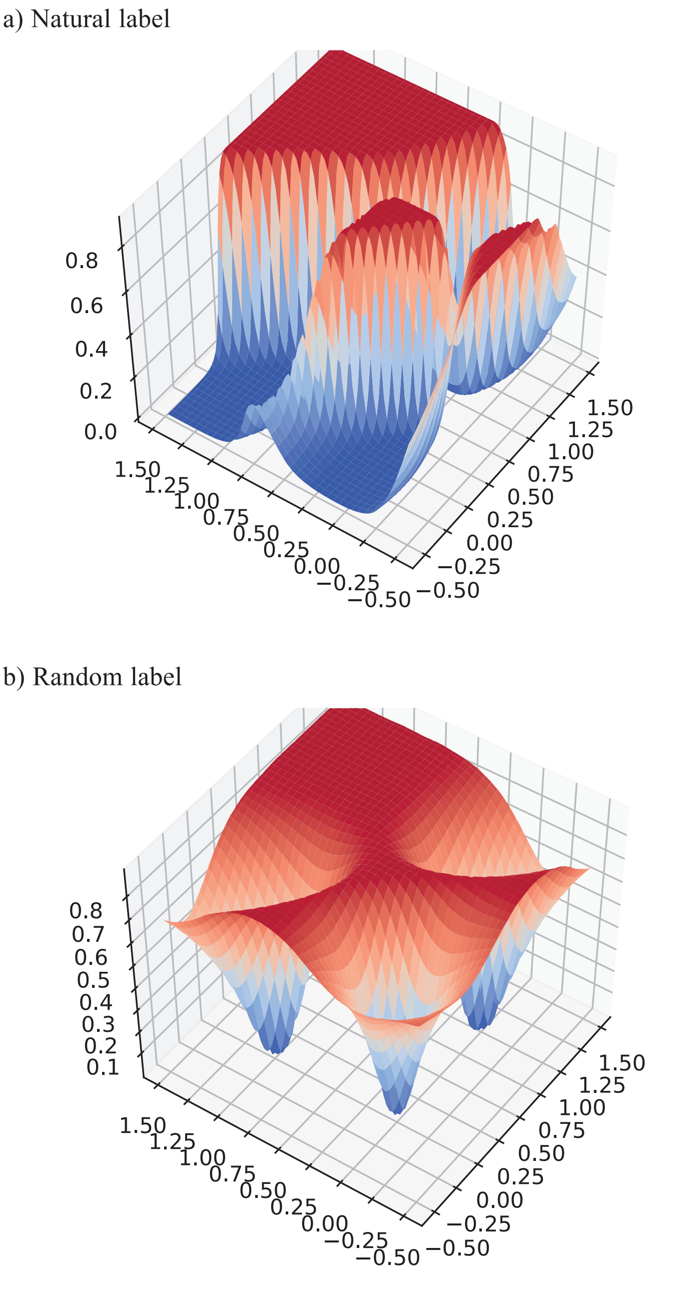


Fig. 13. Illustration of the landscape of the empirical loss on MNIST

We then evaluate the training accuracy for the model defined by each interpolated weights w_λ and make a surface plot by embedding Δ_3 in the 2D X-Y plane. As we can see, the natural label case depict a larger flatness region around each of the three minima than the random label case. There is a direct relation between the range of flatness and the norm λ of the perturbations.

The same phenomenon could be observed more clearly on the MNIST dataset, where the images of the same category are already quite similar to each other in the pixel space, making it more difficult to fit when random labels are used. Therefore, the difference in the characteristics of the landscapes is amplified. As shown in Fig. 13, large flat regions are observed in the natural label case, while the landscape for the random label experiment shows sharper wells.

It is difficult to visualize the flatness of the landscape when the weights are typically in the scale of one million dimensions. To assess flatness, we employ the following procedure around a minimum found by SGD: choose a random direction δw with $\|\delta w\| = 1$, perform a line search to find the "flatness radius" in that direction:

$$r(w, \delta w, \varepsilon) = \sup\left\{r : \left|\hat{I}(w) - \hat{I}(w + r\delta w)\right| \leq \varepsilon\right\}. \quad (11)$$

The procedure is repeated T times and the average radius is calculated. The overall procedure is also repeated multiple times to test the average flatness at different minima. The results are shown in Table 1. For both CIFAR-10 and MNIST, we observe a difference between the natural label and random label.

9. Part A and Part B: summary

In this paper (as a review of [1–3]), we have described properties of the landscape of the empirical risk, for the case of overparametrized convolutional deep networks. Zero empirical error yields a system of polynomial equations that yields a very large number of global minima – when is not inconsistent – which are degenerate, that is flat in several of the dimensions (in CIFAR there are about 10^6 unknown parameters for 6×10^4 equations. The zero empirical error minimizers are global and degenerate while the local minima are in general not degenerate. Consistently with this predictions, we empirically observe that zero-minimizers correspond to flat valleys. We furthermore show that SGD, with respect to GD, is biased to find with high probability degenerate minimizers – flat "valleys" in the landscape – which are likely to be global minima.

Acknowledgment. This work was supported in part by the Center for Brains, Minds and Machines (CBMM), funded by NSF STC award CCF – 1231216 and in part by C-BRIC, one of six centers in JUMP, a Semiconductor Research Corporation (SRC) program sponsored by DARPA, the National Science Foundation, Intel Corporation, and the DoD Vannevar Bush Fellowship. We gratefully acknowledge the support of NVIDIA Corporation with the donation of the DGX-1 used for this research.

References

[1] T. Poggio and Q. Liao, "Theory II: Landscape of the empirical risk in deep learning," *arXiv preprint arXiv:1703.09833*, 2017.

[2] C. Zhang, Q. Liao, A. Rakhlin, B. Miranda, N. Golowich, and T. Poggio, "Musings on deep learning: Properties of SGD".

[3] C. Zhang, Q. Liao, A. Rakhlin, B. Miranda, N. Golowich, and T. Poggio, "Theory of deep learning IIB: Optimization properties of SGD," *arXiv preprint arXiv:1801.02254*, 2018.

[4] M. Shub and S. Smale, "Complexity of bezout theorem V: Polynomial time," *Theoretical Computer Science*, no. 133, pp. 141–164, 1994.

[5] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.

[6] I. Borg and P.J. Groenen, *Modern Multidimensional Scaling: Theory and Applications*. Springer Science & Business Media, 2005.

[7] S. Gelfand and S. Mitter, "Recursive stochastic algorithms for global optimization in R^d", *Siam J. Control and Optimization*, vol. 29, pp. 999–1018, September 1991.

[8] L. Bottou, "Online algorithms and stochastic approximations," in *Online Learning and Neural Networks* (D. Saad, ed.), Cambridge, UK: Cambridge University Press, 1998, revised, oct 2012.

[9] D. Bertsekas and J. Tsitsiklis, "Gradient convergence in gradient methods with errors," *SIAM J. Optim*. 10, 627–642 (2000).

[10] D.P. Bertsekas and J.N. Tsitsiklis, *Neuro-dynamic Programming*. Athena Scientific, Belmont, MA, 1996.

[11] B. Gidas, "Global optimization via the Langevin equation," *Proceedings of the 24th IEEE Conference on Decision and Control*, pp. 774–778, 1985.

[12] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, "Understanding deep learning requires rethinking generalization," in *International Conference on Learning Representations (ICLR)*, 2017.