Proceedings of Machine Learning Research vol 145:1–21, 2021 2nd Annual Conference on Mathematical and Scientific Machine Learning

On the emergence of simplex symmetry in the final and penultimate layers of neural network classifiers

Weinan E WEINAN@MATH.PRINCETON.EDU
Program for Applied and Computational Mathematics
Princeton University
Princeton, NJ 08544

Stephan Wojtowytsch STEPHANW@PRINCETON.EDU
Program for Applied and Computational Mathematics
Princeton University
Princeton, NJ 08544

Editors: Joan Bruna, Jan S Hesthaven, Lenka Zdeborova

Abstract

A recent numerical study observed that neural network classifiers enjoy a large degree of symmetry in the penultimate layer. Namely, if $h(x) = Af(x) + b$ where A is a linear map and f is the output of the penultimate layer of the network (after activation), then all data points $x_{i,1}, \dots, x_{i,N_i}$ in a class C_i are mapped to a single point y_i by f and the points y_i are located at the vertices of a regular $k-1$-dimensional standard simplex in a high-dimensional Euclidean space.

We explain this observation analytically in toy models for highly expressive deep neural networks. In complementary examples, we demonstrate rigorously that even the final output of the classifier h is not uniform over data samples from a class C_i if h is a shallow network (or if the deeper layers do not bring the data samples into a convenient geometric configuration).

Keywords: Classification problem, deep learning, neural collapse, cross entropy, geometry within layers, simplex symmetry

1. Introduction

A recent empirical study Papyan et al. (2020) took a first step towards investigating the inner geometry of neural networks close to the output layer. In classification problems, the authors found that the data in the final and penultimate layers enjoy a high degree of symmetry. Namely, a neural network function $h_L : \mathbb{R}^d \to \mathbb{R}^k$ with L layers can be understood as a composition

$$h_L(x) = A\, f_L(x) + b \tag{1}$$

where $f_L : \mathbb{R}^d \to \mathbb{R}^m$ is (the composition of a componentwise nonlinearity with) a neural network with $L-1$ layers, $b \in \mathbb{R}^k$ and $A : \mathbb{R}^m \to \mathbb{R}^k$ is linear. In applications where h_L was trained by stochastic gradient descent to minimize softmax-crossentropy loss to distinguish elements in various classes $C_1, \dots, C_k$, the authors observed that the following became approximately true in the long time limit.

- f_L maps all elements in a class C_i to a single point y_i.

- The distance between the centers of mass of different classes in the penultimate layer $\|y_i - y_j\|$ does not depend on $i \neq j$.
- Let $M = \frac{1}{k}\sum_{i=1}^k y_i$ be the center of mass of the data distribution in the penultimate center (normalizing the weight of data classes). Then the angle between $y_i - M$ and $y_j - M$ does not depend on $i \neq j$.
- The i-th row of A is parallel to $y_i - M$.

In less precise terms, h_L maps the classes C_i to the vertices of a regular standard simplex in a high-dimensional space. This phenomenon is referred to as 'neural collapse' in Papyan et al. (2020). In this note, we consider the toy model where f_L is merely a bounded measurable function and prove that under certain assumptions such simplex geometries are optimal. An investigation along the same lines has been launched separately in Mixon et al. (2020).

Conversely, we show that even the output $h_L(C_i)$ of a shallow neural network h_L over a data class C_i does not approach a single value z_i when the parameters of h_L are trained by continuous time gradient descent. Since a deep neural network is the composition of a slightly less deep network and a shallow neural network containing the output layer, these results suggest that the h_L cannot be expected to be uniform over a data class unless a convenient geometric configuration has already been reached two layers before the output.

We make the following observations.

1. Overparametrized networks can fit random labels at data points Cooper (2018) and can be efficiently optimized for this purpose in certain scaling regimes, see e.g. Du et al. (2018a,b); E et al. (2020). The use of the class $L^\infty(\mathbb{P};\mathbb{R}^m) := (L^\infty(\mathbb{P}))^m$ as a proxy for very expressive deep neural networks thus can be justified heuristically from the static perspective of energy minimization (but not necessarily from the dynamical perspective of training algorithms).

   In practice, the data distribution $\mathbb{P}$ is estimated on a finite set of sample points $\{x_1, \dots, x_N\}$ and an empirical distribution $\mathbb{P}_N = \frac{1}{N}\sum_{i=1}^N \delta_{x_i}$. A function $f_L \in L^\infty(\mathbb{P};\mathbb{R}^m)$ determined by its values at the points $x_1, \dots, x_N$. A class of sufficiently complex neural networks which can fit any given set of outputs $\{y_1, \dots, y_N\}$ for inputs $\{x_1, \dots, x_N\}$ coincides with $L^p(\mathbb{P};\mathbb{R}^m)$ for any $1 \leq p \leq \infty$. The same is true for many other function models.

   If $\mathbb{P}_N = \frac{1}{N}\sum_{i=1}^N \delta_{x_i}$ or more generally, if all classes $C_1, \dots, C_k$ have a positive distance to each other, a function $f \in L^p(\mathbb{P};\mathbb{R}^m)$ which is constant on every class can be extended to a C^∞-function on $\mathbb{R}^d$. Thus in realistic settings, all functions below can be taken to be fairly regular.

2. As the softmax cross-entropy functional does not have minimizers in sufficiently expressive scaling-invariant function classes, we need to consider norm bounded classes.

   In the hypothesis class given by the ball of radius R in $L^\infty(\mathbb{P};\mathbb{R}^m)$, the optimal map h satisfies $h(x) = z_i$ for all x in a data class C_i and the values z_i form the vertices of a regular simplex. More precisely, the statement is valid under the constraint $\|h(x)\|_{\ell^p} \leq R$ for all $p \in (1, \infty)$, but the precise location of the vertices depends on p. We refer to this as final layer geometry.

   If $h : \mathbb{R}^d \to \mathbb{R}^k$ is given by $h(x) = A\,f(x)$ for $f \in L^\infty(\mathbb{P};\mathbb{R}^m)$ and a linear map $A : \mathbb{R}^m \to \mathbb{R}^k$, the following holds: If $\|A\|_{L(\ell^2,\ell^2)} \leq 1$ and $\|f(x)\|_{\ell^2} \leq R$ for all $x \in \mathbb{R}^d$,

then any energy minimizer satisfies $f(x) = y_i$ for all $x \in C_i$ where the outputs y_i form the vertices of a regular standard simplex in a high-dimensional ambient space. We refer to this as penultimate layer geometry. We note that similar results were obtained in a different framework in Lu and Steinerberger (2020).

3. Considerations on the final layer geometry are generally independent of the choice of norm on $\mathbb{R}^k$ within the class of ℓ^p-norms, while the penultimate layer geometry appears to depend specifically on the use of the Euclidean norm. While the coordinate-wise application of a one-dimensional activation function is not hugely compatible with Euclidean geometry (or at least no more compatible than with ℓ^p-geometry for any $p \in [1, \infty]$), the transition from the penultimate layer to the final layer is described by a single affine map $y \mapsto Ay + b$. If A and b are initilized from a distribution compatible with Euclidean geometry (e.g. a rotation-invariant Gaussian) and optimized by an algorithm such as gradient descent which is based on the Euclidean inner product, then the use of Euclidean geometry for (A, b) is well justified.

   In deeper layers, the significance of Euclidean geometry becomes more questionable. Even for the map $f : \mathbb{R}^d \to \mathbb{R}^m$, it is unclear whether the Euclidean norm captures the constraints on f well.

4. If $h(x) = \sum_{i=1}^m a_i \, \sigma(w_i^T x + b_i)$ is a shallow neural network classifier and the weights (a_i, w_i, b_i) are optimized by gradient descent, then in general h does *not* converge to a classifier which is constant on different data classes (although the hypothesis class contains functions with arbitrarily low risk which are constant on the different classes C_i). This is established in different geometries:

   (a) In the first case, σ is the ReLU activation function and the classes are linearly separable. Under certain conditions, gradient descent approaches a maximum margin classifier, which can be a linear function and thus generally non-constant over the data classes.

   (b) In the second case, σ is constant for large arguments and there are three data points x_1, x_2, x_3 on a line where x_1, x_3 belong to the same class, but the middle point x_2 belongs to a different class. Then the values of h at x_1, x_2, x_3 cannot be chosen independently due to the linear structure of the first layer, and the heuristic behind the toy model does not apply.

   Note that h is of the form $h = Af$, but $f(x) = \sigma(Wx)$ is not sufficiently expressive for the analysis of the *penultimate* layer to apply.

The theoretical analysis raises further questions. As the expressivity of the hypothesis class and the ability to set values on the training set with little interaction between different point evaluations seems crucial to the 'neural collapse' phenomenon, we must question whether this simple geometric configuration is in fact desirable, or merely the optimal configuration in a hypothesis class which is too large to allow any statistical generalization bounds. Such concerns were already raised in Elad et al. (2020). While the latter possibility is suggested by the theoretical analysis, it should be emphasized that in the numerical experiments in Papyan et al. (2020) solutions with good generalization properties are found. This compatibility could be explained by considering a hypothesis class which is not as expressive as $L^\infty(\mathbb{P}; \mathbb{R}^m)$, but contains a function which attains a desired set of values on a realistic data set.

It should be noted that the final layer results apply to any sufficiently expressive function class, not just neural networks. The results for the penultimate layer apply to classes of classifiers which are compositions of a linear function and a function in a very expressive function class. In both cases, we consider (norm-constrained) energy minimizers, not training dynamics. If the norm constraints are meaningful for a function model and an optimization algorithm can find the minimizers, the analysis applies in the long time limit, but the dynamics would certainly depend on the precise function model. This coincides with the situation considered by Papyan et al. (2020), in which the cross-entropy is close to zero after significant training.

If $h = Af$ and f is not sufficiently expressive (as in two-layer neural networks), we observe that classifier collapse does not occur, even in the final layer. Whether there are further causes driving classifier collapse in deep neural networks remains to be seen.

We believe that further investigation in this direction is needed to understand the following: Is neural collapse observed on random data sets or real data sets with randomly permuted labels? Does it occur also on test data or just training data? Is neural collapse observed for ReLU activation functions, or only for activation functions which tend to a limit at positive and negative infinity? Do the outputs over different classes y_i attain a regular simplex configuration also if the weights of the different data classes are vastly different? Is neural collapse observed if a parameter optimization algorithm is used which does not respect Euclidean geometry (e.g. an algorithm with coordinatewise learning rates such as ADAM)? The question when neural collapse occurs and whether it helps generalization in deep learning remains fairly open.

The article is structured as follows. In Section 2, we rigorously introduce the problem we will be studying and obtain some first properties. In Sections 3 and 4, we study a toy model for the geometry of the output layer and penultimate layer of a neural network classifier respectively. In Section 5, we present analytic examples in simple situations where neural network classifiers behave markedly differently and where the toy model analysis does not apply.

1.1. Notation

We consider classifiers $h : \mathbb{R}^d \to \mathbb{R}^k$ in a hypothesis class $\mathcal{H}$. Often, h will be assumed to be a general function on a finite set with norm-bounded output, or the composition of such a function $f : \mathbb{R}^d \to \mathbb{R}^m$ and a linear map $A : \mathbb{R}^m \to \mathbb{R}^k$ for some $m \geq 1$. Variables in $\mathbb{R}^d, \mathbb{R}^m$ and $\mathbb{R}^k$ are denoted by x, y and z respectively.

2. Preliminaries

2.1. Set-up

A classification problem is made up of the following ingredients:

1. A *data distribution*, i.e. a probability measure $\mathbb{P}$ on $\mathbb{R}^d$.
2. A *label function*, i.e. a $\mathbb{P}$-measurable function $\xi : \mathbb{R}^d \to \{e_1, \dots, e_k\} \subset \mathbb{R}^k$. We refer to the sets $C_i = \xi^{-1}(\{e_i\})$ as the classes.
3. A *hypothesis class*, i.e. a class $\mathcal{H}$ of functions $h : \mathbb{R}^d \to \mathbb{R}^k$ for $d \gg 1$ and $k \geq 2$.
4. A *loss function* $\ell : \mathbb{R}^k \times \mathbb{R}^k \to [0, \infty)$.

We always assume that $\mathcal{H} \subseteq L^1(\mathbb{P};\mathbb{R}^k)$ and often even $\mathcal{H} \subseteq L^\infty(\mathbb{P};\mathbb{R}^k)$. These ingredients are combined in the *risk functional*

$$\mathcal{R} : \mathcal{H} \to [0,\infty), \qquad \mathcal{R}(h) = \int_{\mathbb{R}^d} \ell\big(h(x),\xi_x\big)\,\mathbb{P}(\mathrm{d}x), \tag{2}$$

which is approximated by the *empirical risk functional*

$$\widehat{\mathcal{R}}_n(h) = \frac{1}{n}\sum_{i=1}^n \ell\big(h(x_i),\xi_i\big)$$

where x_i are samples drawn from the distribution $\mathbb{P}$ and $\xi_i = \xi_{x_i}$. Since we can write

$$\widehat{\mathcal{R}}_n(h) = \int_{\mathbb{R}^d} \ell\big(h(x),\xi_x\big)\,\mathbb{P}_n(\mathrm{d}x), \qquad \mathbb{P}_n = \frac{1}{n}\sum_{i=1}^n \delta_{x_i},$$

we do not differentiate between empirical risk and (population) risk in this article. This allows us to organically incorporate that all results are independent of the number of data points. We focus on the *softmax cross entropy risk functional* associated to the loss function

$$\ell\big(h,y\big) = -\log\left(\frac{\exp(h\cdot y)}{\sum_{i=1}^k \exp(h\cdot e_k)}\right). \tag{3}$$

This loss function allows the following probabilistic interpretation: For given a given classifier $h \in \mathcal{H}$ and data point $x \in \mathbb{R}^d$, the vector π with entries

$$\pi_i(x) := \frac{\exp(h(x)\cdot e_i)}{\sum_{j=1}^k \exp(h(x)\cdot e_j)}$$

is a counting density on the set of labels $\{1,\dots,k\}$, depending on the input x. The function

$$\Phi : \mathbb{R}^k \to \mathbb{R}^k, \qquad \Phi(h) = \left(\frac{\exp(h\cdot e_1)}{\sum_{i=1}^k \exp(h\cdot e_i)},\dots,\frac{\exp(h\cdot e_k)}{\sum_{i=1}^k \exp(h\cdot e_i)}\right)$$

which converts a k-dimensional vector into a counting density is referred to as the softmax function since it approximates the maximum coordinate function of h for large inputs. The cross-entropy (Kullback-Leibler divergence) of this distribution with respect to the distribution $\bar{\pi}(x)$ which gives the correct label with probability 1 is precisely

$$-\sum_{j=1}^k \log\left(\frac{\pi_j(x)}{\bar{\pi}_j(x)}\right)\bar{\pi}(x) = -\log\left(\frac{\pi_{i(x)}}{1}\right)\cdot 1 = -\log\left(\frac{\exp(h(x)\cdot\xi_x)}{\sum_{i=1}^k \exp(h(x)\cdot e_k)}\right)$$

since $\bar{\pi}_j = \delta_{j,i(x)}$ and $0\cdot\log(\infty) = 0$ in this case by approximation. The risk functional thus is the average integral of the pointwise cross-entropy of the softmax counting densities with respect to the true underlying distribution.

Note the following: $\ell > 0$, but if h is such that $h(x) \cdot \xi_x > \max_{e_1,\dots,e_k \neq \xi_x} h(x) \cdot e_i$ for $\mathbb{P}$-almost every x, then

$$\lim_{\lambda\to\infty} \mathcal{R}(\lambda h) = \lim_{\lambda\to\infty} -\int_{\mathbb{R}^d} \log\left(\frac{\exp(\lambda h(x)\cdot\xi_x)}{\sum_{i=1}^k \exp(\lambda h(x)\cdot e_k)}\right) \mathbb{P}(\mathrm{d}x) = 0.$$

Thus the cross-entropy functional does not have minimizers in suitably expressive function classes which are cones (i.e. $f \in \mathcal{H}, \lambda > 0 \Rightarrow \lambda f \in \mathcal{H}$). So to obtain meaningful results by energy minimization, we must consider

1. a dynamical argument concerning a specific optimization algorithm, or
2. a restricted hypothesis class with meaningful norm bounds, or
3. a higher order expansion of the risk.

We follow the first line of inquiry for shallow neural networks in Section 5 and the second line of inquiry for toy models for deep networks in Sections 3 and 4.

2.2. Convexity of the loss function

For the following, we note that the softmax cross entropy loss function has the following convexity property.

Lemma 1 *The function*

$$\Phi_j : \mathbb{R}^k \to \mathbb{R}, \qquad \Phi(z) = -\log\left(\frac{\exp(z_j)}{\sum_{i=1}^k \exp(z_i)}\right) = \log\left(\sum_{i=1}^k \exp(z_i)\right) - z_j$$

is convex for any $1 \leq j \leq k$ and strictly convex on hyperplanes H_α of the form

$$H_\alpha = \left\{ z \in \mathbb{R}^k : \sum_{j=1}^k z_j = \alpha \right\}.$$

For the sake of completeness, we provide a proof in the Appendix. Since $\Phi\big(z + \lambda(1,\dots,1)\big) = \Phi(z)$ for all $\lambda \in \mathbb{R}$, we note that Φ_j is *not* strictly convex on the whole space $\mathbb{R}^d$.

3. Heuristic geometry: final layer

3.1. Collapse to a point

In this section, we argue that the output $h(C_i)$ of the classifier should be a single point for all classes C_i, $i = 1,\dots,k$ if the hypothesis class is sufficiently expressive. We will discuss the penultimate layer below.

Lemma 2 *Let $h \in \mathcal{H}$ and set*

$$z_i := \frac{1}{|C_i|}\int_{C_i} h(x')\,\mathbb{P}(\mathrm{d}x'), \qquad \bar{h}(x) = z_i \quad \textit{for all } x \in C_i.$$

Then $\mathcal{R}(\bar{h}) \leq \mathcal{R}(h)$ and equality holds if and only if there exists a function $\lambda \in L^1(\mathbb{P})$ such that $h - \bar{h} = \lambda(1,\dots,1)$ $\mathbb{P}$-almost everywhere.

The reasoning behind the Lemma is that

$$\begin{aligned}
\int_{C_i} \Phi_i(h(x))\,\mathbb{P}(\mathrm{d}x) &\approx \int_{C_i} \Phi_i\left(z_i\right) + \nabla\Phi_i(z_i)\cdot\big(h(x)-z_i\big) + \frac{1}{2}\big(h(x)-z_i\big)^T D^2\Phi_i(z_i)\,\big(h(x)-z_i\big)\,\mathbb{P}(\mathrm{d}x)\\
&= \int_{C_i} \Phi_i\left(z_i\right)\,\mathbb{P}(\mathrm{d}x) + \nabla\Phi_i(z_i)\cdot\int_{C_i} h(x) - z_i\,\mathbb{P}(\mathrm{d}x)\\
&\quad + \frac{1}{2}\int_{C_i}\big(h(x)-z_i\big)^T D^2\Phi_i(z_i)\,\big(h(x)-z_i\big)\,\mathbb{P}(\mathrm{d}x)\\
&\geq \int_{C_i} \Phi_i\left(z_i\right)\,\mathbb{P}(\mathrm{d}x)
\end{aligned}$$

since the first order term vanishes. A summation over i establishes the result. A rigorous proof using Jensen's inequality can be found in the appendix.

Thus if a class C_j is mapped to a set $h(C_j) \subseteq \mathbb{R}^k$ with a prescribed center of mass, different classes are mapped to the same centers of mass, it is energetically favorable to reduce the variance to the point that $h(C_j)$ is a single point. Whether or not this is attainable depends primarily on the hypothesis class $\mathcal{H}$, but a very expressive class like deep neural networks is likely to allow this collapse to a single point.

Corollary 3 *If $\mathcal{H} = L^\infty(\mathbb{P}; V)$ is the class of bounded $\mathbb{P}$-measurable functions which take values in a compact convex set $V \subset \mathbb{R}^k$, then a minimizer h or $\mathcal{R}$ in $\mathcal{H}$ can be taken to map the class C_i to a single point $z_i \in V$ for all $i = 1,\dots,k$, and all other minimizer differ from h only in direction $(1,\dots,1)$.*

3.2. Simplex configuration

In this section, we discuss the emergence of the simplex configuration under the assumption that the every class gets mapped to a single point $z_i \in \mathbb{R}^k$, or equivalently that each class consists of a single data point. Again, we consider the *last layer problem*: Assume that

- $\mathcal{X} = \{x_1,\dots,x_d\}$,
- $\mathcal{H}$ is the class of functions from $\mathcal{X}$ to the Euclidean ball $B_R(0)$ in $\mathbb{R}^k$.

Let $\mathbb{P}$ be a probability measure on $\mathcal{X}$ and $p_i := \mathbb{P}(\{x_i\})$. We wish to solve the minimization problem $h^* \in \mathrm{argmin}_{h\in\mathcal{H}}\,\mathcal{R}(h)$ where

$$\mathcal{R}(h) = \int_{\mathcal{X}} -\log\left(\frac{\exp(h(x)\cdot\xi_x)}{\sum_{i=1}^d \exp(h(x)\cdot e_i)}\right)\,\mathbb{P}(\mathrm{d}x) = -\sum_{i=1}^d p_i \log\left(\frac{\exp(h(x_i)\cdot e_i)}{\sum_{j=1}^d \exp(h(x_i)\cdot e_j)}\right).$$

Due to our choice of hypothesis class, there is no interaction between $h(x_i)$ and $h(x_j)$, so we can minimize the sum term by term:

$$z_i := h(x_i) \in \operatorname*{argmin}_{z \in B_R(b)} \left(-\log \left(\frac{\exp(z \cdot e_i)}{\sum_{j=1}^d \exp(z \cdot e_j)} \right) \right) = \min_{z \in B_R(b)} \Phi_i(z)$$

where $\Phi_i(z) = \log\left(\sum_{j=1}^k \exp(z_k)\right) - z_i$ is as in Lemma 1.

Lemma 4 *For every i there exists a unique minimizer z_i of Φ_i in $B_R(0)$ and $z_i = \alpha\, e_i + \beta \sum_{j \neq i} e_j$ for $\alpha, \beta \in \mathbb{R}$ which do not depend on i.*

Since $\Phi\big(z + \lambda(1, \dots, 1)\big) = \Phi(z)$ for all $\lambda \in \mathbb{R}$, the same result holds for the ball $B_R\big(\lambda(1, \dots, 1)\big)$ with any $\lambda \in \mathbb{R}$. We can determine the minimizers by exploiting the relationships

$$\alpha^2 + (k-1)\beta^2 = R^2, \qquad \alpha + (k-1)\beta = 0$$

which are obtained from the Lagrange-multiplier equation (9) in the proof of Lemma 4. The equations reduce to

$$\alpha = (k-1)\sqrt{\frac{R^2 - \alpha^2}{k-1}} = \sqrt{(k-1)\,(R^2 - \alpha^2)} \quad \Rightarrow \quad \alpha^2 = (k-1)\,(R^2 - \alpha^2)$$

and ultimately

$$\alpha^2 = \frac{k-1}{k}\, R^2 \quad \Rightarrow \quad \alpha = \sqrt{\frac{k-1}{k}}\, R, \quad \beta = -\frac{1}{k-1}\alpha = -\frac{R}{\sqrt{k(k-1)}}. \tag{4}$$

Remark 5 *Lemma 2 remains true when $B_R(0)$ is the ball of radius $R > 0$ with respect to an ℓ^p-norm on $\mathbb{R}^k$ for $1 < p < \infty$ (with different values for α and β) – see appendix for further details.*

Corollary 6 *If $\mathcal{H}$ is the unit ball in $L^\infty(\mathbb{P}; \mathbb{R}^k)$ where $\mathbb{R}^k$ is equipped with the ℓ^p-norm for $1 < p < \infty$, then any minimizer h of $\mathcal{R}$ in $\mathcal{H}$ satisfies that $h(C_i)$ is a single point z_i for all $i = 1, \dots, k$ and the points z_i form the vertices of a regular standard simplex.*

Remark 7 *A major simplification in our analysis was the restriction to one-point classes and general functions on the finite collection of points or more generally to bounded $\mathbb{P}$-measurable functions. In other hypothesis classes, the point values $h(x_i)$ and $h(x_j)$ cannot be chosen independently. It is therefore no longer possible to minimize all terms in the sum individually, and trade-offs are expected. In particular, while our analysis was independent of the weight $p_i = \mathbb{P}(C_i)$ of the individual classes, these are expected to influence trade-offs in real applications.*

Nevertheless, we record that simplex configurations are favored for hypothesis classes $\mathcal{H}$ with the following two properties:

1. *$\mathcal{H}$ is expressive enough to collapse classes to single points and to choose the values on different classes almost independently, and*
2. *functions in $\mathcal{H}$ respect the geometry of $\mathbb{R}^k$ equipped with an ℓ^p-norm in a suitable manner.*

4. Heuristic geometry: penultimate layer

Above, we obtained rigorous results for the final layer geometry under heuristic assumptions. In this section, we consider a hypothesis class $\mathcal{H}$ in which functions can be decomposed as

$$h_{f,A,b}(x) = Af(x) + b \qquad \text{where } f : \mathbb{R}^d \to \mathbb{R}^m, \quad A : \mathbb{R}^m \to \mathbb{R}^k, \quad b \in \mathbb{R}^k$$

and we are interested in the geometry of f and A. Typically, we imagine the case that $m \gg k$.

4.1. Collapse to a point

We have given a heuristic proof above that it is energetically favorable to contract $h(C_i)$ to a single point $z_i \in \mathbb{R}^k$ under certain conditions. Since $A : \mathbb{R}^m \to \mathbb{R}^k$ has a non-trivial kernel for $m > k$, this is a weaker statement than claiming that f maps C_i to a single point $y_i \in \mathbb{R}^m$. We note the following: $V_i = (A \cdot +b)^{-1}(z_i)$ is an $m-k$-dimensional affine subspace of $\mathbb{R}^m$. In particular, a strictly convex norm (e.g. an ℓ^p-norm for $1 < p < \infty$) has a unique minimum $y_i \in V_i$. Thus if we subscribe to the idea that f is constrained by an ℓ^p-norm, it is favorable for f to collapse C_i to a single point $y_i \in \mathbb{R}^m$.

Heuristically, this situation arises either if it is more expensive to increase the norm of f than change its direction, or if (A, b) evolve during training and it is desirable to bring $f(x)$ towards the minimum norm element of $(A \cdot b)^{-1}(z_i)$ to increase the stability of training. The first consideration applies when A, b are fixed while the second relies on the variability of (A, b). Their relative importance could therefore be assessed numerically by initializing the final layer variables in a simplex configuration and making them non-trainable.

If σ is a bounded activation function, the direction of the final layer output depends on the coefficients of all layers in a complicated fashion, while its magnitude mostly depends on the final layer coefficients. We can imagine gradient flows as continuous time versions of the minimizing movements scheme

$$\theta_{n+1} \in \operatorname*{argmin}_{\theta} \frac{1}{2\eta} \|\theta_n - \theta\|^2 + \mathcal{R}\big(h(\theta_n, \cdot)\big)$$

where $h(\theta, \cdot)$ is a parameterized function model. Using the unweighted Euclidean norm for the gradient flow, we allow the same budget to adjust final layer and deep layer coefficients. It may therefore be easier to adjust the direction of the output than the norm. For ReLU activation on the other hand, the magnitude of the coefficients in all layers combines to an output in a multiplicative fashion. It may well be that neural collapse is more likely to occur for activation functions which tend to a finite limit at positive and negative infinity.

In section 5.2, we present examples which demonstrates that if all data points are not collapsed to a single point in the penultimate layer, they may not collapse to a single point in the final layer either when the weights of a neural network are trained by gradient descent. This is established in two different geometries for different activation functions

4.2. Simplex configuration

We showed above that *any* ℓ^p-geometry leads to simplex configurations in the last layer for certain toy models. When considering the geometry of the penultimate layer, we specifically consider ℓ^2-geometry. This is justified for A, b since the parameters are typically initialized according to a normal distribution (which is invariant under general rotations) and optimized by (stochastic)

gradient descent, an algorithm based on the Euclidean inner product. For compatibility purposes, also the output of the preceding layers f should be governed by Euclidean geometry.

Again, as a toy model we consider the case of one-point classes. To simplify the problem, we furthermore suppress the bias vector of the last layer. Let

1. $\mathcal{X} = \{x_1, \dots, x_k\} \subset \mathbb{R}^d$,
2. $f : \mathcal{X} \to B_R(0) \subseteq \mathbb{R}^m$, and
3. $A : \mathbb{R}^m \to \mathbb{R}^k$ linear.

As before $B_R(0)$ denotes the Euclidean ball of radius $R > 0$ centered at the origin in $\mathbb{R}^m$. We denote $h(x) = Af(x)$, $y_i := f(x_i) \in \mathbb{R}^m$ and $z_i := h(x_i) \in \mathbb{R}^k$. As we suppressed the bias of the last layer, we could normalize the center of mass in the penultimate layer to be $\frac{1}{k}\sum_{i=1}^k y_i = 0$. Instead, we make the (weaker) assumption that $y_i \in B_R(0)$ for some $R > 0$ and all $i = 1, \dots, k$.

We assume that the outputs $h(x_i)$ are in the optimal positions in the last layer and show that if A has minimal norm, also the outputs $f(x_i)$ in the penultimate layer are located at the vertices of a regular standard simplex. Denote by

$$\|A\|_{L(\ell^2,\ell^2)} = \max_{\|x\|_{\ell^2}\leq 1} \frac{\|Ax\|_{\ell^2}}{\|x\|_{\ell^2}}$$

the operator norm of the linear map A with respect to the Euclidean norm on both domain and range.

Lemma 8 *Let $m \geq k-1$ and $y_i \in B_R(0) \subseteq \mathbb{R}^m$ and $A : \mathbb{R}^m \to \mathbb{R}^k$ linear such that $Ay_i = z_i$ where z_i are the vertices of the regular standard simplex described in Lemma 2 and (4). Then*

1. *the center of mass of outputs y_i of f is $\frac{1}{k}\sum_{i=1}^k y_i = 0$,*
2. *$\|A\|_{L(\ell^2,\ell^2)} \geq 1$, and*
3. *$\|A\|_{L(\ell^2,\ell^2)} = 1$ if and only if*
   (a) *A is an isometric embedding of the $k-1$-dimensional subspace spanned by $\{y_1, \dots, y_k\}$ into $\mathbb{R}^k$ and*
   (b) *y_i are vertices of a regular standard simplex with the same side lengths.*

The proof is given in the appendix. We conclude the following.

Corollary 9 *For any $m \geq k-1$, consider the hypothesis class*

$$\mathcal{H} = \left\{ h : \mathbb{R}^d \to \mathbb{R}^k \;\middle|\; h = Af \text{ where } \begin{array}{ll} f : \mathbb{R}^d \to \mathbb{R}^m \text{ is } \mathbb{P}-\text{measurable}, & \|f(x)\|_{\ell^2} \leq R \;\; \mathbb{P}-\text{a.e.} \\ A : \mathbb{R}^m \to \mathbb{R}^k \text{ is linear}, & \|A\|_{L(\ell^2,\ell^2)} \leq 1 \end{array} \right\}.$$

Then a minimizer $h \in \mathcal{H}$ of $\mathcal{R}$ satisfies $h = Af$ where

1. *there exist values $y_i \in \mathbb{R}^m$ such that $f(x) = y_i$ for almost every $x \in C_i$,*
2. *the points y_i are located at the vertices of a regular $k-1$-dimensional standard simplex in $\mathbb{R}^m$,*

3. *the center of mass of the points y_i (with respect to the uniform distribution) is at the origin, and*

4. *A is an isometric embedding of the $k-1$-dimensional space spanned by $\{y_1, \dots, y_k\}$ into $\mathbb{R}^k$.*

Remark 10 *The restriction to the Euclidean case is because in Euclidean geometry, any $k-1$-dimensional subspace of $\mathbb{R}^d$ is equipped with the Euclidean norm in a natural way. For other ℓ^p-spaces, the restriction of the ℓ^p-norm is not a norm of ℓ^q-type and we cannot apply Lemma 4.*

Thus, we conclude that a simplex geometry is desirable also in the penultimate layer of a function $h(x) = Af(x)$ if

1. the function class $\mathcal{F}$ in which f is chosen and the linear matrix class in which A is chosen respect the Euclidean geometry of $\mathbb{R}^m$,

2. $\mathcal{F}$ is sufficiently expressive to collapse all data points in the class C_i to a single point y_i and

3. $\mathcal{F}$ is so expressive that y_i and y_j can be chosen mostly independently.

5. Caveats: Binary classification using two-layer neural networks

In this section we consider simple neural network classifier models and data sets on which we can show that the classes *are not* collapsed into single points when the model parameters are trained by gradient descent, despite the fact that the function class is sufficiently expressive. This is intended as a complementary illustration that the heuristic considerations of Sections 3 and 4 may or may not be valid, depending on factors which are yet to be understood.

Deep neural networks with many nonlinearities can be a lot more flexible than shallow neural networks, and the intuition we built up above does not quite apply here. However, we emphasize that a deep neural network h can be decomposed as $h = g \circ f$ where $f : \mathbb{R}^d \to \mathbb{R}^k$ is a deep neural network and $g : \mathbb{R}^k \to \mathbb{R}$ is a shallow neural network. All results should therefore be considered also valid in deep classification models where only the outermost two layers are trained. This is a more realistic assumption in applications where large pretrained models are used to preprocess data and only the final layers are trained for a specific new task. Similarly, we note that this indicates that if data is non-collapsed two layers before the output, then it may not collapse in the output layer either.

The examples we consider concern *binary* classification, i.e. all functions take values in $\mathbb{R}$ rather than a higher-dimensional space. The label function $x \mapsto \xi_x$ takes values in $\{-1, 1\}$ instead of the set of basis vectors. For the sake of convenience, the data below are assumed to be one-dimensional, but similar results are expected to hold when data in a high-dimensional space is either concentrated on a line or classification only depends on the projection to a line.

5.1. Two-layer ReLU-networks in the mean field scaling

Consider the *mean field scaling* of shallow neural networks, where a network function is described as

$$f(x) = \frac{1}{m} \sum_{i=1}^{m} a_i \, \sigma(w_i^T x + b_i) \qquad \text{rather than} \qquad f(x) = \sum_{i=1}^{m} a_i \, \sigma(w_i^T x + b_i).$$

In this regime, it is easy to take the infinite width limit

$$f(x) = \int_{\mathbb{R}^k \times \mathbb{R}^d \times \mathbb{R}} a\, \sigma(w^T x + b)\, \pi(\mathrm{d}a \otimes \mathrm{d}w \otimes \mathrm{d}b) \tag{5}$$

with general weight distributions π on $\mathbb{R}^{k+d+1}$. We denote the functions as represented in (5) by h_π. Finite neural networks are a special case in these considerations with distribution $\pi = \frac{1}{m}\sum_{i=1}^m \delta_{(a_i,w_i,b_i)}$. We recall the following results.

Proposition 11 *Chizat and Bach (2018) All weights (a_i, w_i, b_i) evolve by the gradient flow of*

$$(a_i, w_i, b_i)_{i=1}^m \mapsto \mathcal{R}\left(\frac{1}{m}\sum_{i=1}^m a_i\, \sigma(w_i^T x + b_i)\right)$$

in $(\mathbb{R}^{k+d+1})^m$ if and only if the empirical distribution $\pi = \frac{1}{m}\sum_{i=1}^m \delta_{(a_i,w_i,b_i)}$ evolves by the Wasserstein gradient flow of

$$\pi \mapsto \mathcal{R}\left(h_\pi\right) \tag{6}$$

(up to time rescaling).

Consider specifically $\sigma(z) = \max\{z, 0\}$ and $k = 1$ with the risk functional

$$\mathcal{R}(h) = -\int_{\mathbb{R}^d} \log\left(\frac{\exp(-h(x)\cdot \xi_x)}{\exp(h(x)) + \exp(-h(x))}\right) \mathbb{P}(\mathrm{d}x).$$

The following result applies specifically to the Wasserstein gradient flow of certain continuous distributions, which can be approximated by finite sets of weights.

Proposition 12 *Chizat and Bach (2020) Assume that π^0 is such that $|a|^2 \leq |w|^2 + |b|^2$ almost surely and such that*

$$\pi^0\left(\{(w,b) \in \Theta\}\right) > 0$$

for every open cone Θ in $\mathbb{R}^{d+1}$. Let π^t evolve by the Wasserstein gradient flow of (6) with initial condition π^0. Then (under additional technical conditions), the following hold:

1. *$\xi_x\, h_{\pi^t}(x) \to +\infty$ for $\mathbb{P}$-almost every x.*
2. *There exist*

$$\pi_* \in \operatorname{argmax}\left\{\min_{x \in \operatorname{spt}\mathbb{P}}\left(\xi_x \cdot h_\pi(x)\right) \,\middle|\, \pi \text{ s.t. } \int_{\mathbb{R}^{d+2}} |a|\left[|w| + |b|\right] \mathrm{d}\pi \leq 1\right\} \tag{7}$$

and a normalizing function $\mu : [0,\infty) \to (0,\infty)$ such that $\mu(t)\, h_{\pi_t} \to h_{\pi^*}$ locally uniformly on $\mathbb{R}^d$.

Remark 13 *We call h^* the* maximum margin classifier *in Barron space. Both the normalization condition in (7) and the normalizing function μ are related to the Barron norm or variation norm of classifier functions. The existence of a minimizer in (7) is guaranteed by compactness. Existence of a limit of π^t in some weak sense has to be assumed a priori in Chizat and Bach (2018).*

Remark 14 *The open cone condition is satisfied for example if π_0 is a normal distribution on $\mathbb{R}^{d+1}$, which is a realistic distribution. This property ensures a diversity in the initial distribution, which is required to guarantee convergence. The smallness condition on a is purely technical and required to deal with the non-differentiability of the ReLU activation function, see also Wojtowytsch (2020). The same result holds without modification for leaky-ReLU activation. With some additional modifications, it is assumed to also extend to smooth and bounded activation functions.*

Remark 15 *The divergence $\xi_x\, h_{\pi^t}(x) \to +\infty$ is expected to be logarithmic in time, which can almost be considered bounded in practice. The convergence $h_{\pi^t} \to h^*$ is purely qualitative, without a rate.*

Consider a binary classification problem in $\mathbb{R}$ where $C_{-1} = [-2, -1]$ and $C_1 = [1, 2]$.

Lemma 16 *Consider a binary classification problem in $\mathbb{R}$ where one class C_{-1} with label $\xi = -1$ is contained in $[-2, -1]$ and the other class C_1 with label $\xi = +1$ is contained in $[1, 2]$. Assume that $-1 \in C_{-1}, 1 \in C_1$ and that both classes contain at least one additional point.*

The classification problem admits a continuum of maximum margin classifiers

$$f_b(x) = \frac{1}{2\,[1+b]} \begin{cases} x + b & x > b \\ 2x & -b < x < b \\ x - b & x < -b \end{cases}$$

parametrized by $b \in [0, 1]$.

In particular, we expect that h_{π_t} is not constant on either of the classes $[1, 2]$ or $[-2, -1]$. The proof is postponed until the appendix.

Remark 17 *We described the mean field setting in its natural scaling. However, the same results are true (with a different time rescaling) if f is represented in the usual fashion as $f(x) = \sum_{i=1}^m a_i\, \sigma(w_i^T x + b_i)$ without the normalizing factor $\frac{1}{m}$, assuming that the weights are initialized such that $a_i, w_i, b_i \sim m^{-1/2}$.*

5.2. Two-layer networks with non-convex input classes

Assume that

$$\mathbb{P} = p_1\, \delta_{-1} + p_2\, \delta_0 + p_3\, \delta_1, \qquad p_1, p_2, p_3 \geq 0, \qquad p_1 + p_2 + p_3 = 1$$

and that $\xi_{-1} = \xi_1 = 1$ and $\xi_0 = -1$. We consider the risk functional

$$\mathcal{R}(h) = \int_{\mathbb{R}} \exp\big(-\xi_x h(x)\big)\, \mathbb{P}(\mathrm{d}x) = p_1\, \exp\big(-h(-1)\big) + p_2\, \exp\big(h(0)\big) + p_3\, \exp\big(-h(1)\big),$$

which is similar to cross-entropy loss in its tails since

$$\begin{aligned} -\log\left(\frac{\exp(\xi_x\, h(x))}{\exp(\xi_x\, h(x)) + \exp(-\xi_x\, h(x))}\right) &= -\log\left(\frac{1}{1 + \exp(-2\,\xi_x h(x))}\right) \\ &\approx 1 - \frac{1}{1 + \exp(-2\,\xi_x h(x))} \\ &= \frac{\exp(-2\,\xi_x h(x))}{1 + \exp(-2\,\xi_x h(x))} \\ &\approx \exp(-2\,\xi_x h(x)) \end{aligned}$$

if $\xi_x\, h(x)$ is large. Further assume that the classifier is a shallow neural network with three neurons

$$h(x) = \sum_{i=1}^{3} a_i\, \sigma(w_i x + b_i).$$

To make life easier, we consider a simplified sigmoid activation function $\sigma : \mathbb{R} \to \mathbb{R}$ which satisfies $\sigma(z) = 0$ for $z \leq 0$ and $\sigma(z) = 1$ for $z \geq 1$, and we assume that the parameters (a_i, w_i, b_i) are initialized such that

$$h(x) = a_1\, \sigma(-x) - a_2 \sigma(x+1) + a_3\, \sigma(x) \tag{8}$$

In particular, $\sigma'(w_i x + b_i) = 0$ for $\mathbb{P}$-almost every x at initialization and all $i = 1, 2, 3$. This implies that (w_i, b_i) are constant along gradient descent training, so only a_1, a_2, a_3 evolve. We can write

$$\mathcal{R}\big(-a_1\, \sigma(x) + a_2 \sigma(x+1) - a_3\, \sigma(x-1)\big) = p_1\, \exp(-a_1) + p_2\, \exp(-a_2) + p_3\, \exp(a_2 - a_3).$$

Lemma 18 *Let $h = h_{a_1,a_2,a_3}$ be as in (8) for $a_1, a_2, a_3 \in \mathbb{R}$. Assume that a_1, a_2, a_3 evolve by the gradient flow of $F(a_1, a_2, a_3) = \mathcal{R}(h_{a_1,a_2,a_3})$. Then*

$$\lim_{t\to\infty} \big[h(t,1) - h(t,-1)\big] = 0 \qquad \Leftrightarrow \qquad p_3 = 2p_1$$

independently of the initial condition $(a_1, a_2, a_3)(0)$.

In general, assume that $h = f \circ g$ where f is a shallow neural network. Assume that there are two classes C_i, C_j such that the convex hull of $g(C_i)$ intersects $g(C_j)$. Then it is questionable that classes can collapse to a single point in the final layer. While this does not imply that $g(C_i)$ and $g(C_j)$ should concentrate around the vertices of a regular standard simplex, it suggests that simple geometries are preferred already *before* the penultimate layer if the h is to collapse C_i to a single point.

The proof of Lemma 18 is given in the appendix.

Remark 19 *We note that the probabilities of the different data points crucially enter the analysis, while considerations above in Lemma 4 were entirely independent of the weight of different classes. The toy model does not capture interactions between the function values at different data points, which is precisely what drives the dynamics here.*

References

Lenaic Chizat and Francis Bach. On the global convergence of gradient descent for over-parameterized models using optimal transport. In *Advances in neural information processing systems*, pages 3036–3046, 2018.

Lenaic Chizat and Francis Bach. Implicit bias of gradient descent for wide two-layer neural networks trained with the logistic loss. *arxiv:2002.04486 [math.OC]*, 2020.

Yaim Cooper. The loss landscape of overparameterized neural networks. *arXiv:1804.10200 [cs.LG]*, 2018.

Simon S Du, Jason D Lee, Haochuan Li, Liwei Wang, and Xiyu Zhai. Gradient descent finds global minima of deep neural networks. *arXiv:1811.03804 [cs.LG]*, 2018a.

Simon S Du, Xiyu Zhai, Barnabas Poczos, and Aarti Singh. Gradient descent provably optimizes over-parameterized neural networks. *arXiv:1810.02054 [cs.LG]*, 2018b.

Weinan E, Chao Ma, and Lei Wu. A comparative analysis of optimization and generalization properties of two-layer neural network and random feature models under gradient descent dynamics. *Sci. China Math.*, https://doi.org/10.1007/s11425-019-1628-5, 2020.

Michael Elad, Dror Simon, and Aviad Aberdam. Another step toward demystifying deep neural networks. *Proceedings of the National Academy of Sciences*, 117(44):27070–27072, 2020.

Jianfeng Lu and Stefan Steinerberger. Neural collapse with cross-entropy loss. *arxiv: 2012.08465 [cs.LG]*, 2020.

Dustin G Mixon, Hans Parshall, and Jianzong Pi. Neural collapse with unconstrained features. *arXiv:2011.11619 [cs.LG]*, 2020.

Vardan Papyan, XY Han, and David L Donoho. Prevalence of neural collapse during the terminal phase of deep learning training. *Proceedings of the National Academy of Sciences*, 117(40): 24652–24663, 2020.

Stephan Wojtowytsch. On the global convergence of gradient descent training for two-layer Relu networks in the mean field regime. *arXiv:2005.13530 [math.AP]*, 2020.