The Unexpected Deterministic and Universal Behavior of Large Softmax Classifiers

Anonymous Author(s)
Affiliation
Address
`email`

Abstract

In this paper, we provide a large dimensional analysis of the Softmax classifier, extensively used in modern neural networks. We discover and prove that, when the classifier is trained on data satisfying reasonable concentration assumptions, its weights become deterministic and solely depend on the statistical means and covariances of the data. As a striking consequence, despite the implicit and non-linear nature of the underlying optimization problem, the performance of the Softmax classifier is the same as if performed on a mere Gaussian mixture model, thereby disrupting the intuition that non-linearities inherently extract advanced statistical features from the data. Our findings are in particular theoretically sustained as well as numerically confirmed on CNN representations of images produced by GANs.

1 Introduction

The intricate nature of deep neural network training leaves little insight on the specific information encoded into the inter-layer connectivity weights of a fully trained network, thereby so far not allowing for particularly useful interpretation and control of their performances [YKYR18].

At the very source of these difficulties are the multiple non-linearities and the implicit optimization scheme involved in the network design: the activation functions in the intermediate layers as well as the soft or hard final decision layer [LWL+17]. For lack of a tractable comprehensive analysis, literature studies have mostly focused on individual components which, when isolated, become tractable. For instance, the effect of non-linearities in a single-hidden layer network was analysed in [PW17, LLC+18], the learning dynamics in elementary network designs in [SMG13, dCPS+18] and the overall understanding of the geometry of the loss surface in a largely approximated version of a deep neural net in [PB17, CHM+15].

These works are however restricted to the analysis either of the intermediate layers of practical neural nets, or oversimplify the network to an extent that makes the results rather impractical. The present article instead focuses on the training of the weights of the last decision layer, by specifically studying the widely used Softmax component in neural networks classifiers. The Softmax classifier has the property, which we will see to be of importance here, to be optimal for Gaussian mixture inputs with equal covariance [YW19]. Specifically, assuming the feature representations of the data fixed at the penultimate layer of the network, and modelling these features as *concentrated random vectors* [Led05] (which is a natural assumption as concentrated random vectors enjoy the property to be stable through Lipschitz maps, and thus through the action of intermediate neural network layers [SLTC20]), the article studies the statistical behavior of this last layer once trained (see Figure 1).

Our analysis leverages recent advances in random matrix theory by supposing the realistic setting where the number of data samples n (here their representations at the penultimate layer) and their

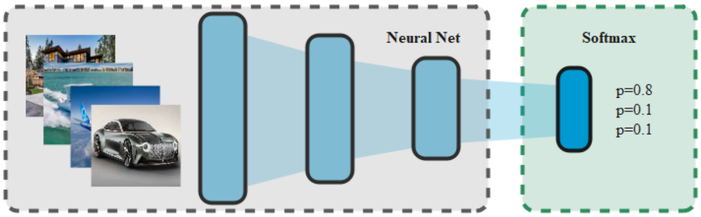

Figure 1: A neural network with fixed representations in the penultimate layer (in gray) and a trained Softmax output (in green).

dimension p (the size of this representation, i.e., the number of neurons in the one-before-last layer) are both large and comparable.

From a technical standpoint, as the Softmax classifier training corresponds to a (possibly non-convex) optimization problem, our analysis of the Softmax weights is performed by first expressing the optimization problem as a contracting fixed point equation, and then showing that the assumed *concentration properties* of the data features naturally transfer to the solution of the fixed-point equation, and thus to the Softmax weights. This has the major consequence that, as $n, p \to \infty$, the Softmax weights tend to have a deterministic behavior which we express explicitly as a function of the data statistics and the Softmax parameters.

Our most fundamental findings may be summarized as follows:
1. the above deterministic behavior exhibits a surprising *universality* of the Softmax classifier, in the sense that the large dimensional statistics of the weights solely depend on the statistical means and covariances of the input data features;
2. this suggests in turn that, quite counter-intuitively, at least as far as the last Softmax classification layer is concerned, no further discriminative feature of the data is extracted and, possibly most outstandingly, *the Softmax layer treats the input data as if they were Gaussian random vectors*; this, in passing, supports the Gaussianity assumption on the data representations commonly considered in the literature [HRU$^+$17, PRU$^+$18];
3. combined to the aforementioned optimality of the Softmax classifier on Gaussian mixture models with strongly discriminative class-wise means, this compellingly supports an overall classification optimality of the Softmax classifier on large dimensional representations of real data. A similar behavior was already pointed out, yet not well understood, by the authors in [MVPC13, GCM18];
4. Our findings are supported both theoretically and practically by considering the input data features as CNN-representations of images generated by the BigGAN model [BDS18].

In the remainder of the article, we introduce more precisely the present model of Softmax classification training and recall some concentration of measure tools necessary for best understanding (Section 2), before stating our main theoretical results (Section 3) along with supporting experiments and further concluding remarks (Section 4). The detailed derivations of our results are deferred to Section 5 and the Supplementary Material.

Notation: For $m \in \mathbb{N}$, $[m] \equiv \{1, \ldots, m\}$. Vectors are denoted by boldface lowercase and matrices by boldface uppercase letters. The set of matrices of size $p \times n$ is denoted $\mathcal{M}_{p,n}$, the set of squared matrices and diagonal matrices of size n respectively $\mathcal{M}_n$ and $\mathcal{D}_n$. $\|\cdot\|$ is the Euclidean (resp., spectral) norm for vectors (resp., matrices with $\|\cdot\| : \boldsymbol{M} \mapsto \sup_{\|\boldsymbol{u}\| \leq 1} \|\boldsymbol{M}\boldsymbol{u}\|$); $\|\cdot\|_F$ stands for the Frobenius norm $\|\cdot\|_F : \boldsymbol{M} \mapsto \sqrt{\operatorname{Tr}(\boldsymbol{M}\boldsymbol{M}^T)}$ and $\|\cdot\|_*$ stands for the nuclear norm $\|\cdot\|_* : \boldsymbol{M} \mapsto \operatorname{Tr}(\sqrt{\boldsymbol{M}\boldsymbol{M}^T})$ (which is the dual norm of the spectral norm for the scalar product $\boldsymbol{A}, \boldsymbol{B} \to \operatorname{Tr}(\boldsymbol{A}^\intercal \boldsymbol{B})$). $\otimes$ stands for the Kronecker product.

2 Model setting

2.1 The Softmax classifier

Let $(\boldsymbol{x}_1, \boldsymbol{y}_1), \ldots, (\boldsymbol{x}_n, \boldsymbol{y}_n)$ be a set of n labeled data associated to one of k classes $\mathcal{C}_1, \ldots, \mathcal{C}_k$, where $\boldsymbol{x}_i \in \mathbb{R}^p$ and $\boldsymbol{y}_i \in \mathbb{R}^k$ is one-hot encoded vectors such that $y_{i\ell} = 1$ if $\boldsymbol{x}_i \in \mathcal{C}_\ell$. The $\boldsymbol{x}_i$'s are assumed to be the input of an ℓ_2-regularized Softmax classifier with regularization parameters $(\lambda_\ell)_{\ell\in[k]} \in \mathbb{R}^+$, which aims to determine the class-wise weight vectors $\boldsymbol{w}_1, \dots, \boldsymbol{w}_k \in \mathbb{R}^p$ minimizing the loss[1], for some real-valued function $\phi : \mathbb{R} \to \mathbb{R}$:

$$\mathcal{L}(\boldsymbol{w}_1, \dots, \boldsymbol{w}_k) = -\frac{1}{n}\sum_{i=1}^{n}\sum_{\ell=1}^{k} y_{i\ell}\log p_{i\ell} + \frac{1}{2}\sum_{\ell=1}^{k}\lambda_\ell\|\boldsymbol{w}_\ell\|^2 \quad \text{with} \quad p_{i\ell} = \frac{\phi(\boldsymbol{w}_\ell^\intercal \boldsymbol{x}_i)}{\sum_{j=1}^{k}\phi(\boldsymbol{w}_j^\intercal \boldsymbol{x}_i)}$$

In particular, the classical Softmax classifier corresponds to the case where $\phi(t) = e^t$ [GP17]. Cancelling the loss function gradient with respect to each weight vector $\boldsymbol{w}_\ell$ yields

$$\lambda_\ell \boldsymbol{w}_\ell = -\frac{1}{n}\sum_{i=1}^{n}\left(y_{i\ell}\psi(\boldsymbol{w}_\ell^\intercal \boldsymbol{x}_i) - \frac{\phi(\boldsymbol{w}_\ell^\intercal \boldsymbol{x}_i)}{\sum_{j=1}^{k}\phi(\boldsymbol{w}_j^\intercal \boldsymbol{x}_i)}\sum_{j=1}^{k} y_{ij}\psi(\boldsymbol{w}_j^\intercal \boldsymbol{x}_i)\right)\boldsymbol{x}_i, \quad \ell \in [k], \tag{1}$$

where $\psi \equiv \phi'/\phi$. Under appropriate statistical assumptions on the data matrix $\boldsymbol{X} \equiv [\boldsymbol{x}_1, \dots, \boldsymbol{x}_n] \in \mathcal{M}_{p,n}$ and on ψ, and assuming p, n large, we subsequently show that the vector $\boldsymbol{W} \equiv [\boldsymbol{w}_1^\intercal, \dots, \boldsymbol{w}_k^\intercal]^\intercal \in \mathbb{R}^{pk}$ has a well defined behavior, which in turn allows us to accurately predict the performances of the Softmax classifier.

2.2 Mixture of concentrated random vectors

We first characterize the data classes: if $y_{i\ell} = 1$, then $\boldsymbol{x}_i \in \mathbb{R}^p$ is a random vector with

$$\mathbb{E}[\boldsymbol{x}_i] \equiv \boldsymbol{\mu}_\ell, \quad \mathbb{E}_{\boldsymbol{x}_i}[\boldsymbol{x}_i\boldsymbol{x}_i^\intercal] - \boldsymbol{\mu}_\ell\boldsymbol{\mu}_\ell^\intercal \equiv \boldsymbol{\Sigma}_\ell.$$

The vectors $\boldsymbol{x}_1, \dots, \boldsymbol{x}_n$ are further assumed to be independent and are such that $\boldsymbol{X} = [\boldsymbol{x}_1, \dots, \boldsymbol{x}_n] \in \mathcal{M}_{p,n}$ satisfies a *concentration* property. To properly state this assumption (Assumption 1), which is central to our analysis, we first define the notion of random vector concentration.

Definition 1 (Concentrated vector)**.** *Given a set of indices $\mathbb{S}$, a sequence of normed vector spaces $(E_s, \|\cdot\|_s)_{s\in\mathbb{S}}$, a sequence of random vectors $\boldsymbol{Z}_s \in E_s$, a sequence of positive numbers σ_s, we say that $\boldsymbol{Z}_s$ is q-*exponentially concentrated with an observable diameter of order $O(\sigma_s)$ *if there exists two constants $C, c > 0$ such that for all sequence of 1-Lipschitz mappings $f_s : E_s \to \mathbb{R}$:*

$$\forall s \in S, \forall t > 0 \, : \, \mathbb{P}\left(|f_s(\boldsymbol{Z}_s) - \mathbb{E}[f_s(\boldsymbol{Z}_s)]| \geq t\right) \leq Ce^{-c(t/\sigma_s)^q}. \tag{2}$$

We note then $\boldsymbol{Z}_s \propto \mathcal{E}_q(\sigma_s)$, or simply $\boldsymbol{Z} \propto \mathcal{E}_q(\sigma)$; when $\sigma_s = O(1)$, we write $\boldsymbol{Z}_s \propto \mathcal{E}_q$.

The prototypical example of a concentrated random vector is the Gaussian random vector $\boldsymbol{Z}_s \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_p)$ for which $\boldsymbol{Z}_s \propto \mathcal{E}_2$ ([Led05]). But the richness of concentrated random vectors lies in their fundamental stability property through Lipschitz operations, which naturally generates wide families of concentrated random vectors.

Remark 2.1 (Stability through Lipschitz transformations)**.** *It is easily deduced from Definition 1 that given a sequence of positive numbers $L_s > 0$ and a sequence of L_s-Lipschitz transformations $\phi_s : (E_s, \|\cdot\|_s) \to (F_s, \|\cdot\|_s')$, if $\boldsymbol{Z}_s \propto \mathcal{E}_q(\sigma_s)$, then $\phi_s(\boldsymbol{Z}_s) \propto \mathcal{E}_q(L_s\sigma_s)$ (indeed, for all $f_s : F_s \to \mathbb{R}$, 1-Lipschitz, $\frac{1}{L_s}f_s \circ F_s$ is 1-Lipschitz, and one can employ inequality 2 to $\frac{t}{L_s}$).*

In particular, the concentration of Gaussian vectors combined with the stability through Lipschitz transformations as per Remark 2.1 provides a wide range of random vectors, among which random vectors with possibly quite complex dependence structures between their entries. A remarkable example of such random vectors are random vectors produced by generative adversarial networks (GANs) [GPAM+14]: GAN random vectors notably satisfy that their outputs has the same concentration[2] as their inputs [SLTC20]; in particular, for Gaussian $\mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_m)$ inputs (as traditionally assumed) whose observable diameter does not depend on the dimension m, the observable diameter of the GAN outputs does not increase with the data dimension. Besides, further operations through

[1] Biases are not introduced in the present formulation as their effect is known to be negligible in practice [KXR+19] and would only decrease the readability and accessibility of our results.

[2] When the GAN model has a controlled Lipschitz constant, which is practically ensured by Spectral Normalization as in the BigGAN model [BDS18].

neural network layers with controlled Lipschitz norms (as is again traditionally done) on concentrated random vectors also maintain the concentration and observable diameter.

As a consequence of the above remark, making the approximation that GAN-generated data are alike real data, we may assume that GAN data fed into the first layer of a deep neural network are output in the one-before-last layer as a concentrated random vector with observable diameter independent of its dimension. This is summarized into our present Softmax input data assumption:

Assumption 1 (Concentrated data)**.** *Letting* $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n]$*,* $\boldsymbol{X} \propto \mathcal{E}_2$*.*

In the terms of Definition 1, Assumption 1 holds here for $s = (p, n)$ with $\mathbb{S} = \mathbb{N}^2$. In order to be able to transfer the concentration of $\boldsymbol{X}$ to the Softmax weights $\boldsymbol{w}_1, \ldots, \boldsymbol{w}_k$, a further condition is needed: the number of data n must scale with the data dimension p, i.e., $\mathbb{S} = \{(p, n) \in \mathbb{N}^2, \kappa p \leq n \leq Kp\}$ for some $K > \kappa > 0$.[3] This is summarized by the request:

Assumption 2 (Growth rate)**.** $n = O(p)$ *and* $p = O(n)$*.*

Concentrated vectors satisfy a host of interesting properties (the reader being referred to [Led05] for a detailed account and to [LLC$^+$18] for their application to random matrix asymptotics, closer to the present work). We merely stress here one of these properties, of central importance to the present work, and which fundamentally justifies the appearance of Gaussian-like behaviors in large neural networks, even when the neural network input is far from Gaussian [KGC18, NBA$^+$18].

Theorem 2.2 (CLT for concentrated vectors [Kla07, FGP07])**.** *Let* $\boldsymbol{X} \in \mathbb{R}^p$ *be a random vector with* $\mathbb{E}[\boldsymbol{X}] = 0$ *and* $\mathbb{E}[\boldsymbol{X}\boldsymbol{X}^\intercal] = I_p$*, and* σ *be the uniform measure on the sphere* $\mathcal{S}^{p-1} \subset \mathbb{R}^p$ *of radius* 1*. Then, if* $\boldsymbol{X} \propto \mathcal{E}_2$*, there exists two constants* $C, c > 0$ *and a set* $\Theta \subset \mathcal{S}^{p-1}$ *such that* $\sigma(\Theta) \geq 1 - \sqrt{p}Ce^{-c\sqrt{p}}$ *and* $\forall \boldsymbol{\theta} \in \Theta$*:*

$$\sup_{t \in \mathbb{R}} |\mathbb{P}(\boldsymbol{\theta}^\intercal \boldsymbol{X} \geq t) - G(t)| \leq p^{-1/4}$$

for G *the cumulative distribution function of an* $\mathcal{N}(0, 1)$ *random variable.*

3 Main results

3.1 Behavior of the Softmax classifier weights and performance estimation

This section characterizes the statistical behavior of the Softmax classifier weights $\boldsymbol{W} = [\boldsymbol{w}_1^\intercal, \ldots, \boldsymbol{w}_k^\intercal]^\intercal \in \mathbb{R}^{pk}$ under Assumptions 1–2 and, as a result, accesses the asymptotic performances of the classifier.

To this end, our approach is to first write the implicit defining equation 1 of $\boldsymbol{W}$ under the formal form $\boldsymbol{W} = \Psi(\boldsymbol{W})$, for $\Psi : \mathbb{R}^{pk} \to \mathbb{R}^{pk}$ to characterize, and then to *transfer* the concentration of $\boldsymbol{X}$ (Assumption 1) to a concentration of $\boldsymbol{W}$.

For the concentration of $\boldsymbol{X}$ to propagate into $\boldsymbol{W}$ defined through the formal form $\boldsymbol{W} = \Psi(\boldsymbol{W})$, Ψ is required to have contracting properties, which in turn will enforce structural conditions on the operator ϕ and on the regularizers $(\lambda_\ell)_{\ell \in [k]}$. Specifically Ψ is requested to be $(1 - \varepsilon)$-Lipschitz (for some $\varepsilon > 0$) so to ensure, thanks to the Banach fixed point theorem, the existence and uniqueness of $\boldsymbol{W} \in \mathbb{R}^{pk}$. However, being a *random map* depending on $\boldsymbol{X}$, Ψ is only contracting under the (asymptotically highly probable) event $\mathcal{A}_X$ (see Section 2 of the Supplementary Material) that the norm of $\boldsymbol{X}$ is not too large. With these informal steps in mind, we are in position to state our main results.

Theorem 3.1 (Concentration of $\boldsymbol{W}$)**.** *Under Assumptions 1 and 2 and additional assumptions on* ϕ *and* $(\lambda_\ell)_{\ell \in [k]}$ *provided in Section 2 of the Supplementary Material (Assumptions 3, 4 and 5), there exist two constants* $C, c > 0$ *and an event* $\mathcal{A}_X$ *with* $\mathbb{P}(\mathcal{A}_X) > 1 - Ce^{-cn}$ *such that*[4]

$$(\boldsymbol{W} \mid \mathcal{A}_X) \propto \mathcal{E}_2\left(\sqrt{\log n / n}\right).$$

[3] Formally, in the present setting, it is sufficient that $p \leq \frac{1}{\kappa} n$. However, to obtain simpler expressions, it is convenient to assume, in addition, that $n \leq Kp$.

[4] Formally, the random vector $\boldsymbol{W}$ is a measurable mapping $\Omega \to \mathbb{R}^{pk}$, where $(\Omega, \mathcal{F}, \mathbb{P})$ is the underlying probability space. If $\mathbb{P}(\mathcal{A}) > 0$, for $\mathcal{A} \in \mathcal{F}$, the random vector $(\boldsymbol{W} \mid \mathcal{A})$ is the measurable mapping $\mathcal{A} \to \mathbb{R}^{pk}$ such that, $\forall \omega \in \mathcal{A}$, $(\boldsymbol{W} \mid \mathcal{A})(\omega) = \boldsymbol{W}(\omega)$. The statistics of $(\boldsymbol{W} \mid \mathcal{A})$ are then computed in the probability space $(\mathcal{A}, \mathcal{F} \wedge \mathcal{A}, \mathbb{P}_\mathcal{A})$, where $\mathcal{F} \wedge \mathcal{A} = \{B \cap \mathcal{A}, B \in \mathcal{F}\}$ and $\forall B \in \mathcal{F}$, $\mathbb{P}_\mathcal{A}(B) = \mathbb{P}(B)/\mathbb{P}(\mathcal{A})$.

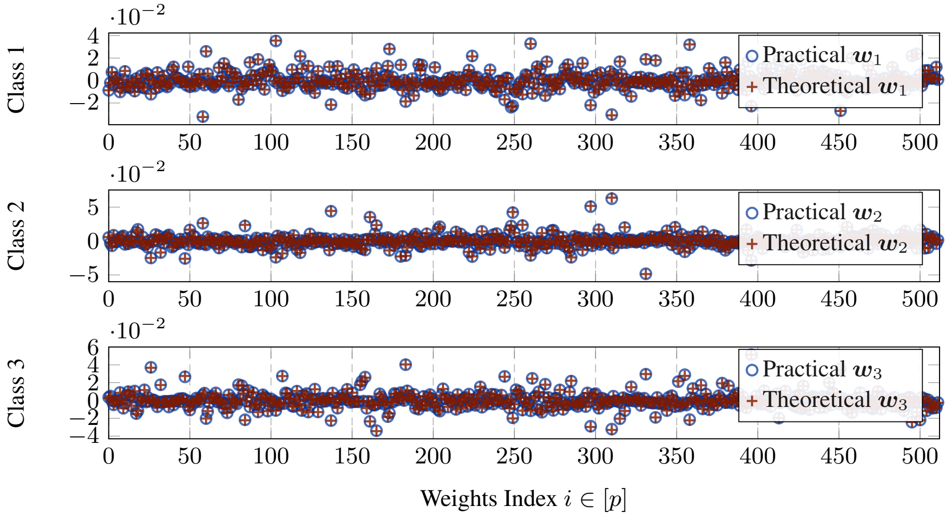

Figure 2: Learned weights (in blue circles) versus their theoretical estimations (in red crosses) as per Theorem 3.2. The used data are Resnet15 [SIVA17] representations ($p = 512$) of images generated by the BigGAN model [BDS18] which are concentrated vectors [SLTC20]. We considered $k = 3$ classes which are: *hamburger*, *mushroom*, *pizza*. $n = 3000$ and the regularization constants $\lambda_1, \lambda_2, \lambda_3 = 1.5$. The data are normalized such that their norm is $0.1 \cdot \sqrt{p}$ to ensure $\mathcal{A}_X$.

Since their observable diameter ($\sqrt{\log n/n}$) vanishes at large n, it therefore entails from Theorem 3.1 that *the random weights vector $\boldsymbol{W}$ tend to be deterministic as p, n grow large*. The subsequent result further characterizes its first and second order statistics at large p, n.

Theorem 3.2 (Asymptotic statistics of $\boldsymbol{W}$)**.** *Define the statistics*

$$\boldsymbol{m}_W \equiv \mathbb{E}[\boldsymbol{W}], \quad \boldsymbol{C}_W \equiv [\boldsymbol{W}\boldsymbol{W}^\intercal] - \boldsymbol{m}_W\boldsymbol{m}_W^\intercal.$$

Then, under Assumptions 1 and 2 and additional assumptions on ϕ and $(\lambda_\ell)_{\ell\in[k]}$ provided in Section 2 of the Supplementary Material (Assumptions 3, 4 and 5), there exists a deterministic mapping $\mathcal{F}_{\boldsymbol{\mu},\boldsymbol{\Sigma}} = \mathcal{F}_{\boldsymbol{\mu},\boldsymbol{\Sigma}}(\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_1, \ldots, \boldsymbol{\Sigma}_k) : \mathbb{R}^{pk} \times \mathcal{M}_{pk} \longrightarrow \mathbb{R}^{pk} \times \mathcal{M}_{pk}$ depending only on the statistics $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_1, \ldots, \boldsymbol{\Sigma}_k$ of $\boldsymbol{X}$, such that the equation

$$(\boldsymbol{m}, \boldsymbol{C}) = \mathcal{F}_{\boldsymbol{\mu},\boldsymbol{\Sigma}}(\boldsymbol{m}, \boldsymbol{C}) \quad \textit{with} \quad \boldsymbol{m} \in \mathbb{R}^{pk}, \ \boldsymbol{C} \in \mathcal{M}_{pk}$$

admits a unique solution $(\bar{\boldsymbol{m}}_W, \bar{\boldsymbol{C}}_W)$. Besides,

$$\|\bar{\boldsymbol{m}}_W - \boldsymbol{m}_W\| \leq O\left(\sqrt{\log n/n}\right) \quad \textit{and} \quad \|\bar{\boldsymbol{C}}_W - \boldsymbol{C}_W\|_* \leq O\left(\sqrt{\log n/n}\right).$$

The exact expression of $\mathcal{F}_{\boldsymbol{\mu},\boldsymbol{\Sigma}}$, explicitly given in (8), is rather elaborate and of limited interest at this point. Section 5 provides more extensive details.

The central outcome of Theorem 3.2 is that, under the data concentration Assumption 1, the behavior of the Softmax classifier *only depends on the class-wise means and covariances of the input data*. This arises as a direct consequence of the Lipschitz character of the Softmax classifier which preserves concentration (by the stability result of Remark 2.1), and of the presence of a *projection of the parameter vectors $\boldsymbol{w}_\ell$ onto the concentrated data $\boldsymbol{x}_i$* at the core of the optimization formulation: according to Theorem 2.2, these projections induce an asymptotic Gaussian behavior with mean and variance depending *only* on the first statistics of the data and the weights vector $\boldsymbol{W}$.

Once the Softmax classifier is trained, the probability for a new datum $\boldsymbol{x}$ to belong to class $\ell \in [k]$ is explicitly given by $p_\ell(\boldsymbol{x}) = \phi(\boldsymbol{w}_\ell^\intercal \boldsymbol{x}) / \sum_{j\in[k]} \phi(\boldsymbol{w}_j^\intercal \boldsymbol{x})$. As a consequence of Theorem 2.2, $\boldsymbol{w}_\ell^\intercal \boldsymbol{x}$ has a high probability to be Gaussian (since $\boldsymbol{x}$ is concentrated and $\boldsymbol{w}_\ell$ has a deterministic behavior). The performances of the Softmax classifier are therefore theoretically tractable.

Corollary 3.3 (Generalization performance of the Softmax classifier)**.** *For $\ell \in [k]$, there exists $\bar{\kappa}^\ell \in \mathbb{R}^{k-1}$ and $\bar{\boldsymbol{K}}^\ell \in \mathcal{M}_{k-1}$ both depending only on $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_1, \ldots, \boldsymbol{\Sigma}_k$ such that the*

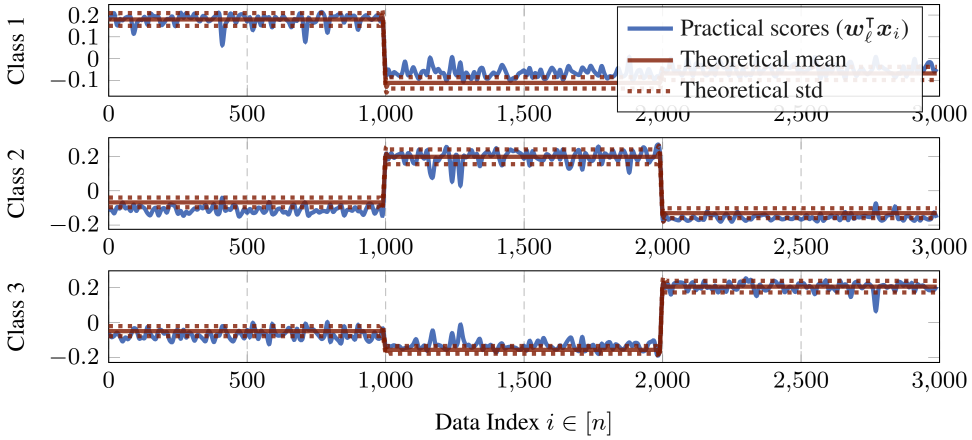

Figure 3: Scores (in blue) versus their theoretical estimations (in red) as per Corollary 3.3, with the theoretical means (through $\bar{\kappa}^\ell$) and standard deviations (through $\bar{\boldsymbol{K}}^\ell$), on a test set independent from the training set. The used data are Resnet15 [SIVA17] representations ($p = 512$) of images generated by the BigGAN model [BDS18] which are concentrated vectors [SLTC20]. We considered $k = 3$ classes which are: *hamburger*, *mushroom*, *pizza*. $n = 3000$ and the regularization constants $\lambda_1, \lambda_2, \lambda_3 = 1.5$. The data are normalized such that their norm is $0.1 \cdot \sqrt{p}$ to ensure $\mathcal{A}_X$.

asymptotic misclassification error $E_t(\boldsymbol{x} \in \mathcal{C}_\ell)$ *of a new datum* $\boldsymbol{x}$ *belonging to class* $\ell \in [k]$ *defined as* $E_t(\boldsymbol{x} \in \mathcal{C}_\ell) \equiv 1 - \mathbb{P}(\forall j \in [k] \setminus \{\ell\} : p_\ell(\boldsymbol{x}) \geq p_j(\boldsymbol{x}))$ *is*

$$E_t(\boldsymbol{x} \in \mathcal{C}_\ell) = 1 - \mathbb{P}(\boldsymbol{Z}_\ell \in \mathbb{R}_+^{k-1}) \quad \textit{with} \quad \boldsymbol{Z}_\ell \sim \mathcal{N}(\bar{\kappa}^\ell, \bar{\boldsymbol{K}}^\ell). \tag{3}$$

In essence, Corollary 3.3 states that the generalization performance of the Softmax classifier reduces to the cumulative distribution of a low-dimensional Gaussian vector, the mean and covariance of which only depend on the class-wise means and covariances of the input data. This demonstrates the remarkable *universality* property of the Softmax classifier with respect to the data distribution, which we recall is only requested to satisfy a very loose concentration behavior (Assumption 1). The exact expressions of $\bar{\kappa}^\ell$ and $\bar{\boldsymbol{K}}^\ell$, along with a justification of the corollary, are provided in Sections 3–4 of the Supplementary Material.

3.2 Experimental validation

This section provides an experimental setup to support our theoretical findings. The input data $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n]$ are independent Resnet15[5] representations of size $p = 512$ [SIVA17] of images generated by the BigGAN generative adversarial network model [BDS18]: as such, being the composition of two neural networks (BigGAN and Resnet15) applied to random standard Gaussian noise (as per the BigGAN model), $\boldsymbol{X}$ is concentrated by construction and satisfies Assumption 1, as requested (see [SLTC20] for a detailed analysis of the Lipschitz properties of these networks). Under this setting, Figure 2 depicts the learned Softmax weights against their expected large n, p asymptotics as per Theorem 3.2. Despite the finite p, n setting of the simulation, a perfect match is observed between the learned weights and the theoretical predictions. In Section 5 of the Supplementary Material, further experiments are performed on *real images* from the ImageNet dataset [DDS+09], which again show a perfect match between theory and practice, thereby strongly suggesting that the conclusions of Theorem 3.2 extend to real data.

Figure 3 next displays the class-wise scores of a practical Softmax classifier on an independent test set against their estimated statistics according to Corollary 3.3. An almost perfect match is again observed between empirical values and theoretical statistics. Section 5 of the Supplementary Material reports similar outputs for real ImageNet data. We importantly stress that, as per Corollary 3.3, the theoretical estimates were obtained using *only the empirical class-wise means and covariances* of the input data. Figure 3 thus confirms the theoretically predicted universality of the Softmax classifier. A Python implementation is attached for reproductivity of these experiments.

[5] We used its Pytorch implementation [PGM+19] pre-trained on the Imagenet dataset [DDS+09].

4 Concluding Remarks

As a consequence of Corollary 3.3, we have demonstrated that, even though the Softmax classifier has a non-linear nature, a property supposedly useful to extract “deep” non-linear features, for n, p rather large, the input data are in fact treated as if they were distributed as a mere Gaussian mixture model. This large dimensional universality phenomenon fundamentally revisits the conventional insights acquired along the years on non-linear classification methods.

As an aftermath, the Softmax classifier being optimal for Gaussian mixture inputs with common covariance, our study is strongly suggestive of the optimality of Softmax as the last layer of a deep neural network classifier.

Yet, the present study assumes a clear-cut separation between a back-end network training isolated from the front-end Softmax layer (as depicted in Figure 1). A thorough validation of the equivalence between full network training and this divided approach is a necessary final step to confirm the claimed optimality and anticipate the performances of Softmax classification.

5 Proof of the Main Results

We now provide the main ingredients to obtain the result of Theorem 3.2, which mainly unfolds from two essential steps: (i) the control of the statistical dependencies between $\boldsymbol{W}$ and $\boldsymbol{X}$, presented in Subsection 5.1, and (ii) the estimation of the statistics $\boldsymbol{m}_W$ and $\boldsymbol{C}_W$ of the weight vector $\boldsymbol{W}$ (in Subsection 5.2). We start by reformulating (1) in the compact and convenient form

$$\boldsymbol{\Lambda} \boldsymbol{W} = \frac{1}{n} \sum_{i=1}^{n} \tilde{\boldsymbol{x}}_i f_i(\tilde{\boldsymbol{x}}_i^\intercal \boldsymbol{W}) \tag{4}$$

where $\tilde{\boldsymbol{X}} = [\tilde{\boldsymbol{x}}_1, \ldots, \tilde{\boldsymbol{x}}_n] \in \mathcal{M}_{kp,kn}$, $f(\tilde{\boldsymbol{X}}^\intercal \boldsymbol{W}) \in \mathbb{R}^{kn}$ and $\boldsymbol{\Lambda} = \mathrm{Diag}(\lambda_1, \ldots, \lambda_k) \otimes \boldsymbol{I}_p \in \mathcal{M}_{kp}$. For lack of space, the expressions of $\tilde{\boldsymbol{x}}_i \in \mathcal{M}_{pk,k}$ (which relates directly to $\boldsymbol{x}_i$), and of the functions $(f_i)_{i \in [n]}$ are deferred to Section 1 of the Supplementary Material.

5.1 Control of the dependencies

Applying the expectation operator both sides to (4), the main technical difficulty arises from the evaluation of $\mathbb{E}[\tilde{\boldsymbol{x}}_i f_i(\tilde{\boldsymbol{x}}_i^\intercal \boldsymbol{W})]$ due to the elaborate dependencies between the weight vector $\boldsymbol{W}$ and the data $\tilde{\boldsymbol{x}}_i$. Note that $\tilde{\boldsymbol{x}}_i^\intercal \boldsymbol{W}$ *a priori* has no reason of being Gaussian (even in the limit) and the performances of the Softmax classifier may depend on high order statistics of $\boldsymbol{X}$. These statistical dependencies are dealt with by introducing a mapping $\boldsymbol{W}_{-i} : [0, 1] \to \mathbb{R}^{pk}$, defined for $i \in [n]$, as the unique solution to:

$$\forall t \in [0, 1] \, : \, \boldsymbol{\Lambda} \boldsymbol{W}_{-i}(t) = \frac{1}{n} \sum_{j \neq i} \tilde{\boldsymbol{x}}_j f_j(\tilde{\boldsymbol{x}}_j^\intercal \boldsymbol{W}_{-i}(t)) + \frac{1}{n} t \tilde{\boldsymbol{x}}_i f_i \left(\tilde{\boldsymbol{x}}_i^\intercal \boldsymbol{W}_{-i}(t)\right). \tag{5}$$

This mapping can be seen as a path between the weights vector $\boldsymbol{W} = \boldsymbol{W}_{-i}(1)$ of the Softmax classifier and $\boldsymbol{W}_{-i}(0)$ which is completely independent of $\tilde{\boldsymbol{x}}_i$ and which will be simply denoted $\boldsymbol{W}_{-i}$. Using the inverse function theorem, the mapping $t \mapsto \boldsymbol{W}_{-i}(t)$ is differentiable, we then deduce the following central close form formula:

$$\boldsymbol{W}'_{-i}(t) = \frac{1}{n} \boldsymbol{Q}_{-i}(t) \tilde{\boldsymbol{x}}_i \chi'_i(t) \quad \text{with} \quad \boldsymbol{Q}_{-i}(t) \equiv \left(\boldsymbol{\Lambda} - \frac{1}{n} \tilde{\boldsymbol{X}}_{-i} \boldsymbol{D}^{(i)}(t) \tilde{\boldsymbol{X}}_{-i}^\intercal \right)^{-1} \in \mathcal{M}_{kp}, \tag{6}$$

where $\chi_i(t) \equiv t f_i \left(\tilde{\boldsymbol{x}}_i^\intercal \boldsymbol{W}_{-i}(t))\right)$, $\boldsymbol{D}_j^{(i)}(t) \equiv df_j|_{\tilde{\boldsymbol{x}}_j^\intercal \boldsymbol{W}_{-i}(t)} \in \mathcal{M}_k$, $\boldsymbol{D}^{(i)}(t) \in \mathcal{M}_{kn}$ is a block-diagonal matrix with block-diagonal matrices $\boldsymbol{D}_j^{(i)}(t) \in \mathcal{M}_k$ for $j \in [n]$, and finally $\tilde{\boldsymbol{X}}_{-i} \equiv (\tilde{\boldsymbol{x}}_1, \ldots, \tilde{\boldsymbol{x}}_{i-1}, 0, , \tilde{\boldsymbol{x}}_{i+1}, \ldots, \tilde{\boldsymbol{x}}_n) \in \mathcal{M}_{pk,kn}$ ($\tilde{\boldsymbol{X}}_{-i}$ is independent of $\tilde{\boldsymbol{x}}_i$).

Relying on concentration of measure arguments [LC20], the random vector $\boldsymbol{Q}_{-i}(t)\tilde{\boldsymbol{x}}_i$ is almost constant in terms of t and thus almost equal to $\boldsymbol{Q}_{-i}(0)\tilde{\boldsymbol{x}}_i$. Moreover, the fact that $\boldsymbol{Q}_{-i}(0)$ (also simply denoted $\boldsymbol{Q}_{-i}$) is independent of $\tilde{\boldsymbol{x}}_i$ allows us to integrate the identity (6) to obtain the core result of the article relating $\boldsymbol{W}$ and $\boldsymbol{W}_{-i}$. Specifically, we have the following concentration inequality, for some constants $C, c > 0$:

$$\forall t > 0 \,:\, \mathbb{P}\left(\left\| \tilde{\boldsymbol{x}}_i^\intercal \boldsymbol{W} - \tilde{\boldsymbol{x}}_i^\intercal \boldsymbol{W}_{-i} + \frac{1}{n}\tilde{\boldsymbol{x}}_i^\intercal \boldsymbol{Q}_{-i}\tilde{\boldsymbol{x}}_i f_i(\tilde{\boldsymbol{x}}_i^\intercal \boldsymbol{W})\right\| \geq t \mid \mathcal{A}_X\right) \leq Ce^{-cnt^2/\log n}. \tag{7}$$

5.2 Estimating the mean and covariance of the Softmax weights

By breaking the statistical dependencies of the problem through $\boldsymbol{W}_{-i}$, we may now access and estimate the statistics $\boldsymbol{m}_W$ and $\boldsymbol{C}_W$. This precisely comes from a deterministic approximation of the quadratic form $\frac{1}{n}\tilde{\boldsymbol{x}}_i^\intercal \boldsymbol{Q}_{-i}\tilde{\boldsymbol{x}}_i$ in (7) in the large n, p limit, a result inspired by [LC20]:

Proposition 5.1. *For any $\ell \in [k]$, let n_ℓ be the number of columns of $\boldsymbol{X}$ in class $\mathcal{C}_\ell$ and, for any block diagonal matrix $\boldsymbol{\Delta} = \operatorname{Diag}(\boldsymbol{\Delta}_\ell)_{1 \leq \ell \leq k} \in \mathcal{M}_{k^2}$ ($\forall \ell \in [k], \boldsymbol{\Delta}_\ell \in \mathcal{M}_k$), let*

$$\bar{\boldsymbol{Q}}(\boldsymbol{\Delta}) \equiv \left(\boldsymbol{\Lambda} - \sum_{a=1}^{k}\frac{n_a}{n}\Gamma_a(\boldsymbol{\Delta}_a) \otimes \boldsymbol{S}_a\right)^{-1} = \begin{pmatrix} \bar{\boldsymbol{Q}}_{1,1}(\boldsymbol{\Delta}) & \dots & \bar{\boldsymbol{Q}}_{1,k}(\boldsymbol{\Delta}) \\ \vdots & \ddots & \vdots \\ \bar{\boldsymbol{Q}}_{k,1}(\boldsymbol{\Delta}) & \dots & \bar{\boldsymbol{Q}}_{k,k}(\boldsymbol{\Delta}) \end{pmatrix} \in \mathcal{M}_{kp},$$

where $\Gamma_\ell(\boldsymbol{\Delta}_a) = \mathbb{E}\left[(\boldsymbol{I}_k - \boldsymbol{D}_j^{-i}(0)\boldsymbol{\Delta}_a)^{-1}\boldsymbol{D}_j^{-i}(0)\right]$ for $\boldsymbol{x}_j$ in class $\mathcal{C}_\ell$ and $\boldsymbol{S}_\ell = \mathbb{E}[\boldsymbol{x}_j\boldsymbol{x}_j^\intercal]$. Then the fixed point equation

$$\boldsymbol{\Delta}_\ell = \left[\frac{1}{n}\operatorname{Tr}\left(\boldsymbol{S}_\ell \bar{\boldsymbol{Q}}_{a,b}(\boldsymbol{\Delta})\right)\right]_{1 \leq a,b \leq k}$$

admits a unique solution $\boldsymbol{\Delta} \in \mathcal{M}_{k^2}$ that satisfies, for any $\boldsymbol{x}_i$ is in class $\ell \in [k]$,

$$\forall t > 0 \,:\, \mathbb{P}\left(\left\|\frac{1}{n}\tilde{\boldsymbol{x}}_i^\intercal \boldsymbol{Q}_{-i}\tilde{\boldsymbol{x}}_i - \boldsymbol{\Delta}_\ell\right\| \geq t \mid \mathcal{A}_X\right) \leq Ce^{-cnt^2/\log n} \quad \textit{for some constants } C, c > 0.$$

From this result, using the identity (7), we then obtain an estimation for $\tilde{\boldsymbol{x}}_i^\intercal \boldsymbol{W}$:

Proposition 5.2. *For any $\boldsymbol{v} \in \mathbb{R}^k$, there exists a unique point $g_i(\boldsymbol{v}) \in \mathbb{R}^k$ satisfying:*

$$g_i(\boldsymbol{v}) = \boldsymbol{v} - \boldsymbol{\Delta}_i f_i(g_i(\boldsymbol{v})),$$

and, for some constants $C, c > 0$,

$$\mathbb{P}\left(\|\tilde{\boldsymbol{x}}_i^\intercal \boldsymbol{W} - g_i(\tilde{\boldsymbol{x}}_i^\intercal \boldsymbol{W}_{-i})\| \geq t \mid \mathcal{A}_X\right) \leq Ce^{-cnt^2/\log n}.$$

Therefor, letting $h_i = f_i \circ g_i$, by Hölder's inequality [Fin92], since $\tilde{\boldsymbol{x}}_i$ is concentrated,

$$\left\|\boldsymbol{m}_W - \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{\Lambda}^{-1}\mathbb{E}[\tilde{\boldsymbol{x}}_i h_i(\tilde{\boldsymbol{x}}_i^\intercal \boldsymbol{W}_{-i})]\right\| = O\left(\sqrt{\frac{\log n}{n}}\right)$$

$$\left\|\boldsymbol{C}_W - \frac{1}{n^2}\sum_{i=1}^{n}\boldsymbol{\Lambda}^{-1}\mathbb{E}[\tilde{\boldsymbol{x}}_i h_i(\tilde{\boldsymbol{x}}_i^\intercal \boldsymbol{W}_{-i})h_i(\tilde{\boldsymbol{x}}_i^\intercal \boldsymbol{W}_{-i})^\intercal \tilde{\boldsymbol{x}}_i^\intercal]\boldsymbol{\Lambda}^{-1}\right\|_* = O\left(\sqrt{\frac{\log n}{n}}\right)$$

Knowing from Theorem 2.2 that $\tilde{\boldsymbol{x}}_i^\intercal \boldsymbol{W}_{-i}$ is asymptotically Gaussian, $\mathbb{E}[\tilde{\boldsymbol{x}}_i h_i(\tilde{\boldsymbol{x}}_i^\intercal \boldsymbol{W}_{-i})]$ and $\mathbb{E}[\tilde{\boldsymbol{x}}_i h_i(\tilde{\boldsymbol{x}}_i^\intercal \boldsymbol{W}_{-i})h_i(\tilde{\boldsymbol{x}}_i^\intercal \boldsymbol{W}_{-i})^\intercal \tilde{\boldsymbol{x}}_i^\intercal]$ can be explicitly evaluated (for instance using Stein's Lemma [LN08]), and only depend on the statistical means and covariances of $\tilde{\boldsymbol{x}}_1, \dots, \tilde{\boldsymbol{x}}_n$ and of $\boldsymbol{W}_{-i}$ (which has the same statistics as $\boldsymbol{W}$). Their exact expressions are provided in Section 3 of the Supplementary Material. Finally, let us introduce the $2k$ functions $m_1, \dots, m_k : \mathbb{R}^{kp} \times \mathcal{M}_{kp} \to \mathbb{R}^{kp}$ and $c_1, \dots, c_k : \mathbb{R}^{kp} \times \mathcal{M}_{kp} \to \mathcal{M}_{kp}$ defined, $\forall i \in [n]$, by

$$m_{k(i)}(\boldsymbol{m}_W, \boldsymbol{C}_W) = \boldsymbol{\Lambda}^{-1}\mathbb{E}[\tilde{\boldsymbol{x}}_i h_i(\tilde{\boldsymbol{x}}_i^\intercal \boldsymbol{W}_{-i})]$$

$$c_{k(i)}(\boldsymbol{m}_W, \boldsymbol{C}_W) = \frac{1}{n}\boldsymbol{\Lambda}^{-1}\mathbb{E}[\tilde{\boldsymbol{x}}_i h_i(\tilde{\boldsymbol{x}}_i^\intercal \boldsymbol{W}_{-i})h_i(\tilde{\boldsymbol{x}}_i^\intercal \boldsymbol{W}_{-i})^\intercal \tilde{\boldsymbol{x}}_i^\intercal]\boldsymbol{\Lambda}^{-1},$$

where $k(i)$ denotes the class of $\tilde{\boldsymbol{x}}_i$ and $\boldsymbol{m}_W$ and $\boldsymbol{C}_W$ are respectively the mean and covariance of $\boldsymbol{W}$. The mappings $(m_\ell)_{1 \leq \ell \leq k}$ and $(\sigma_\ell)_{1 \leq \ell \leq k}$ are uniquely determined by the means $\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_n$ and the covariances $\boldsymbol{\Sigma}_1, \dots, \boldsymbol{\Sigma}_n$. In particular, the deterministic pair $(\bar{\boldsymbol{m}}_W, \bar{\boldsymbol{C}}_W)$, defined as the unique solution of

$$\bar{\boldsymbol{m}}_W = \sum_{\ell=1}^{k}\frac{n_\ell}{n}m_\ell(\bar{\boldsymbol{m}}_W, \bar{\boldsymbol{C}}_W) \quad \text{and} \quad \bar{\boldsymbol{C}}_W = \sum_{\ell=1}^{k}\frac{n_\ell}{n}c_\ell(\bar{\boldsymbol{m}}_W, \bar{\boldsymbol{C}}_W), \tag{8}$$

is a good approximation for $(\boldsymbol{m}_W, \boldsymbol{C}_W)$ as stated in Theorem 3.2.

Broader Impact

In this work, we provided a theoretical analysis of the widely used Softmax component in neural network classifiers, by deriving the exact high dimensional asymptotic performances of this classifier in terms of its inputs statistics. An important positive impact of this work is the in-depth analysis and understanding of the interplay between the data representations and the last trained layer of neural networks. The developed theoretical approach in this study presents fair and non-offensive societal consequences.

References

[BDS18] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018.

[CHM+15] Anna Choromanska, Mikael Henaff, Michael Mathieu, Gérard Ben Arous, and Yann LeCun. The loss surfaces of multilayer networks. In *Artificial intelligence and statistics*, pages 192–204, 2015.

[dCPS+18] Remi Tachet des Combes, Mohammad Pezeshki, Samira Shabanian, Aaron Courville, and Yoshua Bengio. On the learning dynamics of deep neural networks. *arXiv preprint arXiv:1809.06848*, 2018.

[DDS+09] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[FGP07] B. Fleury, O. Guédon, and G. Paouris. A stability result for mean width of l p -centroid bodies. *Advances in Mathematics*, 214:865–877, 2007.

[Fin92] Helmut Finner. A generalization of holder's inequality and some probability inequalities. *The Annals of probability*, pages 1893–1901, 1992.

[GCM18] Samantha Guerriero, Barbara Caputo, and Thomas Mensink. Deep nearest class mean classifiers. In *International Conference on Learning Representations, Worskhop Track*, 2018.

[GP17] Bolin Gao and Lacra Pavel. On the properties of the softmax function with application in game theory and reinforcement learning. *arXiv preprint arXiv:1704.00805*, 2017.

[GPAM+14] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[HRU+17] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in neural information processing systems*, pages 6626–6637, 2017.

[KGC18] Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7482–7491, 2018.

[Kla07] B. Klartag. A central limit theorem for convex sets. *Inventiones mathematicae volume*, 168:pages91–131, 2007.

[KXR+19] Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Albert Gordo, Jiashi Feng, and Yannis Kalantidis. Decoupling representation and classifier for long-tailed recognition, 2019.

[LC20] Cosme Louart and Romain Couillet. Concentration of measure and large random matrices with an application to sample covariance matrices. *submitted to Random Matrices: Theory and Applications*, 2020.

[Led05] Michel Ledoux. *The concentration of measure phenomenon*. Number 89. American Mathematical Soc., 2005.

[LLC+18] Cosme Louart, Zhenyu Liao, Romain Couillet, et al. A random matrix approach to neural networks. *The Annals of Applied Probability*, 28(2):1190–1248, 2018.

[LN08] Zinoviy Landsman and Johanna Nešlehová. Stein's lemma for elliptical random vectors. *Journal of Multivariate Analysis*, 99(5):912–927, 2008.

[LWL+17] Xuezhi Liang, Xiaobo Wang, Zhen Lei, Shengcai Liao, and Stan Z Li. Soft-margin softmax for deep classification. In *International Conference on Neural Information Processing*, pages 413–421. Springer, 2017.

[MVPC13] Thomas Mensink, Jakob Verbeek, Florent Perronnin, and Gabriela Csurka. Distance-based image classification: Generalizing to new classes at near-zero cost. *IEEE transactions on pattern analysis and machine intelligence*, 35(11):2624–2637, 2013.

[NBA+18] Roman Novak, Yasaman Bahri, Daniel A Abolafia, Jeffrey Pennington, and Jascha Sohl-Dickstein. Sensitivity and generalization in neural networks: an empirical study. *arXiv preprint arXiv:1802.08760*, 2018.

[PB17] Jeffrey Pennington and Yasaman Bahri. Geometry of neural network loss surfaces via random matrix theory. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 2798–2806. JMLR. org, 2017.

[PGM+19] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, pages 8024–8035, 2019.

[PRU+18] Kristina Preuer, Philipp Renz, Thomas Unterthiner, Sepp Hochreiter, and Günter Klambauer. Fréchet chemblnet distance: A metric for generative models for molecules. *arXiv preprint arXiv:1803.09518*, 2018.

[PW17] Jeffrey Pennington and Pratik Worah. Nonlinear random matrix theory for deep learning. In *Advances in Neural Information Processing Systems*, pages 2637–2646, 2017.

[SIVA17] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Thirty-first AAAI conference on artificial intelligence*, 2017.

[SLTC20] Mohamed El Amine Seddik, Cosme Louart, Mohamed Tamaazousti, and Romain Couillet. Random matrix theory proves that deep learning representations of gan-data behave as gaussian mixtures. *arXiv preprint arXiv:2001.08370*, 2020.

[SMG13] Andrew M Saxe, James L McClelland, and Surya Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *arXiv preprint arXiv:1312.6120*, 2013.

[YKYR18] Chih-Kuan Yeh, Joon Kim, Ian En-Hsu Yen, and Pradeep K Ravikumar. Representer point selection for explaining deep neural networks. In *Advances in Neural Information Processing Systems*, pages 9291–9301, 2018.

[YW19] Yaqiong Yao and HaiYing Wang. Optimal subsampling for softmax regression. *Statistical Papers*, 60(2):235–249, 2019.