The Unexpected Deterministic and Universal Behavior of Large Softmax Classifiers

Anonymous Author
Anonymous Institution

Abstract

This paper provides a large dimensional analysis of the Softmax classifier. We discover and prove that, when the classifier is trained on data satisfying loose statistical modeling assumptions, its weights become deterministic and solely depend on the data statistical means and covariances. 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 theoretically as well as numerically sustained on CNN representations of images produced by GANs.

1 Introduction

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

At the very source of these difficulties are the implicit optimization scheme as well as the multiple non-linearities involved in the network design: the activation functions in the intermediate layers and the soft or hard decision in the last layer [LWL+17]. For lack of a tractable comprehensive analysis, literature studies have mostly focused on individual network components or rough network approximations. For instance, the effect of non-linearities in a single-hidden layer network was analysed in [PW17, LLC+18],

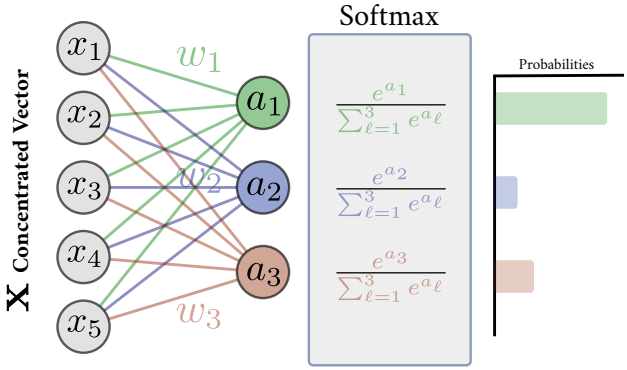

Figure 1: Illustration of the Softmax classifier with *concentrated random vectors* [Led05, LC18] (belonging to some space $\mathcal{X}$) as input data, i.e., satisfying the concentration property $\mathbb{P}\left(|\varphi(\boldsymbol{x}) - \mathbb{E}\varphi(\boldsymbol{x})| > t\right) \leq C\, e^{-(t/\sigma)^q}$, valid for all 1-Lipschitz $\varphi : \mathcal{X} \to \mathbb{R}$ (see Definition 1). GAN data as well as their deep network-based representations are practical examples of such random vectors [SLTC20].

the learning dynamics in elementary network designs in [SMG13, dCPS+18] and the basic understanding of the loss surface geometry in a largely approximated version of the deep network in [PB17, CHM+15].

These analyses provide basic behavioral intuitions but fail to provide a performance assessment at the final decision stage. As a first answer, the present article instead focuses on the analysis of the weights and performance of the Softmax classifier, commonly used as the last decision step in neural networks. This classifier has the property, of key importance here, to be optimal for Gaussian mixture inputs with equal covariance [YW19]. Modelling the input features of this classifier as *concentrated random vectors* [Led05] (see Figure 1), 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

Preliminary work. Under review by AISTATS 2021. Do not distribute.

article studies the statistical behavior of the Softmax classifier *when trained independently of the remainder of the network*.

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

From a technical standpoint, as the (isolated) Softmax classifier training corresponds to a (possibly non-convex) optimization problem, our analysis of the Softmax weights is performed by first expressing the optimization as a contracting fixed point equation, and then showing that the *concentration properties* of the data features "propagate" to the solutions of the fixed-point equation and thus to the Softmax weights. This importantly implies that, for large n, p, the Softmax weights become fully deterministic and can be explicitly evaluated as a function of the data statistics and the Softmax parameters. These conclusions 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, in particular, *the Softmax classifier 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, KG17];

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].

The remainder of the article introduces works related to the present analysis (Section 2), before precisely introducing the Softmax classifier and data model under study, along with basic concentration of measure prerequisites (Section 3). Our main theoretical results are developed in Section 4 along with supporting experiments, finally discussed in Section 5. The main derivations of our results are deferred to the Supplementary Material.

2 Related Works

The Softmax activation is commonly used as an output activation of deep neural networks in many applications [HZRS16, SVL14, CVMG+14, GMH13] to model categorical probability distributions [Bri90]. It is also used in some recent learning mechanisms such as attention models [VSP+17], at the core of a variety of very efficient NLP models known as *transformers* [TDBM20].

Significant efforts have been made on the analysis and improvement of the Softmax classifier: the authors in [KFYA18] highlight the source of the bottleneck effect of Softmax and propose an alternative which improves the performance for language modelling; in order to reduce the computational cost of training with Softmax, the authors in [RCY+19] propose a sampled version of Softmax relying on random Fourier features; in [LWYY16], a generalized large-margin Softmax is devised to enforce intra-class compactness and inter-class separability between learned features in convolutional neural networks; finally and closer to our present findings, the article [KG17] develops a structured classification model relying on Softmax, which is the state-of-the-art approach for deep learning heteroscedastic classification – specifically, the model places a Gaussian distribution on the logits of a standard Softmax classification model. By describing the actual behavior of the Softmax classifier on realistic data models (based on concentration assumptions on data; see next), our present findings support the Gaussianity assumption on the logits as made by [KG17], which, to the best of our knowledge, constitutes a first theoretical justification of this assumption.

Our approach is closely related to the analysis of the logistic regression model in [EKBB+13, MLC19] with Gaussian assumptions on data, although we generalize these ideas to a k-class mixture model under the more general *concentration* assumption on the input data.

Notation: For $m \in \mathbb{N}$, $[m] \equiv \{1, \dots, m\}$. 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); $\|\cdot\|_F$ stands for the Frobenius norm.

3 Model setting

3.1 The Softmax classifier

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

$$\mathcal{L}(\boldsymbol{w}_1, \dots, \boldsymbol{w}_k) = -\frac{1}{n} \sum_{i=1}^n \sum_{\ell=1}^k y_{i\ell} \log p_{i\ell} + \frac{1}{2} \sum_{\ell=1}^k \lambda_\ell \|\boldsymbol{w}_\ell\|^2$$

with $p_{i\ell} = \frac{\phi(\boldsymbol{w}_\ell^\intercal \boldsymbol{x}_i)}{\sum_{j=1}^k \phi(\boldsymbol{w}_j^\intercal \boldsymbol{x}_i)}$ for some real-valued function $\phi : \mathbb{R} \to \mathbb{R}$. In particular, $\phi(t) = e^t$ for the classical Softmax classifier [GP17]. Cancelling the gradient of the loss with respect to each weight vector $\boldsymbol{w}_\ell$ yields, for each $\ell \in [k]$,

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

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 assuming p, n large, we subsequently show that the stacked vector $\boldsymbol{W} \equiv [\boldsymbol{w}_1^\intercal, \dots, \boldsymbol{w}_k^\intercal]^\intercal \in \mathbb{R}^{pk}$ has a tractable behavior, which in turn allows us to accurately predict the performances of the Softmax classifier.

3.2 Mixture of concentrated vectors as input

This section introduces the statistical data model used to study the behavior of the weight vector $\boldsymbol{W}$. We first characterize the data classes: if $\boldsymbol{x}_i \in \mathcal{C}_\ell$, then $\boldsymbol{x}_i \in \mathbb{R}^p$ is a random vector with

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

and we note $\gamma_\ell = \frac{\#\mathcal{C}_\ell}{n}$, the proportion of data in class $\mathcal{C}_\ell$. The vectors $\boldsymbol{x}_1, \dots, \boldsymbol{x}_n$ are further assumed to be independent and such that $\boldsymbol{X} = [\boldsymbol{x}_1, \dots, \boldsymbol{x}_n] \in \mathcal{M}_{p,n}$ satisfies a *concentration* property. To properly state this central assumption (Assumption 1 below), the notion of concentrated random vectors needs be defined.

Definition 1 (Concentrated vector)**.** *Given a normed vector space $(\mathcal{X}, \|\cdot\|)$ and $q > 0$, a random vector $\boldsymbol{x} \in \mathcal{X}$ is said to be q-exponentially concentrated if for any 1-Lipschitz $\varphi : \mathcal{X} \to \mathbb{R}$, there exists $C \geq 0$ independent of $\dim(\mathcal{X})$ and $\sigma > 0$ such that, for all $t \geq 0$,*

$$\mathbb{P}\left(|\varphi(\boldsymbol{x}) - \mathbb{E}\varphi(\boldsymbol{x})| > t\right) \leq C\, e^{-(t/\sigma)^q}. \tag{2}$$

This is denoted as $\boldsymbol{x} \propto \mathcal{E}_q(\sigma)$, where σ is called the observable diameter*. If σ does not depend on $\dim(\mathcal{X})$ we simply write $\boldsymbol{x} \propto \mathcal{E}_q$.*

The prototypical example of a concentrated random vector is the Gaussian random vector $\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_p)$ for which $\boldsymbol{z} \propto \mathcal{E}_2$ [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 3.1 (Stability through Lipschitz transformations)**.** *It is easily deduced from Definition 1 that, given some $\mathcal{Z} \ni \boldsymbol{z} \propto \mathcal{E}_q$ and an L-Lipschitz transformation $\mathcal{G} : \mathcal{Z} \to \mathcal{X}$ (L might depend on $\dim(\mathcal{X})$), the concentration property on $\boldsymbol{z}$ is transferred to $\mathcal{G}(\boldsymbol{z})$. Specifically, $\mathcal{G}(\boldsymbol{z}) \propto \mathcal{E}_q(L)$. Indeed, for all $\varphi : \mathcal{X} \to \mathbb{R}$, 1-Lipschitz, $\frac{1}{L}\varphi \circ \mathcal{G}$ is 1-Lipschitz, and one can apply (2) to $\frac{t}{L}$.*

The concentration of Gaussian vectors combined with the stability through Lipschitz transformations as per Remark 3.1 provides a wide range of concentrated random vector families with possibly quite complex dependence structures. A remarkable example of such random vectors are random vectors produced by generative adversarial networks (GANs) [GPAM+14]: GAN networks[2] are such that their outputs have the same concentration[3] as their inputs [SLTC20]; in particular, for Gaussian $\mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_d)$ inputs (as traditionally assumed) whose observable diameter does not depend on the dimension d, the observable diameter of the GAN generator outputs does not increase with the output data dimension. Further operations through 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 further consequence of the above remark, making the reasonable 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 concentrated random vectors with observable diameter independent of their dimension. For our present interest, this assumption is summarized as follows:

Assumption 1 (Concentrated data)**.** *Letting $\boldsymbol{X} = [\boldsymbol{x}_1, \dots, \boldsymbol{x}_n] \in \mathcal{M}_{p,n}$ with independent $\boldsymbol{x}_i$'s, $\boldsymbol{X} \propto \mathcal{E}_2$ in the sense of Definition 1.*

[1] Biases are not introduced in the present formulation as their effect is known to be negligible in practice [KXR+19] and would impede readability.

[2] Specifically, the generator part after training.

[3] When the GAN model has a controlled Lipschitz constant, which is practically ensured by *spectral normalization* as in the BigGAN model [BDS18].

Our objective is to "transfer" the concentration of $\boldsymbol{X}$ to the weight vector $\boldsymbol{W}$ as defined in (1). To this end, we demand that the number of data n be of the same order of magnitude as their dimension p.

Assumption 2 (Growth rate)**.** *As $n \to \infty$, $p/n \to c \in (0, \infty)$ and for each $\ell \in [k]$, $\|\boldsymbol{\mu}_\ell\| \leq \mathcal{O}(1)$*[4]*.*

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

Theorem 3.2 (CLT for concentrated vectors [Kla07, FGP07])**.** *Let $\boldsymbol{x} \in \mathbb{R}^p$ be a random vector with $\mathbb{E}[\boldsymbol{x}] = \boldsymbol{0}$ and $\mathbb{E}[\boldsymbol{x}\boldsymbol{x}^\intercal] = \boldsymbol{I}_p$, and ν 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 exist two constants $C, c > 0$ and a set $\Theta \subset \mathcal{S}^{p-1}$ such that $\nu(\Theta) \geq 1 - \sqrt{p}Ce^{-c\sqrt{p}}$ 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 $G(t) = \frac{1}{\sqrt{2\pi}} \int_t^\infty e^{-u^2} du$.

4 Main results

4.1 Convergence of the Softmax weights

This section characterizes the statistical behavior of the softmax classifier weights $\boldsymbol{W} \equiv [\boldsymbol{w}_1^\intercal, \ldots, \boldsymbol{w}_k^\intercal]^\intercal \in \mathbb{R}^{pk}$, under Assumptions 1–2 and, as a result, retrieves the (asymptotic) exact classifier performance. The complete proofs of the results can be found in [LC20b].

For readability in the following, we restrict ourselves to *scalar labels* (thus in $\mathbb{R}$ rather than $\mathbb{R}^k$) and synthesize (1) under the compact form

$$\boldsymbol{w} = \frac{1}{n} \sum_{i=1}^n f(\boldsymbol{w}^\intercal \boldsymbol{x}_i) \boldsymbol{x}_i, \tag{3}$$

where $\boldsymbol{w} \in \mathbb{R}^p$ plays the role of the weight vector and $f : \mathbb{R} \to \mathbb{R}$ is a scalar function (parametrized by ϕ and λ in the formulation of the Softmax classifier). The results specific to the generic Softmax classifier (1) with arbitrary k are only more technical; the detail is left to the Supplementary Material.

[4]When not satisfied, this assumption is classically obtained after a re-centering of the data, the dependence between the data brought by the re-centering is limited and can be controlled.

Equation (3) can be further simplified as the fixed-point equation $\boldsymbol{w} = \Psi(\boldsymbol{w})$ where Ψ is the random mapping defined for any $\boldsymbol{z} \in \mathbb{R}^p$ as: $\Psi(\boldsymbol{z}) = \frac{1}{n} \boldsymbol{X} f(\boldsymbol{X}^T \boldsymbol{z})$ (with f applied entry-wise). For $\boldsymbol{w}$ to be well-defined and for the concentration of $\boldsymbol{X}$ to propagate into $\boldsymbol{w}$, the mapping $\Psi : \boldsymbol{w} \mapsto \frac{1}{n} \boldsymbol{X} f(\boldsymbol{X}^T \boldsymbol{w})$ is required to have contraction properties; assuming f differentiable, this holds under the event

$$\mathcal{A}_{\boldsymbol{w}} = \left\{ \frac{1}{n} \|f'\|_\infty \|\boldsymbol{X}\boldsymbol{X}^T\| \geq 1 - \varepsilon \right\}.$$

For this event to be highly probable, we introduce some regularizing properties on f and $\boldsymbol{X}$.

Assumption 3 (Contractivity)**.** *The mapping f is differentiable and there exists $\varepsilon > 0$ independent of n, p such that*

$$\frac{1}{n} \|f'\|_\infty \mathbb{E}[\|\boldsymbol{X}\boldsymbol{X}^T\|] \leq 1 - 2\varepsilon.$$

Remark 4.1 (Regularization parameter thresholding)**.** *For the generic Softmax classification problem, Assumption 3 implies that the parameters λ_ℓ's cannot be chosen too small. Indeed, $\|f'\|_\infty$ being proportional to $1/(\inf_{\ell \in [k]} \lambda_\ell)$, $\frac{1}{n} \|f'\|_\infty \mathbb{E}[\|\boldsymbol{X}\boldsymbol{X}^T\|] \to \infty$ as $\lambda_\ell \to 0$. The upcoming results are thus only valid for sufficiently large λ_ℓ's. Yet, since (1) can be solved for small λ_ℓ's by gradient descent (rather than by fixed-point iterates), one may hope that the article core results (notably Theorem 4.8) still hold irrespective of $\lambda_\ell > 0$ (although the theoretical estimates may not be accessible through fixed-point iterations).*

Then we have the following lemma, proved in the Supplementary Material.

Lemma 4.2. *There exist two constants $C, c > 0$ independent of p, n such that $\mathbb{P}(\mathcal{A}_{\boldsymbol{w}}^c) \leq Ce^{-cn}$.*

Under these conditions, our main result guarantees the transfer of the concentration of $\boldsymbol{X}$ into $\boldsymbol{w}$.

Theorem 4.3 (Concentration of $\boldsymbol{w}$)**.** *Under Assumptions 1-3, $\boldsymbol{w}$* concentrates w.r.t. the event $\mathcal{A}_{\boldsymbol{w}}$[5] *as*

$$(\boldsymbol{w} \mid \mathcal{A}_{\boldsymbol{w}}) \propto \mathcal{E}_2 \left(\frac{1}{\sqrt{n}} \right).$$

Since their observable diameter $1/\sqrt{n}$ vanishes for n, p large, Theorem 4.3 ensures that *the random weight vector $\boldsymbol{w}$ becomes deterministic as p, n grow.* We now

[5]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 $\forall B \in \mathcal{F}$, $\mathbb{P}_{\mathcal{A}}(B) = \mathbb{P}(B)/\mathbb{P}(\mathcal{A})$.

fine-tune this result by characterizing its first and second order statistics

$$\boldsymbol{\mu}_{\boldsymbol{w}} \equiv \mathbb{E}[\boldsymbol{w}], \quad \boldsymbol{C}_{\boldsymbol{w}} \equiv [\boldsymbol{w}\boldsymbol{w}^\intercal] - \boldsymbol{\mu}_{\boldsymbol{w}}\boldsymbol{\mu}_{\boldsymbol{w}}^\intercal,$$

for all finite but large n, p. The estimation of $\boldsymbol{\mu}_{\boldsymbol{w}}$ and $\boldsymbol{C}_{\boldsymbol{w}}$ unfolds in two steps: (i) a first control of the statistical dependence between $\boldsymbol{w}$ and $\boldsymbol{X}$, delineated in Subsection 4.2, and (ii) the proper evaluation of $\boldsymbol{m}_{\boldsymbol{w}}$ and $\boldsymbol{C}_{\boldsymbol{w}}$, in Subsection 4.3.

4.2 Control of the weight-data dependence

Taking the expectation on both sides of (3), the main technical difficulty arises from the evaluation of $\mathbb{E}[\boldsymbol{x}_i f(\boldsymbol{x}_i^\intercal \boldsymbol{w})]$ due to the elaborate dependence between $\boldsymbol{w}$ and $\boldsymbol{x}_i$. Our approach consists in approximating $f(\boldsymbol{x}_i^\intercal \boldsymbol{w})$ with a functional $\xi_{k(i)}(\boldsymbol{x}_i^\intercal \boldsymbol{w}_{-i})$ where $k(i)$ is the class of $\boldsymbol{x}_i$, $\xi_{k(i)} : \mathbb{R} \to \mathbb{R}$ is deterministic and $\boldsymbol{w}_{-i}$ is the vector $\boldsymbol{w}$ deprived of the contribution of $\boldsymbol{x}_i$, i.e., the solution to

$$\boldsymbol{w}_{-i} = \frac{1}{n} f(\boldsymbol{X}_{-i}^\intercal \boldsymbol{w}_{-i}) \boldsymbol{X}_{-i},$$

where $\boldsymbol{X}_{-i} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{i-1}, \boldsymbol{0}, \boldsymbol{x}_{i+1}, \ldots, \boldsymbol{x}_n] \in \mathcal{M}_{p,n}$. From there, we are left to estimating $\mathbb{E}[\boldsymbol{x}_i \xi_{k(i)}(\boldsymbol{x}_i^\intercal \boldsymbol{w}_{-i})]$ which is far easier to handle as Theorem 3.2 ensures that $z_i \equiv \boldsymbol{x}_i^\intercal \boldsymbol{w}_{-i}$ behaves with high probability like a Gaussian variable whose mean and variance can be estimated from the statistics of $\boldsymbol{x}_i$ and $\boldsymbol{w}$ (the latter having the same statistics as $\boldsymbol{w}_{-i}$); see next Figure 3.

The link between $\boldsymbol{w}_{-i}$ and $\boldsymbol{w}$ is made thanks to the interpolation mapping $\boldsymbol{w}_{-i} : [0,1] \to \mathbb{R}^p$, defined for $i \in [n]$ as the unique solution, for all $t \in [0,1]$, to

$$\boldsymbol{w}_{-i}(t) = \frac{f(\boldsymbol{X}_{-i}^\intercal \boldsymbol{w}_{-i}(t))}{n} \boldsymbol{X}_{-i} + \frac{t f'(\boldsymbol{x}_i^\intercal \boldsymbol{w}_{-i}(t))}{n} \boldsymbol{x}_i. \tag{4}$$

This mapping can be seen as a path between the weights vector $\boldsymbol{w} = \boldsymbol{w}_{-i}(1)$ of the Softmax classifier and $\boldsymbol{w}_{-i} = \boldsymbol{w}_{-i}(0)$.

By the inverse function theorem, $t \mapsto \boldsymbol{w}_{-i}(t)$ is shown to be differentiable, and we obtain the explicit formula:

$$\begin{aligned} \boldsymbol{w}_{-i}'(t) &= \frac{1}{n} \chi_i'(t) \boldsymbol{Q}_{-i}(t) \boldsymbol{x}_i, \quad \text{with} \\ \boldsymbol{Q}_{-i}(t) &\equiv \left(\boldsymbol{I}_p - \frac{1}{n} \boldsymbol{X}_{-i} \boldsymbol{D}^{(i)}(t) \boldsymbol{X}_{-i}^\intercal \right)^{-1} \in \mathcal{M}_p, \end{aligned} \tag{5}$$

where $\chi_i(t) \equiv t f'(\boldsymbol{x}_i^\intercal \boldsymbol{w}_{-i}(t))$, $D_j^{(i)}(t) \equiv df_j|_{\boldsymbol{x}_j^\intercal \boldsymbol{w}_{-i}(t)} \in \mathbb{R}$ and $\boldsymbol{D}^{(i)}(t) \in \mathcal{M}_n$ is a diagonal matrix with diagonal entries $D_j^{(i)}(t) \in \mathcal{M}_k$ for $j \in [n]$.

Relying on concentration of measure arguments [LC20a], the random vector $\boldsymbol{Q}_{-i}(t)\boldsymbol{x}_i$ is almost constant w.r.t. t and thus almost equal to $\boldsymbol{Q}_{-i}(0)\boldsymbol{x}_i$. The fact that $\boldsymbol{Q}_{-i}(0)$ (now simply denoted $\boldsymbol{Q}_{-i}$) is additionally independent of $\boldsymbol{x}_i$ allows us to integrate (5) to obtain the core technical result of the article, which relates $\boldsymbol{w}_{-i}$ to $\boldsymbol{w}$. This is achieved under a last very light assumption.

Assumption 4. $\|f''\|_\infty \leq \infty$.

Theorem 4.4. *Under Assumptions 1-4 there exist $C, c > 0$ independent of p, n such that, $\forall t > 0$,*

$$\mathbb{P}_{\mathcal{A}_{\boldsymbol{w}}} \left(\left| \boldsymbol{x}_i^\intercal \boldsymbol{w} - \boldsymbol{x}_i^\intercal \boldsymbol{w}_{-i} + \frac{1}{n} \boldsymbol{x}_i^\intercal \boldsymbol{Q}_{-i} \boldsymbol{x}_i f(\boldsymbol{x}_i^\intercal \boldsymbol{w}) \right| \geq t \right) \leq C e^{-cnt^2}.$$

To estimate $\boldsymbol{x}_i^\intercal \boldsymbol{w}$ as a deterministic functional of $\boldsymbol{x}_i^\intercal \boldsymbol{w}_{-i}$ we still need to estimate $\frac{1}{n} \boldsymbol{x}_i^\intercal \boldsymbol{Q}_{-i} \boldsymbol{x}_i$. This follows from the following random matrix argument (see [LC20a]).

Proposition 4.5. *For any $\ell \in [k]$, let $\bar{D}_\ell = \mathbb{E}_{\mathcal{A}_{\boldsymbol{w}}}[f'(\boldsymbol{x}_i^T \boldsymbol{w})]$ for $\boldsymbol{x}_i$ in $\mathcal{C}_\ell$ and define, for any parameter vector $\boldsymbol{\delta} \in \mathbb{R}^k$, the deterministic matrix*

$$\bar{\boldsymbol{Q}}(\boldsymbol{\delta}) \equiv \left(\boldsymbol{I}_p - \sum_{a=1}^k \frac{\gamma_a \bar{D}_a}{1 - \delta_a \bar{D}_a} \boldsymbol{C}_a \right)^{-1} \in \mathcal{M}_p.$$

Then the system of fixed point equations

$$\forall \ell \in [k] : \delta_\ell = \frac{1}{n} \operatorname{Tr} \left(\boldsymbol{\Sigma}_\ell \bar{\boldsymbol{Q}}(\boldsymbol{\delta}) \right)$$

admits a unique solution $\boldsymbol{\delta} \in \mathbb{R}^k$ such that, for any $i \in [n]$ and for all $t > 0$,

$$\mathbb{P}_{\mathcal{A}_{\boldsymbol{w}}} \left(\left| \frac{1}{n} \boldsymbol{x}_i^\intercal \boldsymbol{Q}_{-i} \boldsymbol{x}_i - \delta_{k(i)} \right| \geq t \right) \leq C e^{-cnt^2},$$

for some $C, c > 0$ independent of p, n.

Theorem 4.4 combined with Proposition 4.5 lead to the approximation:

$$f(\boldsymbol{x}_i^\intercal \boldsymbol{w}) \approx f\left(\boldsymbol{x}_i^\intercal \boldsymbol{w}_{-i} + \delta_{k(i)} f(\boldsymbol{x}_i^\intercal \boldsymbol{w}) \right),$$

which allows us to connect $\boldsymbol{x}_i^\intercal \boldsymbol{w}$ and $z_i \equiv \boldsymbol{x}_i^\intercal \boldsymbol{w}_{-i}$.

Proposition 4.6. *Under Assumptions 1-4 and for any $\ell \in [k]$ and $z \in \mathbb{R}$, the fixed point equation*

$$x = f(z + \delta_\ell x),$$

admits a unique solution $\xi_{\delta_\ell}(z)$. Besides, $\forall t > 0$,

$$\mathbb{P}_{\mathcal{A}_{\boldsymbol{w}}} \left(\left| f(\boldsymbol{x}_i^\intercal \boldsymbol{w}) - \xi_{\delta_{k(i)}}(\boldsymbol{x}_i^\intercal \boldsymbol{w}_{-i}) \right| \geq t \right) \leq C e^{-cnt^2}.$$

Note that for all $z \in \mathbb{R}$ and $\ell \in [k]$, $\xi_{\delta_\ell}'(z) = \frac{f'(z + \delta_\ell \xi_{\delta_\ell}(z))}{1 + \delta_\ell f'(z + \delta_\ell \xi_{\delta_\ell}(z))}$ from which the following result entails.

Lemma 4.7. *Under Assumptions 1-4, $\forall \ell \in [k]$,*

$$\left| \delta_\ell - \frac{1}{n} \operatorname{Tr} \boldsymbol{C}_\ell \left(\boldsymbol{I}_p - \sum_{a=1}^k \gamma_a \mathbb{E}_{\mathcal{A}_{\boldsymbol{w}}}[\xi_{\delta_a}'(\tilde{z}_a)] \boldsymbol{C}_a \right)^{-1} \right| \leq \mathcal{O}(n^{-\frac{1}{2}})$$

where $\tilde{z}_\ell$ is a copy of $z_i \equiv \boldsymbol{x}_i^\intercal \boldsymbol{w}_{-i}$ for $k(i) = \ell$.

4.3 Estimation of the weight statistics

By breaking the statistical dependence of the problem through $\boldsymbol{w}_{-i}$, we may now estimate the statistics $\boldsymbol{m}_{\boldsymbol{w}}$ and $\boldsymbol{C}_{\boldsymbol{w}}$. Indeed, letting $\|\cdot\|_*$ be the nuclear norm[6] and $k(i)$ the class of $\boldsymbol{x}_i$, from the identities

$$\left\| \boldsymbol{\mu}_{\boldsymbol{w}} - \frac{1}{n}\sum_{i=1}^n \mathbb{E}_{\mathcal{A}_w}[\xi_{\delta_{k(i)}}(\boldsymbol{x}_i^\intercal \boldsymbol{w})\boldsymbol{x}_i] \right\| \leq \mathcal{O}\left(n^{-\frac{1}{2}}\right),$$

$$\left\| \boldsymbol{C}_{\boldsymbol{w}} - \frac{1}{n}\sum_{i,j=1}^n \mathbb{E}_{\mathcal{A}_w}[\xi_{\delta_{k(i)}}(z_i)\xi_{\delta_{k(j)}}(z_j)\boldsymbol{x}_i\boldsymbol{x}_j^\intercal] \right\|_* \leq \mathcal{O}\left(n^{-\frac{1}{2}}\right)$$

and Stein-like formulas [Bri90] provided in the Supplementary Material (those can be used since $\boldsymbol{x}_i$ behaves like a Gaussian vector by Theorem 3.2) we deduce an estimate of $\boldsymbol{\mu}_{\boldsymbol{w}}$ and $\boldsymbol{C}_{\boldsymbol{w}}$ depending on the input data statistics $(\boldsymbol{\mu}_\ell)_{\ell\in[k]}$ and $(\boldsymbol{C}_\ell)_{\ell\in[k]}$ as well as on

$$\mathbb{E}_{\mathcal{A}_w}[\xi_{\delta_\ell}(\tilde{z}_\ell)], \quad \mathbb{E}_{\mathcal{A}_w}[\xi'_{\delta_\ell}(\tilde{z}_\ell)], \quad \mathbb{E}_{\mathcal{A}_w}[\xi_{\delta_\ell}(\tilde{z}_\ell)^2],$$

for $\ell \in [k]$. In turn, $\tilde{z}_\ell$ behaving like a Gaussian random variable, the latter quantities only depends on the first order statistics of $\boldsymbol{w}$ and $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$. This brings us to the final result of the article.

Theorem 4.8 (Asymptotic statistics of $\boldsymbol{w}$)**.** *Under Assumptions 1-4, there exists a unique tuple of parameters $(\boldsymbol{\delta}, \boldsymbol{m}, \boldsymbol{\sigma}) \in (\mathbb{R}^k)^3$ satisfying the identities:*

- $\forall \ell \in [k]$: $\tilde{z}_\ell \sim \mathcal{N}(m_\ell, \sigma_\ell^2)$;
- $\forall \ell \in [k]$: $\delta_\ell = \frac{1}{n}\operatorname{Tr}\left(\boldsymbol{C}_\ell\left(\boldsymbol{I}_p - \tilde{\boldsymbol{C}}\right)^{-1}\right)$;
- $\tilde{\boldsymbol{\mu}} \equiv \sum_{\ell=1}^k \gamma_\ell \mathbb{E}[\xi_{\delta_\ell}(\tilde{z}_\ell)]\boldsymbol{\mu}_\ell \in \mathbb{R}^p$;
- $\tilde{\boldsymbol{C}} \equiv \sum_{\ell=1}^k \gamma_\ell \mathbb{E}[\xi_{\delta_\ell}(\tilde{z}_\ell)^2]\boldsymbol{C}_\ell \in \mathcal{M}_p$;
- $\boldsymbol{K} \equiv \sum_{\ell=1}^k \gamma_\ell \mathbb{E}[\xi'_{\delta_\ell}(\tilde{z}_\ell)]\boldsymbol{C}_\ell \in \mathcal{M}_p$;
- $\boldsymbol{R}_1 \equiv (\boldsymbol{I}_p - \boldsymbol{K})^{-1}$;
- $\boldsymbol{R}_2 : \mathcal{M}_p \to \mathcal{M}_p$ *defined, for* $\boldsymbol{M} \in \mathcal{M}_p$, *as*
$$\boldsymbol{R}_2(\boldsymbol{M}) = \boldsymbol{M} + \boldsymbol{K}(\boldsymbol{R}_2(\boldsymbol{M}))\boldsymbol{K};$$
- $m_\ell \equiv \boldsymbol{\mu}_\ell^\intercal \boldsymbol{R}_1 \tilde{\boldsymbol{\mu}}$;
- $\sigma_\ell^2 \equiv \frac{1}{n}\operatorname{Tr}(\boldsymbol{C}_\ell \boldsymbol{R}_2(\tilde{\boldsymbol{C}})) + \tilde{\boldsymbol{\mu}}^\intercal \boldsymbol{R}_1 \boldsymbol{C}_\ell \boldsymbol{R}_1 \tilde{\boldsymbol{\mu}}$.

With these definitions,

$$\|\boldsymbol{\mu}_{\boldsymbol{w}} - \boldsymbol{R}_1\tilde{\boldsymbol{\mu}}\| \leq \mathcal{O}\left(n^{-\frac{1}{2}}\right)$$

$$\left\|\boldsymbol{C}_{\boldsymbol{w}} - \frac{1}{n}\boldsymbol{R}_2(\tilde{\boldsymbol{C}})\right\|_* \leq \mathcal{O}\left(n^{-\frac{1}{2}}\right)$$

and, for all $\ell \in [k]$ and any $\boldsymbol{x} \in \mathcal{C}_\ell$ independent of $\boldsymbol{X}$,

$$|\mathbb{E}[\boldsymbol{x}^\intercal \boldsymbol{w}] - m_\ell|, \left|\mathbb{E}[(\boldsymbol{x}^\intercal \boldsymbol{w})^2] - (\sigma_\ell^2 + m_\ell^2)\right| \leq \mathcal{O}\left(n^{-\frac{1}{2}}\right).$$

Extrapolating Theorem 4.8 to the generic Softmax classifier (thoroughly covered in the Supplementary Material), we obtain that, under data concentration (Assumption 1), the (large n, p) behavior of the Softmax classifier *only depends on the class-wise means and covariances of the input data.* This, we recall, is a direct consequence of (i) the Lipschitz character of the Softmax classifier which preserves concentration (by Lipschitz stability: Remark 3.1) and of (ii) 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 (by Theorem 3.2, these projections induce an asymptotic Gaussian behavior with mean and variance depending *only* on the first order statistics of the data and the weight vector $\boldsymbol{w}$).

As an aftermath of this stable large n, p behavior, the performances of the Softmax classifier are in turn theoretically tractable. Specifically, the asymptotic misclassification probability

$$E_t(\boldsymbol{x} \in \mathcal{C}_\ell) \equiv 1 - \mathbb{P}(\forall j \in [k] \setminus \{\ell\} : p_\ell(\boldsymbol{x}) \geq p_j(\boldsymbol{x}))$$

for $\boldsymbol{x}$ genuinely belonging to class $\ell \in [k]$, with

$$p_\ell(\boldsymbol{z}) = \frac{\phi(\boldsymbol{w}_\ell^\intercal \boldsymbol{z})}{\sum_{j\in[k]} \phi(\boldsymbol{w}_j^\intercal \boldsymbol{z})}, \tag{6}$$

the probability for $\boldsymbol{z}$ to belong to class $\ell \in [k]$, can be inferred as an immediate corollary of Theorem 4.8. Again, $E_t(\boldsymbol{x} \in \mathcal{C}_\ell)$ is only a function of the means and variances $(\boldsymbol{\mu}_\ell)_{\ell\in[k]}$ and $(\boldsymbol{C}_\ell)_{\ell\in[k]}$. This demonstrates the remarkable *universality* property of the Softmax classifier with respect to the data distribution, which we recall is only requested to satisfy the very loose concentration condition of Assumption 1.

4.4 Experimental validation

4.4.1 Synthetic Gaussian and MNIST data

This section aims to validate Theorem 4.8 by means of Algorithm 1 which estimates the quantities $(\boldsymbol{\delta}, \boldsymbol{m}, \boldsymbol{\sigma})$ as defined in Theorem 4.8.[7]

Figure 2 depicts the practical versus theoretical accuracies $1 - E_t$ (based on Theorem 4.8) on synthetic Gaussian data, for varying data dimension p. A perfect match between theory and empirical results are observed both for training and test data, thereby supporting Theorem 4.8 even for not-so-large n, p couples.

[6] For $A \in \mathcal{M}_{p,n}$, $\|A\|_* = \sup_{\|M\|\leq 1} \operatorname{Tr}(AM) = \operatorname{Tr}(\sqrt{AA^T})$.

[7] A Python and Julia implementations of Algorithm 1 will be provided on a Github link.

Algorithm 1: Estimation of the statistics of $\boldsymbol{x}^\intercal \boldsymbol{w}$

Input: Data statistics $\{\boldsymbol{\mu}_\ell, \boldsymbol{C}_\ell, \boldsymbol{\Sigma}_\ell\}_{\ell=1}^k$, scalar function $f : \mathbb{R} \to \mathbb{R}$, precision parameter ϵ and number of drawings T for Monte Carlo (MC) estimation.

Output: $\boldsymbol{\delta}, \boldsymbol{m}, \boldsymbol{\sigma} \in \mathbb{R}^k$

$\boldsymbol{\delta}, \boldsymbol{m}, \boldsymbol{\sigma} \leftarrow \mathbf{1}_k$; $\boldsymbol{\delta}', \boldsymbol{m}', \boldsymbol{\sigma}' \leftarrow 2 \cdot \mathbf{1}_k$;

while $\|\boldsymbol{m} - \boldsymbol{m}'\| + \|\boldsymbol{\sigma} - \boldsymbol{\sigma}'\| + \|\boldsymbol{\delta} - \boldsymbol{\delta}'\| \geq \epsilon$ **do**

  $\boldsymbol{m}' \leftarrow \boldsymbol{m}$; $\boldsymbol{\sigma}' \leftarrow \boldsymbol{\sigma}$; $\boldsymbol{\delta}' \leftarrow \boldsymbol{\delta}$;

  for $\ell \in [k]$ **do**

    - Sample $(z_t)_{t\in[T]} \sim \mathcal{N}(m_\ell, \sigma_\ell^2)$;

    - Estimate $\mathbb{E}[\xi_{\delta_\ell}(\tilde{z}_\ell)]$, $\mathbb{E}[\xi_{\delta_\ell}(\tilde{z}_\ell)^2]$ and $\mathbb{E}[\xi'_{\delta_\ell}(\tilde{z}_\ell)]$ with MC based on $(z_t)_{t\in[T]}$.

  end

  - Compute the quantities $\tilde{\boldsymbol{\mu}}$, $\tilde{\boldsymbol{C}}$, $\boldsymbol{K}$, $\boldsymbol{R}_1$ and $\boldsymbol{R}_2(\tilde{\boldsymbol{C}})$;

  for $\ell \in [k]$ **do**

    - $m_\ell \leftarrow \boldsymbol{\mu}_\ell^\intercal \boldsymbol{R}_1 \tilde{\boldsymbol{\mu}}$;

    - $\sigma_\ell^2 \leftarrow \frac{1}{n} \operatorname{Tr}(\boldsymbol{C}_\ell \boldsymbol{R}_2(\tilde{\boldsymbol{C}})) + \tilde{\boldsymbol{\mu}}^\intercal \boldsymbol{R}_1 \boldsymbol{C}_\ell \boldsymbol{R}_1 \tilde{\boldsymbol{\mu}}$;

    - $\delta_\ell \leftarrow \frac{1}{n} \operatorname{Tr}\left(\boldsymbol{C}_\ell \left(\boldsymbol{I}_p - \tilde{\boldsymbol{C}}\right)^{-1}\right)$.

  end

end

Since our results hold under the broader "quasi-realistic data" Assumption 1, our results are next applied, step by step, to raw data from the MNIST dataset [LeC98], specifically to classify images of the digits "1" and "2" (so $k = 2$ here). Figure 3 first depicts the histograms of the random variables $\boldsymbol{x}_i^\intercal \boldsymbol{w}_{-i}$ and $\boldsymbol{x}_j^\intercal \boldsymbol{w}_{-j}$ with $\boldsymbol{x}_i \in \mathcal{C}_1$ and $\boldsymbol{x}_j \in \mathcal{C}_2$, as well as their estimated Gaussian limits as per Theorem 4.8. Remarkably, even though the input data is far from Gaussian, their projections onto $\boldsymbol{w}$ have clear Gaussian distributions, the means and variances of which are obtained in Theorem 4.8; this result supports the Gaussianity assumption on the logits previously made by Kendall and Gal [KG17]. As a result, the classification accuracy on MNIST data, here depicted in Figure 4, is consistently estimated.

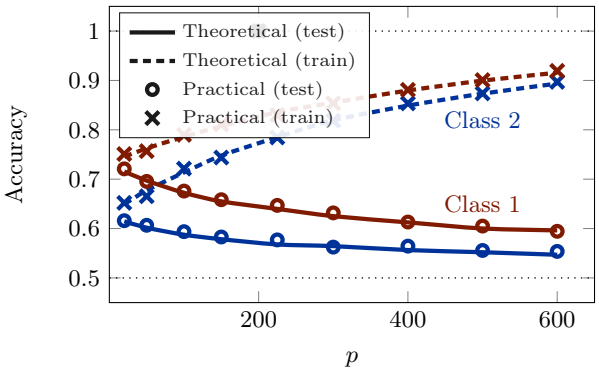

Figure 2: Classification accuracy on Gaussian mixtures; $n = 400$, $\gamma_1 = 1/3$, $\gamma_2 = 2/3$, $\lambda = 20$, $\boldsymbol{C}_1 = \boldsymbol{C}_2 = \boldsymbol{I}_p$, $\|\boldsymbol{\mu}_{1,2}\| = \frac{1}{2}$ and $\boldsymbol{\mu}_1^\intercal \boldsymbol{\mu}_2 = 0$.

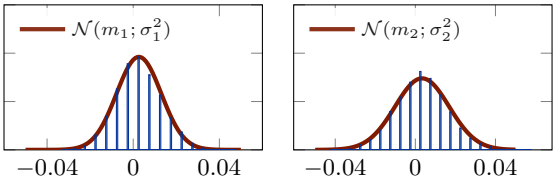

Figure 3: Histogram of **(left)** $z_i \equiv \boldsymbol{x}_i^\intercal \boldsymbol{w}_{-i}$ and **(right)** $z_j \equiv \boldsymbol{x}_j^\intercal \boldsymbol{w}_{-j}$, for $k(i) = 1$, $k(j) = 2$ on MNIST dataset; $\mathcal{C}_1$: digit "1", $\mathcal{C}_2$: digit "2", $\gamma_1 = \gamma_2 = 1/2$, $\lambda = 20$, with centered means. m_ℓ, σ_ℓ, for $\ell \in [k]$, defined in Theorem 4.8.

4.4.2 CNN features of GAN images

This section provides further experiments to support our theoretical findings on CNN representations of GAN-generated images which, unlike the previously studied MNIST images, are truthfully concentrated random vectors. The input data $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n]$ are here independent Resnet18[8] 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 Resnet18) applied to random standard Gaussian noise (as per the BigGAN model), $\boldsymbol{X}$ is concentrated by

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

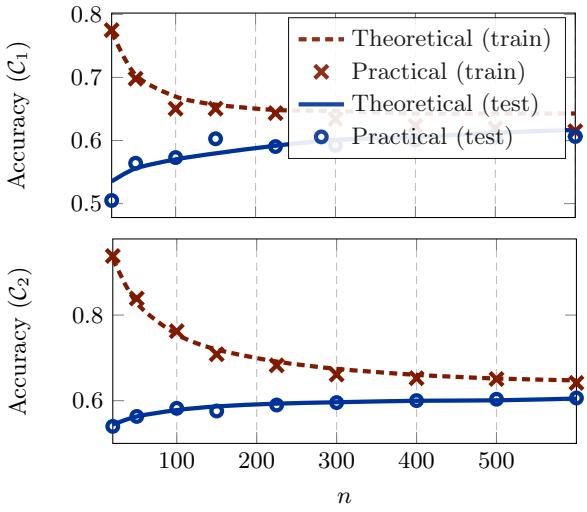

Figure 4: Classification accuracy on MNIST data; $\mathcal{C}_1$: digit "1", $\mathcal{C}_2$: digit "2"; $p = 784$, $\gamma_1 = \gamma_2 = 1/2$, $\lambda = 20$, with centered means.

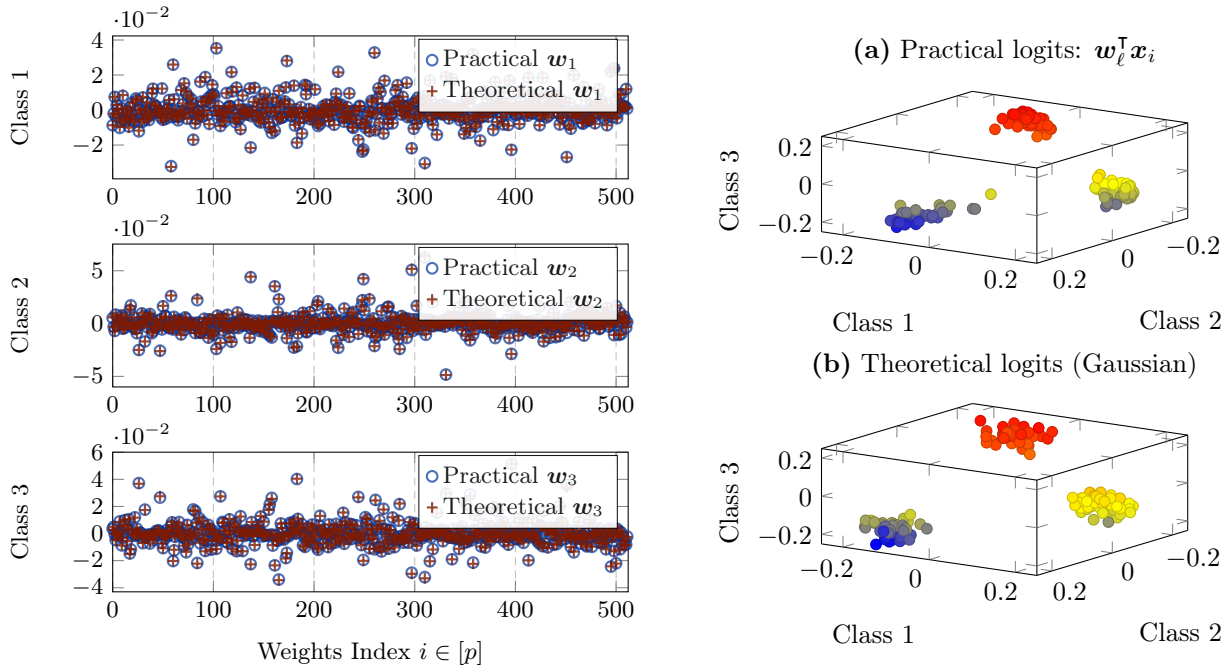

Figure 5: **(Left)** Learned weights (blue circles) versus theoretical estimates (red crosses) from Theorem 4.8. **(Right)** Practical **(a)** versus theory-predicted **(b)** logits, on a test set independent from the training set. The data are Resnet18 [SIVA17] representations ($p = 512$) of BigGAN-generated images [BDS18], which are concentrated vectors by definition [SLTC20]; $k = 3$ classes: *hamburger*, *mushroom*, *pizza*; $n = 3000$; regularization constants $\lambda_1 = \lambda_2 = \lambda_3 = 1.5$; data normalized such that $\|\boldsymbol{x}_i\| = 0.1 \cdot \sqrt{p}$ to ensure $\mathcal{A}_{\boldsymbol{w}}$.

construction and satisfies Assumption 1 (see [SLTC20] for a detailed analysis of the Lipschitz properties of these networks). Under this setting, Figure 5-(left) depicts the learned Softmax weights against their expected large n, p asymptotics as per Theorem 4.8 (see the Supplementary Material for the adaptation of the theorem to the generic Softmax classifier). Despite the finite p, n setting of the simulation, a perfect match is again observed between the learned weights and the theoretical predictions. Further experiments, available in the Supplementary Material, were performed on *real images* from the ImageNet dataset [DDS+09], which confirm this perfect match between theory and practice.

Figure 5-(right) then displays the class-wise scores of a practical Softmax classifier on an independent test set against their simulated Gaussian equivalents predicted by Theorem 4.8. Again here, the empirical and theoretical values agree. The Supplementary Material reports similar outputs for *real* (rather than GAN-produced) ImageNet data. We insist again that, in compliance with Theorem 4.8, the theoretical estimates in all these figures were obtained using *only the empirical class-wise means and covariances* of the input data.[9] Figure 5 thus confirms the theoretically predicted universality of the Softmax classifier.

5 Concluding Remarks

Even though the Softmax classifier has a non-linear nature, a property supposedly useful to extract "deep" non-linear features, the article proved instead that, for reasonably large n, p, the input data are in fact treated as if generated from a mere Gaussian mixture model. This universality phenomenon fundamentally revisits the conventional insights acquired along the years on non-linear classification methods. As an aftermath, being optimal for Gaussian mixture inputs with common covariance, our study strongly suggests that the Softmax classifier is indeed *the optimal last layer of a deep neural network classifier*.

This claim however assumes a clear-cut separation between a back-end network training *isolated* from the front-end Softmax layer. A thorough validation of the equivalence between full network training and this divided approach would be necessary to confirm the claimed optimality and anticipate the performances of Softmax classification for an end-to-end deep neural network.

[9] For GAN images, these can be estimated accurately by drawing a large number of independent realizations, while for real images, the whole dataset is used to obtain empirical estimates.

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.

[Bri90] John S Bridle. Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition. In *Neurocomputing*, pages 227–236. Springer, 1990.

[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.

[CVMG+14] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[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.

[EKBB+13] Noureddine El Karoui, Derek Bean, Peter J Bickel, Chinghway Lim, and Bin Yu. On robust regression with high-dimensional predictors. *Proceedings of the National Academy of Sciences*, 110(36):14557–14562, 2013.

[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.

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

[GMH13] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. Speech recognition with deep recurrent neural networks. In *2013 IEEE international conference on acoustics, speech and signal processing*, pages 6645–6649. IEEE, 2013.

[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.

[HZRS16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[KFYA18] Sekitoshi Kanai, Yasuhiro Fujiwara, Yuki Yamanaka, and Shuichi Adachi. Sigsoftmax: Reanalysis of the softmax bottleneck. In *Advances in Neural Information Processing Systems*, pages 286–296, 2018.

[KG17] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? In *Advances in neural information processing systems*, pages 5574–5584, 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.

[LC18] Cosme Louart and Romain Couillet. Concentration of measure and large random matrices with an application to sample covariance matrices. *arXiv preprint arXiv:1805.08295*, 2018.

[LC20a] 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.

[LC20b] Cosme Louart and Romain Couillet. Concentration of solutions to random equations with concentration of measure hypotheses. *arXiv preprint*, 2020.

[LeC98] Yann LeCun. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998.

[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.

[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.

[LWYY16] Weiyang Liu, Yandong Wen, Zhiding Yu, and Meng Yang. Large-margin softmax loss for convolutional neural networks. In *ICML*, volume 2, page 7, 2016.

[MLC19] Xiaoyi Mai, Zhenyu Liao, and Romain Couillet. A large scale analysis of logistic regression: Asymptotic performance and new insights. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3357–3361. IEEE, 2019.

[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.

[RCY+19] Ankit Singh Rawat, Jiecao Chen, Felix Xinnan X Yu, Ananda Theertha Suresh, and Sanjiv Kumar. Sampled softmax with random fourier features. In *Advances in Neural Information Processing Systems*, pages 13857–13867, 2019.

[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.

[SVL14] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.

[TDBM20] Yi Tay, Mostafa Dehghani, Dara Bahri, and Donald Metzler. Efficient transformers: A survey. *arXiv preprint arXiv:2009.06732*, 2020.

[VSP+17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[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.