Random Matrix Theory Proves that Deep Learning Representations of GAN-data Behave as Gaussian Mixtures

ICML 2020

MEA. Seddik[12*], C.Louart[13], M. Tamaazousti[1], R. Couillet[23]

[1]CEA List, France
[2]CentraleSupélec, L2S, France
[3]GIPSA Lab Grenoble-Alpes University, France

[*]http://melaseddik.github.io/

June 8, 2020

Abstract

Context:

- Study of large **Gram** matrices of **concentrated** data.

Motivation:

- **Gram** matrices are at the core of various ML algorithms.
- RMT predicts their performances under **Gaussian** assumptions on the data.
- **BUT Real data** are **unlikely close** to **Gaussian** vectors.

Results:

- **GAN data ($\approx$ Real data)** fall within the class of **Concentrated** vectors.
- **Universality result:**

Only **first** and **second** order statistics of **Concentrated** data matter to describe the behavior of **Gram** matrices.

Notion of Concentrated Vectors

Definition (Concentrated Vectors)

Given a normed space $(E, \|\cdot\|_E)$ and $q \in \mathbb{R}$, a random vector $\boldsymbol{Z} \in E$ is q-exponentially **concentrated** if for any 1-**Lipschitz**[1] function $\mathcal{F} : E \to \mathbb{R}$, there exists $C, c > 0$ such that

$$\forall t > 0,\ \mathbb{P}\left\{|\mathcal{F}(\boldsymbol{Z}) - \mathbb{E}\mathcal{F}(\boldsymbol{Z})| \geq t\right\} \leq Ce^{-(t/c)^q} \xrightarrow{\text{denoted}} \boxed{\boldsymbol{Z} \in \mathcal{E}_q(c)}$$

If c independent of $\dim(E)$, we denote $\boxed{\boldsymbol{Z} \in \mathcal{E}_q(1)}$

Concentrated vectors enjoy:

(P1) If $\boldsymbol{X} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}_p)$ then $\boldsymbol{X} \in \mathcal{E}_2(1)$

"Gaussian vectors are concentrated vectors"

(P2) If $\boldsymbol{X} \in \mathcal{E}_q(1)$ and $\mathcal{G}$ is a $\lambda_{\mathcal{G}}$-**Lipschitz** map, then $\mathcal{G}(\boldsymbol{X}) \in \mathcal{E}_q(\lambda_{\mathcal{G}})$

"Concentrated vectors are stable through Lipschitz maps"

---

[1] **Reminder:** $\mathcal{F} : E \to F$ is $\lambda_{\mathcal{F}}$-Lipschitz if $\forall (\boldsymbol{x}, \boldsymbol{y}) \in E^2 : \|\mathcal{F}(\boldsymbol{x}) - \mathcal{F}(\boldsymbol{y})\|_F \leq \lambda_{\mathcal{F}} \|\boldsymbol{x} - \boldsymbol{y}\|_E$.

Why Concentrated Vectors?

Figure: Images artificially generated using the BigGAN model [Brock *et al*, ICLR'19].

$$\textbf{Real Data} \approx \textbf{GAN Data} = \underbrace{\mathcal{F}_L \circ \mathcal{F}_{L-1} \circ \cdots \circ \mathcal{F}_1}_{\mathcal{G}}(\textbf{Gaussian})$$

where the $\mathcal{F}_i$'s correspond to Fully Connected layers, Convolutional layers, Sub-sampling, Pooling and activation functions, residual connections or Batch Normalisation.

$\Rightarrow$ The $\mathcal{F}_i$'s are essentially ***Lipschitz*** operations.

Why Concentrated Vectors?

- **Fully Connected Layers and Convolutional Layers** are affine operations:

$$\mathcal{F}_i(\boldsymbol{x}) = \boldsymbol{W}_i \boldsymbol{x} + \boldsymbol{b}_i,$$

  and $\|\mathcal{F}_i\|_{lip} = \sup_{\boldsymbol{u}\neq \boldsymbol{0}} \frac{\|\boldsymbol{W}_i \boldsymbol{u}\|_p}{\|\boldsymbol{u}\|_p}$, for any p-norm.
- **Pooling Layers and Activation Functions:** Are 1-Lipschitz operations with respect to any p-norm (e.g., ReLU and Max-pooling).
- **Residual Connections:** $\mathcal{F}_i(\boldsymbol{x}) = \boldsymbol{x} + \mathcal{F}_i^{(\ell)} \circ \cdots \circ \mathcal{F}_i^{(1)}(\boldsymbol{x})$
  where the $\mathcal{F}_i^{(j)}$'s are Lipschitz operations, thus $\mathcal{F}_i$ is a Lipschitz operation with Lipschitz constant bounded by $1 + \prod_{j=1}^{\ell} \|\mathcal{F}_i^{(j)}\|_{lip}$.
- ...

By:

(P1) If $\boldsymbol{X} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_p)$ then $\boldsymbol{X} \in \mathcal{E}_2(1)$

(P2) If $\boldsymbol{X} \in \mathcal{E}_q(1)$ and $\mathcal{G}$ is a $\lambda_{\mathcal{G}}$-**Lipschitz** map, then $\mathcal{G}(\boldsymbol{X}) \in \mathcal{E}_q(\lambda_{\mathcal{G}})$

$\Rightarrow$ **GAN data** are **concentrated** vectors by design.

Remark: Still we need to control $\lambda_{\mathcal{G}}$.

Control of $\lambda_{\mathcal{G}}$ with Spectral Normalization

Let $\sigma_* > 0$ and $\mathcal{G}$ be a neural network composed of N affine layers, each one of input dimension d_{i-1} and output dimension d_i for $i \in [N]$, with 1-Lipschitz activation functions. Consider the following dynamics with learning rate η:

$$\boldsymbol{W} \leftarrow \boldsymbol{W} - \eta \boldsymbol{E}, \text{ with } \boldsymbol{E}_{i,j} \sim \mathcal{N}(0,1)$$
$$\boldsymbol{W} \leftarrow \boldsymbol{W} - \max(0, \sigma_1(\boldsymbol{W}) - \sigma_*)\, \boldsymbol{u}_1(\boldsymbol{W}) \boldsymbol{v}_1(\boldsymbol{W})^\intercal.$$

The Lipschitz constant of $\mathcal{G}$ is bounded at convergence with high probability as:

$$\lambda_{\mathcal{G}} \leq \prod_{i=1}^{N} \left(\varepsilon + \sqrt{\sigma_*^2 + \eta^2 d_i d_{i-1}} \right).$$

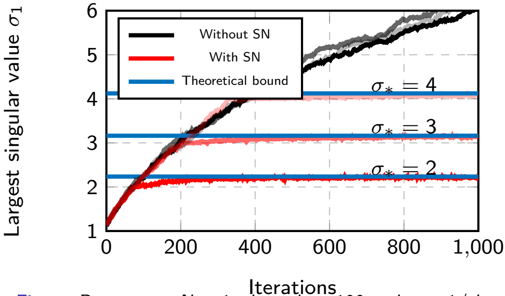


Figure: Parameters $N = 1$, $d_0 = d_1 = 100$ and $\eta = 1/d_0$.

Model & Assumptions

(A1) Data matrix (distributed in k classes $\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_k$):

$$\boldsymbol{X} = \left[\underbrace{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{n_1}}_{\in \mathcal{E}_{q_1}(1)}, \underbrace{\boldsymbol{x}_{n_1+1}, \ldots, \boldsymbol{x}_{n_2}}_{\in \mathcal{E}_{q_2}(1)}, \ldots, \underbrace{\boldsymbol{x}_{n-n_k+1}, \ldots, \boldsymbol{x}_n}_{\in \mathcal{E}_{q_k}(1)} \right] \in \mathbb{R}^{p \times n}$$

Model statistics: $\boldsymbol{\mu}_\ell = \mathbb{E}_{\boldsymbol{x}_i \in \mathcal{C}_\ell}[\boldsymbol{x}_i], \quad \boldsymbol{C}_\ell = \mathbb{E}_{\boldsymbol{x}_i \in \mathcal{C}_\ell}[\boldsymbol{x}_i \boldsymbol{x}_i^\intercal]$

(A2) Growth rate assumptions: As $p \to \infty$,

1. $p/n \to c \in (0, \infty)$.
2. The number of classers k is bounded.
3. For any $\ell \in [k]$, $\|\boldsymbol{\mu}_\ell\| = \mathcal{O}(\sqrt{p})$.

Gram matrix and its resolvent:

$$\boxed{\boldsymbol{G} = \frac{1}{p}\boldsymbol{X}^\intercal \boldsymbol{X}, \quad \boldsymbol{Q}(z) = (\boldsymbol{G} + z\boldsymbol{I}_n)^{-1}}$$

$$m_L(z) = \frac{1}{n}\operatorname{tr}(\boldsymbol{Q}(-z)), \quad \boldsymbol{U}\boldsymbol{U}^\intercal = \frac{-1}{2\pi i}\oint_\gamma \boldsymbol{Q}(-z)dz$$

Theorem

*Under Assumptions **(A1)** and **(A2)**, we have* $\boldsymbol{Q}(z) \in \mathcal{E}_q(p^{-\frac{1}{2}})$. *Furthermore,*

$$\left\|\mathbb{E}[\boldsymbol{Q}(z)] - \tilde{\boldsymbol{Q}}(z)\right\| = \mathcal{O}\left(\sqrt{\frac{\log p}{p}}\right) \text{ where } \tilde{\boldsymbol{Q}}(z) = \frac{1}{z}\boldsymbol{\Lambda}(z) + \frac{1}{p\,z}\boldsymbol{J}\Omega(z)\boldsymbol{J}^\intercal$$

with $\boldsymbol{\Lambda}(z) = diag\left\{\frac{\mathbf{1}_{n_\ell}}{1+\delta_\ell(z)}\right\}_{\ell=1}^k$ *and* $\Omega(z) = diag\{\boldsymbol{\mu}_\ell^\intercal \tilde{\boldsymbol{R}}(z)\boldsymbol{\mu}_\ell\}_{\ell=1}^k$

$$\tilde{\boldsymbol{R}}(z) = \left(\frac{1}{k}\sum_{\ell=1}^k \frac{\boldsymbol{C}_\ell}{1+\delta_\ell(z)} + z\boldsymbol{I}_p\right)^{-1}$$

with $\delta(z) = [\delta_1(z), \ldots, \delta_k(z)]$ *is the unique fixed point of the system of equations*

$$\delta_\ell(z) = tr\left(\boldsymbol{C}_\ell\left(\frac{1}{k}\sum_{j=1}^k \frac{\boldsymbol{C}_j}{1+\delta_j(z)} + z\boldsymbol{I}_p\right)^{-1}\right) \text{ for each } \ell \in [k].$$

Main Result

Theorem

Under Assumptions ***(A1)*** *and* ***(A2)****, we have* $\boldsymbol{Q}(z) \in \mathcal{E}_q(p^{-\frac{1}{2}})$. *Furthermore,*

$$\left\| \mathbb{E}[\boldsymbol{Q}(z)] - \tilde{\boldsymbol{Q}}(z) \right\| = \mathcal{O}\left(\sqrt{\frac{\log p}{p}}\right) \quad \text{where } \tilde{\boldsymbol{Q}}(z) = \frac{1}{z}\boldsymbol{\Lambda}(z) + \frac{1}{p\,z}\boldsymbol{J}\Omega(z)\boldsymbol{J}^\intercal$$

with $\boldsymbol{\Lambda}(z) = diag\left\{\frac{\mathbf{1}_{n_\ell}}{1+\delta_\ell(z)}\right\}_{\ell=1}^{k}$ *and* $\Omega(z) = diag\{\boldsymbol{\mu}_\ell{}^\intercal \tilde{\boldsymbol{R}}(z)\boldsymbol{\mu}_\ell\}_{\ell=1}^{k}$

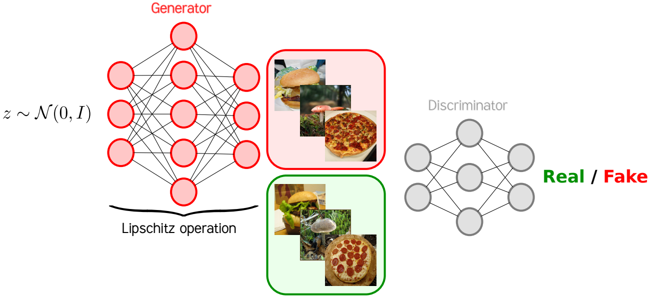
$$\tilde{\boldsymbol{R}}(z) = \left(\frac{1}{k}\sum_{\ell=1}^{k}\frac{\boldsymbol{C}_\ell}{1+\delta_\ell(z)} + z\boldsymbol{I}_p\right)^{-1}$$

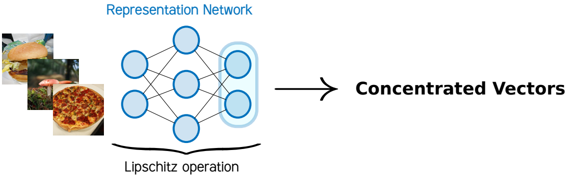
with $\delta(z) = [\delta_1(z), \ldots, \delta_k(z)]$ *is the unique fixed point of the system of equations*

$$\delta_\ell(z) = tr\left(\boldsymbol{C}_\ell\left(\frac{1}{k}\sum_{j=1}^{k}\frac{\boldsymbol{C}_j}{1+\delta_j(z)} + z\boldsymbol{I}_p\right)^{-1}\right) \quad \text{for each } \ell \in [k].$$

Key Observation: Only **first** and **second** order statistics matter!

Application to CNN Representations of GAN Images



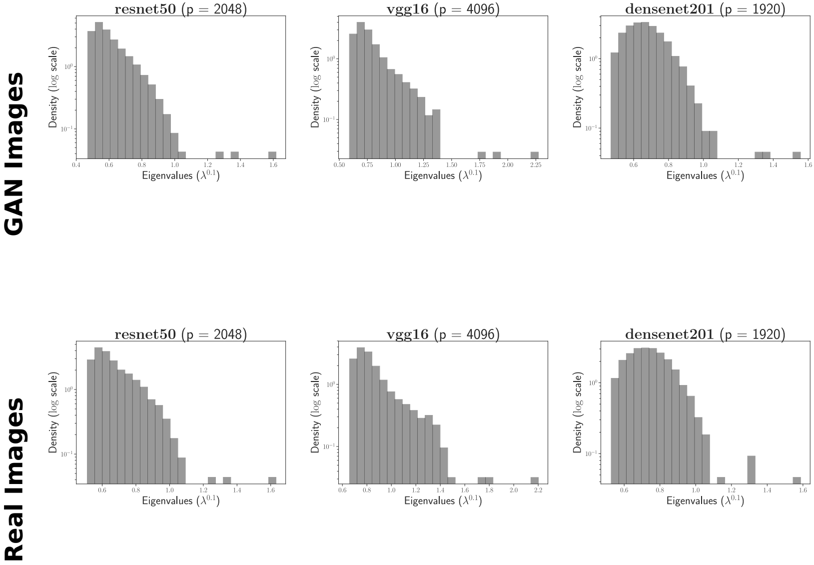


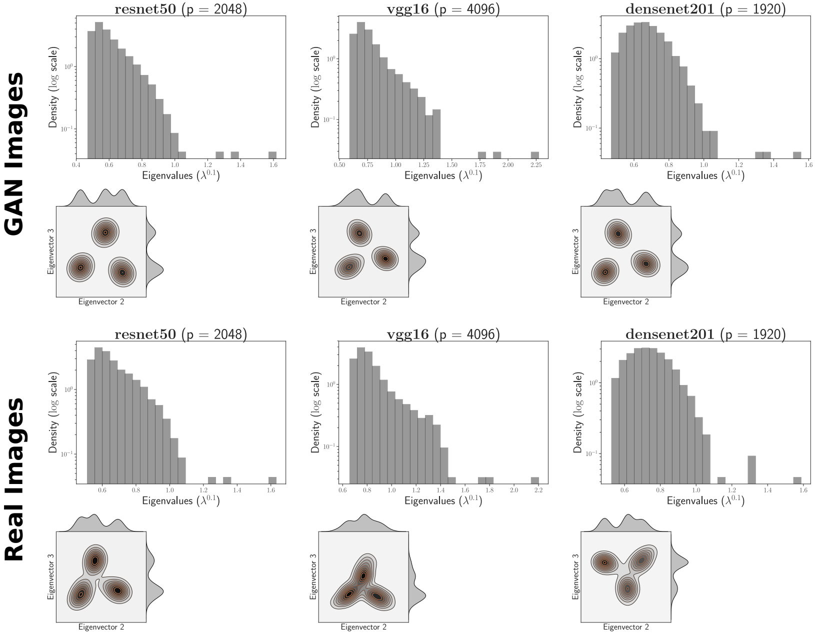
- CNN representations correspond to the **penultimate** layer.
- Popular architectures considered in practice are: **Resnet, VGG, Densenet**.

Application to CNN Representations of GAN Images

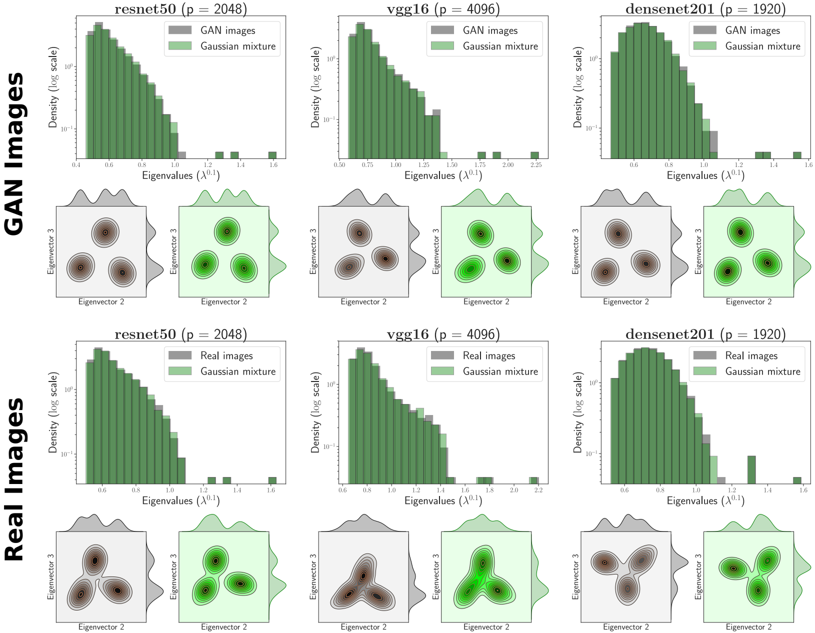
Figure: $k = 3$ classes, $n = 3000$ images.

Application to CNN Representations of GAN Images

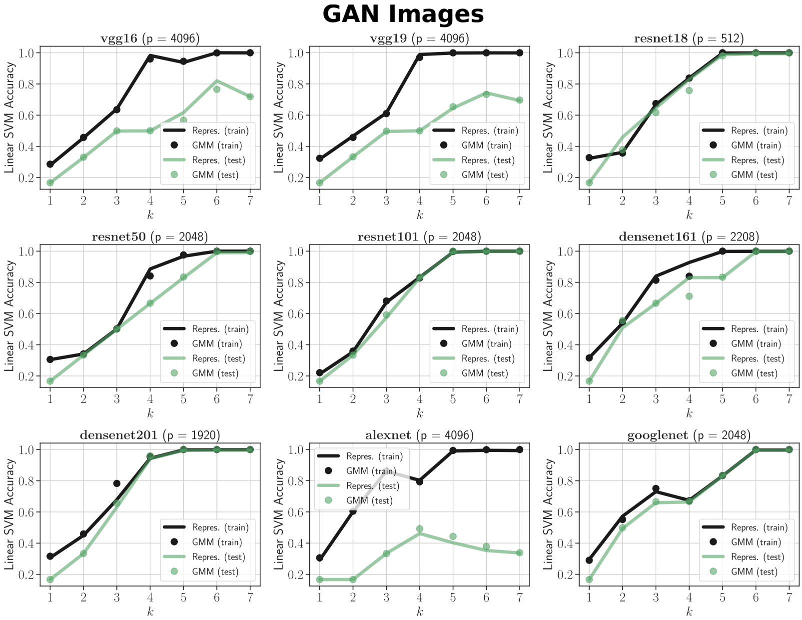



GAN Images
Real Images
resnet50 (p = 2048)
vgg16 (p = 4096)
densenet201 (p = 1920)
Density (log scale)
Eigenvalues ($\lambda^{0.1}$)
Eigenvector 2
Eigenvector 3

Application to CNN Representations of GAN Images

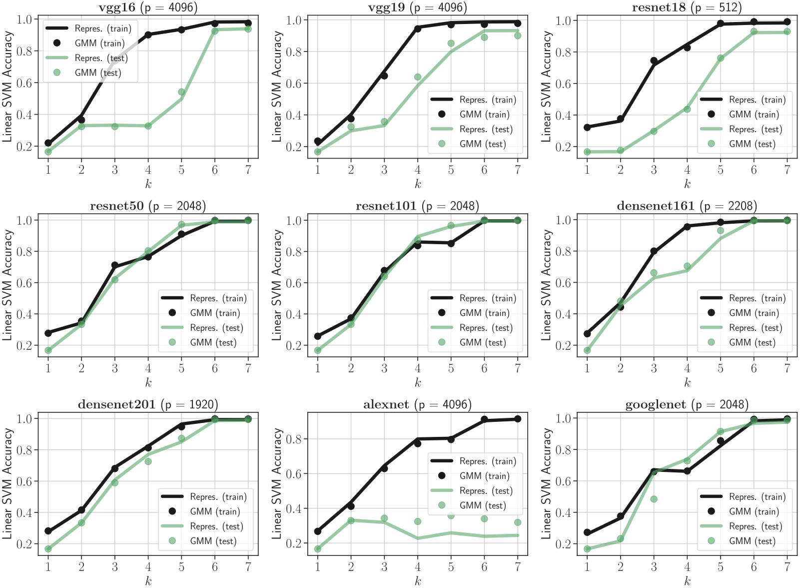


Performance of a linear SVM classifier



Performance of a linear SVM classifier

Real Images



Take away messages

- **Concentrated Vectors** seem appropriate for realistic data modelling.
- ***Universality*** of linear classifiers regardless of the data distribution.
- RMT can ***anticipate*** the performances of standard classifiers for DL representations of GAN images.
- Universality supports the **Gaussianity** assumption on the data representations as considered in the literature, e.g., the FID metric

$$d^2((\boldsymbol{\mu}, \boldsymbol{C}), (\boldsymbol{\mu}_w, \boldsymbol{C}_w)) = \|\boldsymbol{\mu} - \boldsymbol{\mu}_w\|^2 + \operatorname{tr}\left(\boldsymbol{C} + \boldsymbol{C}_w - 2(\boldsymbol{C}\boldsymbol{C}_w)^{\frac{1}{2}}\right).$$