On $1/n$ neural representation and robustness

Josue Nassar[*] **Piotr Aleksander Sokół**[*]
Stony Brook University
{josue.nassar, piotr.sokol}@stonybrook.edu

SueYeon Chung
Columbia University
sueyeonchung@gmail.com

Kenneth Harris
University College London
kenneth.harris@ucl.ac.uk

Il Memming Park
Stony Brook University
memming.park@stonybrook.edu

Abstract

Understanding the nature of representation in neural networks is a goal shared by neuroscience and machine learning. It is therefore exciting that both fields converge not only on shared questions but also similar approaches. A pressing question in these areas is understanding how the structure of the representation used by neural networks affects both their generalization, and robustness to perturbations. In this work, we investigate the latter by juxtaposing experimental results regarding the covariance spectrum of neural representations in the mouse V1 (Stringer et al) with artificial neural networks. We use adversarial robustness to probe Stringer et al's theory regarding the causal role of a 1/n covariance spectrum. We empirically investigate the benefits such a neural code confers in neural networks, and illuminate its role in multi-layer architectures. Our results show that imposing the experimentally observed structure on artificial neural networks makes them more robust to adversarial attacks. Moreover, our findings complement the existing theory relating wide neural networks to kernel methods, by showing the role of intermediate representations.

1 Introduction

Artificial neural networks and theoretical neuroscience have a shared ancestry of models they use and develop, this includes the McCulloch-Pitts model [30], Boltzmann machines [1] and convolutional neural networks [13, 28]. The relation between the disciplines, however, goes beyond the use of cognate mathematical models and includes a diverse set of shared interests – importantly, the overlap in interests increased as more theoretical questions came to the fore in deep learning. As such, the two disciplines have settled on similar questions about the nature of 'representations' or neural codes: how they develop during learning; how they enable generalization to new data and new tasks; their dimensionality and embedding structure; what role attention plays in their modulation; how their properties guard against illusions and adversarial examples.

Central to all these questions, is the exact nature of representations emergent in both artificial and biological neural networks. Even though relatively little is known about either, the known differences between both offer a point of comparison, that can potentially give us deeper insight into the properties of different neural codes, and their mechanistic role in giving rise to some of the observed properties. Perhaps the most prominent example of the difference between artificial and biological neural networks is the existence of adversarial examples [18, 20, 22, 41]– arguably they are of interest primarily not because of their genericity [11, 22], but because they expose the stark difference between computer vision algorithms and human perception.

Preprint. Under review.

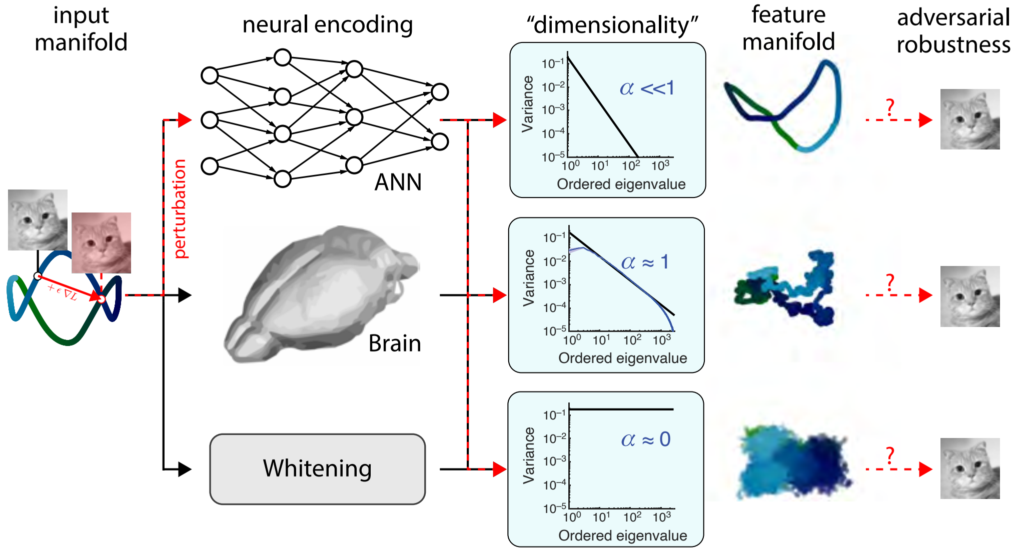

Figure 1: Stuyding the benefits of the spectrum of neural code for adversarial robustness. Neural code of the biological brain shows $1/n$ power-law spectrum and also robust. Meaningful manifolds in the input space can gain nonlinear features or lose its structure depending on the power-law exponent α. Using artificial neural networks and statistical whitening, we investigate how the "dimensionality", controlled by α, of the neural code impacts its robustness.

In this work, we use adversarial robustness to probe ideas regarding the 'dimensionality' of neural representations. The neuroscience community has advanced several normative theories, which include optimal coding [3], sparse coding [14, 32], as well a host of experimental data and statistical models [7, 15, 17, 19, 39] – resulting in, often conflicting, arguments in support of the prevalence of both low-dimensional and high-dimensional neural codes. By comparison, the machine learning community inspected the properties of hidden unit representations through the lens of statistical learning theory [12, 31, 43], information theory [37, 42], and mean-field and kernel methods [10, 25, 33]. The last two, by considering the limiting behavior as the number of neurons per layer goes to infinity, have been particularly successful, allowing analytical treatment of optimization, and generalization [2, 5, 6, 25].

Paralleling this development a recent study has recorded from a large number of mouse early visual area (V1) neurons [40]. The statistical analysis of the data leveraged kernel methods, much like the mean-field methods mentioned above, and has revealed that the covariance between neurons (marginalized over input) had a spectrum that decayed as $1/n$ power-law regardless of the input image statistics. To provide a potential rationale for the representation to be poised between low and high dimensionality, the authors developed a corresponding theory that relates the spectrum of the neural repertoire to the continuity and mean-square differentiability of a manifold in the input space [40]. Even with this proposed theory, the mechanistic role of the $1/n$ neural code is not known, but Stringer et al. [40] conjectured that it strikes a balance between expressivity and robustness. Moreover, the proposed theory only investigated the relationship between the input and output of the neural network; as many neural networks involve multiple layers, it is not clear how the neural code used by the intermediate layers affects the network as a whole. Similarly existing literature that relates the spectra of kernels to their out-of-sample generalization implicitly treats multi-layer architectures as shallow ones [4, 5, 8]. It is therefore desirable that a comprehensive theory ought to explain the role that the covariance spectrum plays at each layer of a neural network.

In this work, we empirically investigate the advantages of an $1/n$ neural code, by enforcing the spectral properties of the biological visual system in artificial neural networks. To this end we propose a spectral regularizer to enforce a $1/n$ eigenspectrum.
With these spectrally-regularized models in hand, we aim to answer the following questions :

- Does having an $1/n$ neural code make the network more robust to adversarial attacks?

Piotr
Should we maybe bold his part to visually draw attention

- For multi-layer networks, how does the neural code employed by the intermediate layers affect the robustness of the network?

The paper is organized as follows: we first provide a brief review of the empirical and theoretical results of Stringer et al. [40], followed by background information on deep neural networks. We then propose a spectral regularization technique and employ it in a number of empirical experiments to investigate the role of the $1/n$ neural code.

2 Background: $1/n$ Neural Representation in Mouse Visual Cortex

In this section, we briefly recap results from Stringer et al. [40]. To empirically investigate the neural representation utilized in mouse visual cortex, the authors recorded neural activity from $\sim 10{,}000$ neurons while the animal was presented with large sets of images, including $2,800$ images from the ImageNet [36]. They observed that the eigenspectra of the estimated covariance matrix of the neural activity follow a power-law, i.e. $\lambda_n \propto n^{-\alpha}$ regardless of the input statistics with a universal exponent of $\alpha \approx 1$.

The authors put forward a corresponding theory as a potential rationale for the existence of the $1/n$ spectra; importantly, the properties of the representation were investigated in the asymptotic regime of the number of neurons tending to infinity. Let $\boldsymbol{s} \in \mathbb{R}^{d_s}$ be an input to a network, where $p(\boldsymbol{s})$ is supported on a manifold of dimension $d \leq d_s$, and let $\boldsymbol{x} \in \mathbb{R}^N$ be the corresponding neural representation formed by a nonlinear encoding f, i.e. $\boldsymbol{x} = f(\boldsymbol{s})$. Let $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_N$ be the ordered eigenvalues of $\operatorname{cov}(f(\boldsymbol{s}))$. Under these assumptions Stringer et al. [40] proved that as $N \to \infty$, λ_n must decay faster than $n^{-\alpha}$ where $\alpha = 1 + 2/d$ for f to be continuous and differentiable, i.e. f locally preserves the manifold structure.

While continuity and differentiability are desirable properties, the advantages of a representation with eigenspectrum decaying slightly faster than n^{-1} is not apparent. Moreover, the theory abstracts away the intermediate layers of the network, focusing on the properties of f but not its constituents in a multi-layered architecture where $f = f_1 \circ \cdots \circ f_D$. To investigate further, we turn to deep neural networks as a test bed.

3 Spectrally regularized Deep Neural Networks

Consider a feed-forward neural network with input, $\boldsymbol{s} \in \mathbb{R}^{d_s}$, D layers of weights, $\mathbf{W}^1, \ldots, \mathbf{W}^D$, biases, $\boldsymbol{b}^1, \ldots, \boldsymbol{b}^D$, and D layers of neural activity, $\boldsymbol{x}^1, \ldots, \boldsymbol{x}^D$, where $\boldsymbol{x}^l \in \mathbb{R}^{N_l}$, $\mathbf{W}^l \in \mathbb{R}^{N_l \times N_{l-1}}$, and $\boldsymbol{b}^l \in \mathbb{R}^{N_l}$. The neural activity is recursively defined as,

$$\boldsymbol{x}^l = f_l(\boldsymbol{x}^{l-1}) \coloneqq \phi\left(\mathbf{W}^l \boldsymbol{x}^{l-1} + \boldsymbol{b}^l\right), \quad \text{for } l = 1, \cdots, D, \tag{1}$$

where $\boldsymbol{x}^0 = \boldsymbol{s}$, $\phi(\cdot)$ is an element-wise non-linearity and $\boldsymbol{b}^l \in \mathbb{R}^{N_l}$ is a bias term. We define the mean and covariance of the neural activity at layer l as $\boldsymbol{\mu}^l = \mathbb{E}[\boldsymbol{x}^l]$ and $\boldsymbol{\Sigma}^l = \mathbb{E}[(\boldsymbol{x}^l - \boldsymbol{\mu}^l)(\boldsymbol{x}^l - \boldsymbol{\mu}^l)^\top]$, respectively. Note that the expectation marginalizes over the input distribution, $p(\boldsymbol{s})$, which we assume has finite mean and variance, i.e. $\boldsymbol{\mu}^0 < \infty$ and $\operatorname{Tr}(\boldsymbol{\Sigma}^0) < \infty$, where $\operatorname{Tr}(\cdot)$ is the trace operator.

To analyze the neural representation of layer l we examine the eigenspectrum of its covariance matrix, $\boldsymbol{\Sigma}^l$, denoted by the ordered eigenvalues $\lambda_1^l \geq \lambda_2^l \geq \cdots \geq \lambda_{N_l}^l$, and corresponding eigenvectors $v_1^l, \ldots, v_{N_l}^l$. While the theory developed by Stringer et al. [40] dealt with infinitely wide networks, in reality both biological and artificial neural networks are of finite width; although, empirical evidence has shown that the consequences of infinitely wide networks are still felt by finite-width networks that are sufficiently wide [25, 33].

3.1 Spectral regularizer

In general, the distribution of eigenvaues of a deep neural network (DNN) is intractable and a priori there is no reason to believe it should follow a power-law — indeed, it will be determined by architectural choices, such as initial weight distribution and non-linearity, but also by the entire trajectory in parameter space traversed during optimization [24, 33]. A simple way to enforce a power-law decay without changing its architecture is to use the finite-dimensional embedding and directly regularize the eigenspectrum of the neural representation used at layer l. To this end we introduce the following regularizer:

$$R_l(\lambda_1^l, \ldots, \lambda_{N_l}^l) = \frac{\beta}{N_l} \sum_{n \geq \tau}^{N_l} \Big((\lambda_n^l / \gamma_n^l - 1)^2 + \max(0, \lambda_n^l / \gamma_n^l - 1) \Big), \tag{2}$$

where γ_n^l is the target spectra that follows a $n^{-\alpha_l}$ power-law, the cut-off τ dictates which eigenvalues should be regularized and β is a hyperparameter that controls the strength of the regularizer. To construct γ_n^l, we create a sequence of the form $\gamma_n^l = \kappa n^{-\alpha_l}$ where κ is chosen such that $\lambda_\tau^l = \gamma_\tau^l$. Since $\gamma_\tau^l = \lambda_\tau^l$, the ratio λ_n^l / γ_n^l for $n \geq \tau$ serves as a proxy measure to compare the rates of decay between the neural representation, λ_n^l, and the target spectra, γ_n^l. Leveraging this ratio, $(\lambda_n^l / \gamma_n^l - 1)^2$ penalizes the network for using neural representations that stray away from γ_n^l; we note that this term equally penalizes a ratio that is greater than or less than 1. Noting that having a slowly decaying spectrum, $\lambda_n^l / \gamma_n^l > 1$, leads to highly undesirable properties (viz. discontinuity and unbounded gradients in the infinite dimensional case), $\max(0, \lambda_n^l / \gamma_n^l - 1)$ is used to further penalize the network for having a spectrum that decays too slowly.

3.2 Training scheme

Naive use of (2) as a regularizer faces practical difficulty which comes from the need to estimate the eigenvalues of the covariance matrix for each layer l of each mini-batch, which has a computational complexity of $\mathcal{O}(N_l^3)$. Obtaining a reasonable estimate of λ_n^j also requires a batch size at least as large as the widest layer in the network [9]. While the second issue is unavoidable, we propose a work around for the first one.

Performing an eigenvalue decomposition of $\mathbf{\Sigma}^l$ gives

$$\mathbf{\Sigma}^l = \mathbf{V}_l \mathbf{\Lambda}^l \mathbf{V}_l^\top, \tag{3}$$

where $\mathbf{V}_l$ is an orthonormal matrix of eigenvectors and $\mathbf{\Lambda}^l$ is a diagonal matrix with the eigenvalues of $\mathbf{\Sigma}^l$ on the diagonal. Using $\mathbf{V}_l$ we can diagonalize $\mathbf{\Sigma}^l$ to obtain

$$\mathbf{V}_l^\top \mathbf{\Sigma}^l \mathbf{V}_l = \mathbf{\Lambda}^l. \tag{4}$$

It's evident from (4) that given the eigenvectors, we could easily obtain the eigenvalues. Thus, we propose the following approach: at the beginning of each epoch an eigenvalue decomposition is performed on the full training set and the eigenvectors, $\mathbf{V}_l$ for $l = 1, \cdots, D$, are stored. Next, for each mini-batch we construct $\mathbf{\Sigma}^l$ and compute

$$\mathbf{V}_l^\top \mathbf{\Sigma}^l \mathbf{V}_l = \hat{\mathbf{\Lambda}}^l, \tag{5}$$

and the diagonal elements of $\hat{\mathbf{\Lambda}}^l$ are taken as an approximation for the true eigenvalues and used to evaluate (2). This approach is correct in the limit of vanishingly small learning rates.

When using the regularizer for training, the approximate eigenvectors are fixed and gradients are not back-propagated through them. Similar to batch normalization [23], the gradients are back-propagated through the construction of the empirical covariance matrix, Σ^l.

4 Experiments

To empirically investigate the benefits of a power-law neural representation and of the proposed regularization scheme, we train a variety of models on MNIST [27]. Notably, while MNIST is considered a toy dataset for most computer vision tasks, it still is a good test-bed for the design of adversarial defenses [38].
Recall that the application of Stringer et al. [40]'s theory requires an estimate of the manifold dimension, d, which is not know of MNIST. However, we make the simplifying assumptions that is sufficiently large such that $1 + 2/d \approx 1$ holds. For this reason we set $\alpha_l = 1$ for all experiments. Based on the empirical observation of Stringer et al. [40], we set $\tau = 10$ as they observed that neural activity of mouse visual cortex followed a power-law approximately after the tenth eigenvalue. For all experiments three different values of β were tested, $\beta \in \{1, 2, 5\}$, and the results of the best one are shown in the main text; the results for all values of β are deferred to the appendix. A learning rate

of 10^{-4} was chosen to ensure stability of the proposed training scheme. All results shown are based on 3 experiments with different random seeds.

The adversarial robustness of the models are evaluated using two popular forms of threat models:

- Fast gradient sign method (FGSM): Given an input image, $\boldsymbol{s}_n$, and corresponding label, y_n, FGSM [20] produces an adversarial image, $\tilde{\boldsymbol{s}}_n$, by

$$\tilde{\boldsymbol{s}}_n = \boldsymbol{s}_n + \epsilon \, \text{sign}(\nabla_{\boldsymbol{s}_n} L(f(\boldsymbol{s}_n; \theta), y_n)), \tag{6}$$

  where $L(\cdot, \cdot)$ is the loss function and sign$(\cdot)$ is the sign function.

- Projected gradient descent (PGD): Given an input image, $\boldsymbol{s}_n$, and corresponding label, y_n, PGD [29] produces an adversarial image, $\tilde{\boldsymbol{s}}_n$, by solving the following optimization problem

$$\begin{aligned} &\arg\max_{\boldsymbol{\delta}} L(f(\boldsymbol{s}_n + \boldsymbol{\delta}; \boldsymbol{\theta}), y_n), \\ &\text{such that} \quad \|\boldsymbol{\delta}\|_\infty \leq \epsilon. \end{aligned} \tag{7}$$

  To approximately solve (7), we use 40 steps of projected gradient descent

$$\boldsymbol{\delta} \leftarrow \text{Proj}_{\|\boldsymbol{\delta}\|_\infty \leq \epsilon} \left(\boldsymbol{\delta} + \eta \nabla_{\boldsymbol{\delta}} L(f(\boldsymbol{s}_n + \boldsymbol{\delta}; \boldsymbol{\theta}), y_n) \right), \tag{8}$$

  where $\eta = 0.01$ and $\text{Proj}_{\|\boldsymbol{\delta}\|_\infty \leq \epsilon}(\cdot)$ is the projection operator.

To attempt to understand the features utilized by the neural representations, we visualize

$$J_n(x_j) = \frac{\partial \lambda_n^D}{\partial x_j}. \tag{9}$$

where λ_n^D is the nth eigenvalue of Σ^D. To compute (9), the full data set is passed through the network to obtain Σ^D; The eigenvlaues are then computed and gradients are back-propagated through the operation to obtain (9).

4.1 Shallow Neural Networks

To isolate the benefits of a $1/n$ neural representation, we begin by applying the proposed spectral regularizer on a sigmoidal neural network with one hidden layer of $N_1 = 2{,}000$ neurons with batch norm [23] (denoted by SpecReg) and examine its robustness to adversarial attacks. As a baseline, we compare it against a vanilla (unregularized) network.

Figure 2A, demonstrates the efficacy of the proposed training scheme as the spectra of the regularized network follows $1/n$ pretty closely. Figures 2B and C demonstrate that a spectrally regularized network is significantly more robust than its vanilla counterpart against **both** FGSM and PGD attacks. While the results are nowhere near SOTA, we emphasize that this robustness was gained *without training on a single adversarial image unlike other approaches*. The spectrally regularized network has a much higher effective dimension (Fig. 2A), and it learns a more relevant set of features as indicated by the sensitivity maps (Fig. 2D). We note that while the use of batch norm helped to increase the effectiveness of the spectral regularizer, it had no affect on the robustness of the vanilla network. In the interest of space, the results for the networks without bacth norm are in the appendix.

4.2 Deep Neural Networks

Inspired by the results in the previous section, we turn our attention to deep neural networks. Specifically, we experiment on a multi-layer percepton (MLP) with three hidden layers where $N_1 = N_2 = N_3 = 1{,}000$ and on a convolutional neural network (CNN) with three hidden layers: a convolutional layer with 16 output channels with a kernel size of (3, 3), followed by another convolutional layer with 32 output channels with a kernel of size (3, 3) and a fully-connected layer of width 1,000 neurons, where max-pooling is applied after each convolutional layer. To regularize the neural representation utilized by the convolutional layers, the output of all the channels are flattened together and the spectrum of their covariance matrix is regularized.

We demonstrate empirically the importance of intermediate layers in a deep neural network, as networks with "bad" intermediate representations are shown to be extremely brittle.

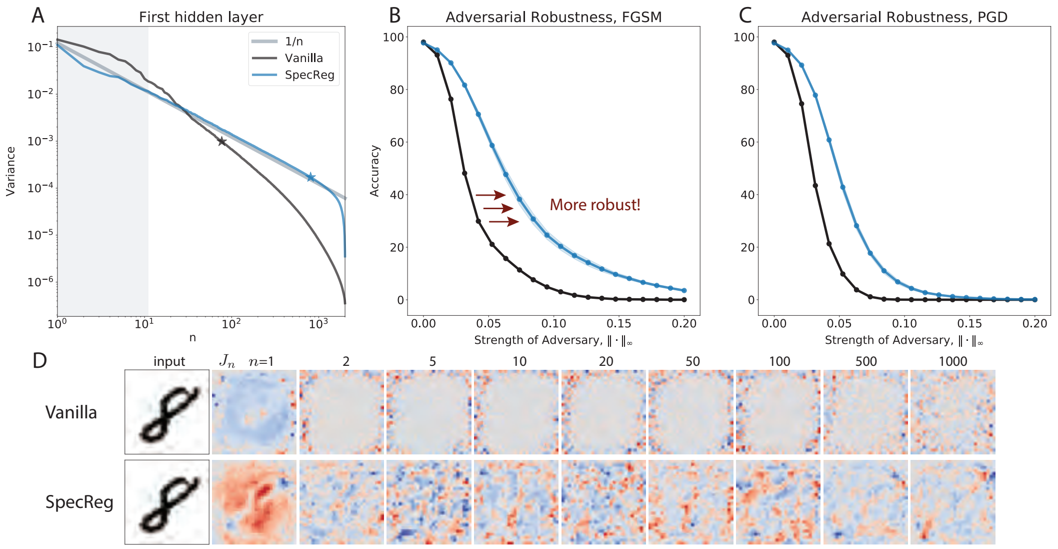

Figure 2: A $1/n$ neural representation leads to more robust networks. A) Eigenspectrum of a regularly trained network, Vanilla, and a spectrally regularized network, SpecReg. The shaded grey area are the eigenvalues that are not regularized. Star indicates the dimension at which the cumulative variance exceeds 90%. B & C) Comparison of adversarial robustness between vanilla and spectrally regularized networks where the shaded region is ± 1 standard deviation computed over 3 random seeds. D) The sensitivity of λ_n with respect to the input image.

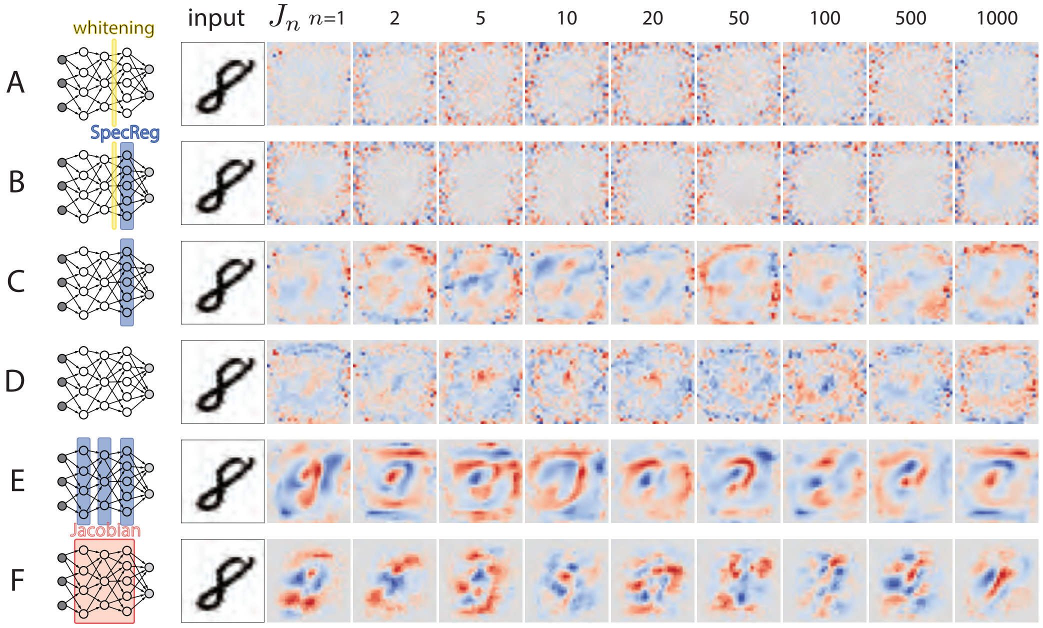

Figure 3: Sensitivity maps for 3-layer multi-layer perceptrons. Each row corresponds to a different experiment: (A) MLP with whitening layer, (B) SpecReg only on the last layer after whitening, (C) SpecReg only on the last layer, (D) Vanilla MLP, (E) SpecReg on every layer, (F) Jacobian regularization. Each sensitivity image corresponds to the n-th eigenvalue on the last hidden layer.

4.2.1 The Importance of Intermediate Layers in Deep Neural Networks

Theoretical insights from Stringer et al. [40] as well as other works [5, 24] do not prescribe how the spectrum of intermediate layers should behave for a "good" neural representation. Moreover, it is not clear how the neural representation employed by the intermediate layers will affect the robustness of the overall network. The theory of Stringer et al. [40] suggests that the intermediate layers of the network do not matter as long as the neural representation of the last hidden layer is $1/n$.

To investigate the importance of the intermediate layers, we "break" the neural representation at the second hidden layer by whitening its neural activity

$$\tilde{\boldsymbol{x}}^2 = \boldsymbol{R}_2^{-1}(\boldsymbol{x}^2 - \boldsymbol{\mu}^2) \tag{10}$$

where $\boldsymbol{R}_2$ is the Cholesky decomposition of $\boldsymbol{\Sigma}^2$, leading to a flat spectrum which is the worst case scenario under the asymptotic theory in Stringer et al. [40]. To compute (10), the sample mean, $\hat{\boldsymbol{\mu}}^2$ and covariance, $\hat{\boldsymbol{\Sigma}}^2$ are computed for each batch. Training with the whitening operation is handled similarly to batch norm [23], where gradients are back-propagated through the sample mean, covariance and Cholesky decomposition.

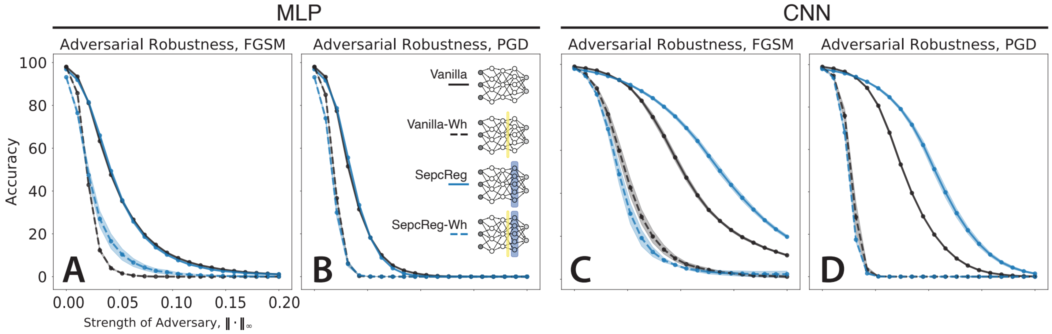

Figure 4: Spectral regularization does not rescue robustness lost by whitening of the intermediate representation. (A,B) For MLP, adversarial robustness for FGSM and PGD indicates more fragile code resulting from whitening. Spectral regularization of the last hidden layer does not improve robustness. (C,D) For CNN, spectral regularization of the last layer enhances robustness, but otherwise consistent trend with MLP. Same conventions as Fig. 2B,C and Fig. 3.

When the intermediate layer is whitened, the resulting sensitivity features loses structure (Fig. 3A compared to Fig. 3D). This network is less robust to adversarial attack (Fig. 4A,B dashed black) consistent with the structureless sensitivity map.

We added spectral regularization to the last hidden layer in hopes of salvaging the network (see Sec. 3.2 for details). Although the resulting spectrum of the last layer show $1/n$ tail (Fig. A7), the robustness is not improved (Fig. 4A,B). Applying the asymptotic theory to the whitened output, the neural representation is "bad" and the manifold becomes nondifferentiable. Spectral regularization of the last hidden layer cannot further fix this broken representation even in a finite sized network (Fig. 3B).

Interestingly, for the MLP, regularizing only the hidden layer (without whitening) improved the sensitivity map (Fig. 3C) but had no effect on the robustness of the network (Fig. 4A,B), suggesting that a $1/n$ neural representation at the last hidden layer is not sufficient. In contrast, regularizing the last hidden layer of the CNN does marginally increase the robustness of the network (Fig. 4C,D).

4.3 Regularizing all layers of the network increases adversarial robustness

The previous section showcased the importance of the intermediate representations in the network. The MLP results showcased that regularizing just the last hidden layer is not enough. Thus, we investigate the effect of spectrally regularizing all the hidden layers in the network. As an additional comparison, we train networks whose Jacobian is regularized [21, 26, 35]

$$\frac{\beta_j}{B} \sum_{n=1}^{B} \left\| \frac{\partial L(f(\boldsymbol{s}_n;\theta), y_n)}{\partial \boldsymbol{s}_n} \right\|_F^2 \tag{11}$$

where $(\boldsymbol{s}_n, y_n)$ is the n^{th} training data point, B is the batch size and $\beta_j = 0.01$ We compare against this type of regularization as it is one of the few approaches that do not rely on training on adversarial examples nor does it require changing inputs, nor models. The spectra of the networks are in the appendix.

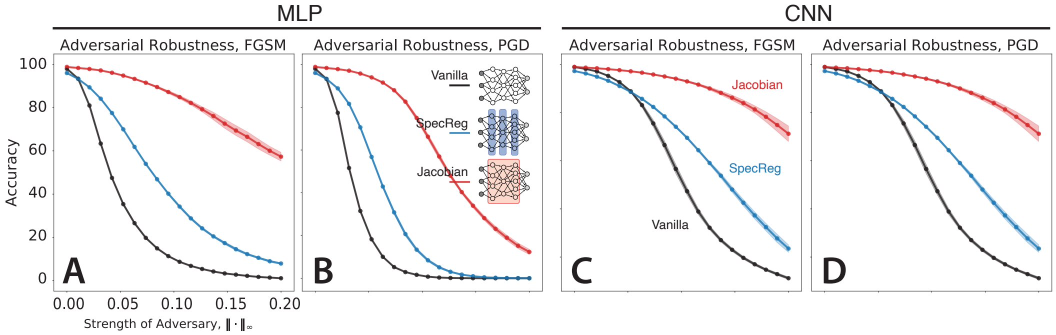

Figure 5: Spectral regularization of all layers improves robustness in 3-layer MLP and CNN. Same conventions as Fig. 4.

Regularizing all the layers of the MLP leads to an increase in robustness, as is evident in figure 5A,B. The neural representation induced by the regularizer favors *smoother*, more global features. This characteristic is shared both by our proposed regularization technique as well as Jacobian regularization, as can be seen in figure 3. Interestingly, these observations find counterpart in existing results on adversarial robustness in CNNs [44].

While the regularizer was able to enforce a $1/n$ decay for the last hidden layer of the network, its affect was less pronounced at earlier layers where it was observed that that the networks preferred to relax towards $1/n$ as opposed to every hidden layer having a $1/n$ neural representation. This suggests an alternative approach where we slowly decrease α over depth, $\alpha_1 \geq \alpha_2 \geq \cdots \alpha_D = 1$; We leave this for future work. While regularizing all the layers in the CNN leads to in increase in robustness figure 5C,D, the effect is comparable to just regularizing the last layer (Fig. 4C,D). We note that regularizing the spectra is inferior to regularizing the Jacobian in terms of adversarial robustness. Examining the spectra in figures A13, A16, we see that the neural representation utilized by the Jacobian regularized networks do not follow $1/n$. Thus, while networks whose neural representation have a $1/n$ eigenspectrum are robust, robust networks do not necessarily posses a $1/n$ neural representation.

5 Discussion

In this study, we trained artificial neural networks using a novel spectral regularizer to further understand the benefits and intricacies of a neural code possessing a $1/n$ eigenspectrum. We note that our current implementation of the spectral regularization is not intended to be used in general but rather a straightforward embodiment of the study objective. As the result suggests, a general encouragement of $1/n$-like spectrum could be beneficial in wide neural networks, and special architectures could be designed to more easily achieve this goal. The results have also helped to elucidate the importance of intermediate layers in DNNs and may offer a potential explanation for why batch normalization reduces the robustness of DNNs [16].

From a neuroscientific perspective, it is interesting to conjecture whether a similar power-law code with similar exponent is a hallmark of canonical cortical computation, or whether it reflects the unique specialization of lower visual areas. This latter question can be simultaneously addressed in-vivo and in-silico, with the latter experiments probing different exponents at higher layers in a deep neural network. Moreover, this curiously relates to the observed, but commonplace spectrum flattening for random deep neural networks [33], and questions about its effect on information propagation. We leave these question to future study.

Broader Impact

Adversarial attacks pose threats to the safe deployment of AI systems– both safety from malicious attacks but also robustness that would be expected in intelligent devices such as self-driving cars [34] but also facial recognition systems. Our neuroscience-inspired approach, unlike widely use adversarial attack defenses, does not require generation of adversarial input samples, therefore it potentially avoids the pitfalls of unrepresentative datasets. Furthermore, our study shows possibilities for the improvement of artificial systems with insights gained from biological systems which are naturally robust. Conversely it also provides a deeper, mechanistic understanding of experimental data from the field of neuroscience, thereby advancing both fields at the same time.

Acknowledgments and Disclosure of Funding

This work is supported by the generous support of Stony Brook Foundation's Discovery Award, NSF CAREER IIS-1845836, NSF IIS-1734910, NIH/NIBIB EB026946, and NIH/NINDS UF1NS115779.

References

[1] D. H. Ackley, G. E. Hinton, and T. J. Sejnowski. A Learning Algorithm for Boltzmann Machines. *Cognitive Science*, 9(1):147–169, 1985. doi: 10.1207/s15516709cog0901_7.

[2] Z. Allen-Zhu, Y. Li, and Y. Liang. Learning and Generalization in Overparameterized Neural Networks, Going Beyond Two Layers. *arXiv:1811.04918 [cs, math, stat]*, May 2019.

[3] H. B. Barlow. *Possible Principles Underlying the Transformations of Sensory Messages*. The MIT Press, 1953. ISBN 978-0-262-31421-3.

[4] P. L. Bartlett, O. Bousquet, and S. Mendelson. Local Rademacher complexities. *The Annals of Statistics*, 33(4):1497–1537, Aug. 2005. ISSN 0090-5364, 2168-8966. doi: 10.1214/009053605000000282.

[5] B. Bordelon, A. Canatar, and C. Pehlevan. Spectrum Dependent Learning Curves in Kernel Regression and Wide Neural Networks. *arXiv:2002.02561 [cs, stat]*, Feb. 2020.

[6] L. Chizat, E. Oyallon, and F. Bach. On Lazy Training in Differentiable Programming. In H. Wallach, H. Larochelle, A. Beygelzimer, F. Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 2937–2947. Curran Associates, Inc., 2019.

[7] M. M. Churchland, J. P. Cunningham, M. T. Kaufman, J. D. Foster, P. Nuyujukian, S. I. Ryu, and K. V. Shenoy. Neural population dynamics during reaching. *Nature*, 487(7405):51–56, July 2012. ISSN 0028-0836, 1476-4687. doi: 10.1038/nature11129.

[8] C. Cortes, M. Kloft, and M. Mohri. Learning kernels using local rademacher complexity. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 2760–2768. Curran Associates, Inc., 2013.

[9] R. Couillet and M. Debbah. *Random matrix methods for wireless communications*. Cambridge University Press, 2011.

[10] A. Daniely, R. Frostig, and Y. Singer. Toward Deeper Understanding of Neural Networks: The Power of Initialization and a Dual View on Expressivity. In D. D. Lee, M. Sugiyama, U. von Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 2253–2261, 2016.

[11] E. Dohmatob. Generalized No Free Lunch Theorem for Adversarial Robustness. In *International Conference on Machine Learning*, pages 1646–1654, May 2019.

[12] G. K. Dziugaite and D. M. Roy. Computing Nonvacuous Generalization Bounds for Deep (Stochastic) Neural Networks with Many More Parameters than Training Data. In G. Elidan, K. Kersting, and A. T. Ihler, editors, *Proceedings of the Thirty-Third Conference on Uncertainty in Artificial Intelligence, UAI 2017, Sydney, Australia, August 11-15, 2017*. AUAI Press, 2017.

[13] K. Fukushima. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, 36(4):193–202, Apr. 1980. ISSN 0340-1200, 1432-0770. doi: 10.1007/BF00344251.

[14] S. Fusi, E. K. Miller, and M. Rigotti. Why neurons mix: high dimensionality for higher cognition. *Current opinion in neurobiology*, 37:66–74, Apr. 2016. ISSN 0959-4388, 1873-6882. doi: 10.1016/j.conb.2016.01.010.

[15] J. A. Gallego, M. G. Perich, S. N. Naufel, C. Ethier, S. A. Solla, and L. E. Miller. Cortical population activity within a preserved neural manifold underlies multiple motor behaviors. *Nature Communications*, 9(1):1–13, Oct. 2018. ISSN 2041-1723. doi: 10.1038/s41467-018-06560-z.

[16] A. Galloway, A. Golubeva, T. Tanay, M. Moussa, and G. W. Taylor. Batch normalization is a cause of adversarial vulnerability. *arXiv preprint arXiv:1905.02161*, 2019.

[17] P. Gao and S. Ganguli. On simplicity and complexity in the brave new world of Large-Scale neuroscience. *Current opinion in neurobiology*, 32:148–155, Mar. 2015. ISSN 0959-4388. doi: 10.1016/j.conb.2015.04.003.

[18] R. Geirhos, P. Rubisch, C. Michaelis, M. Bethge, F. A. Wichmann, and W. Brendel. Imagenet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. In *International Conference on Learning Representations*, 2019.

[19] M. D. Golub, P. T. Sadtler, E. R. Oby, K. M. Quick, S. I. Ryu, E. C. Tyler-Kabara, A. P. Batista, S. M. Chase, and B. M. Yu. Learning by neural reassociation. *Nature neuroscience*, 21(4): 607–616, Apr. 2018. ISSN 1097-6256, 1546-1726. doi: 10.1038/s41593-018-0095-3.

[20] I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[21] J. Hoffman, D. A. Roberts, and S. Yaida. Robust learning with jacobian regularization. *arXiv preprint arXiv:1908.02729*, 2019.

[22] A. Ilyas, S. Santurkar, D. Tsipras, L. Engstrom, B. Tran, and A. Madry. Adversarial examples are not bugs, they are features. *arXiv preprint arXiv:1905.02175*, 2019.

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

[24] A. Jacot, F. Gabriel, and C. Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 8571–8580. Curran Associates, Inc., 2018.

[25] A. Jacot, F. Gabriel, and C. Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in neural information processing systems*, pages 8571–8580, 2018.

[26] D. Jakubovitz and R. Giryes. Improving DNN Robustness to Adversarial Attacks using Jacobian Regularization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 514–529, 2018.

[27] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[28] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521(7553):436–444, May 2015. ISSN 1476-4687. doi: 10.1038/nature14539.

[29] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018.

[30] W. S. McCulloch and W. Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133, Dec. 1943. ISSN 1522-9602. doi: 10.1007/BF02478259.

[31] B. Neyshabur, S. Bhojanapalli, D. Mcallester, and N. Srebro. Exploring Generalization in Deep Learning. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5947–5956. Curran Associates, Inc., 2017.

[32] B. A. Olshausen and D. J. Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381(6583):607–609, June 1996. ISSN 0028-0836. doi: 10.1038/381607a0.

[33] B. Poole, S. Lahiri, M. Raghu, J. Sohl-Dickstein, and S. Ganguli. Exponential expressivity in deep neural networks through transient chaos. In *Advances in neural information processing systems*, pages 3360–3368, 2016.

[34] A. Ranjan, J. Janai, A. Geiger, and M. J. Black. Attacking Optical Flow. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pages 2404–2413. IEEE, 2019. doi: 10.1109/ICCV.2019.00249.

[35] A. S. Ross. Improving the Adversarial Robustness and Interpretability of Deep Neural Networks by Regularizing Their Input Gradients. pages 1660–1669, 2018.

[36] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

[37] A. M. Saxe, Y. Bansal, J. Dapello, M. Advani, A. Kolchinsky, B. D. Tracey, and D. D. Cox. On the Information Bottleneck Theory of Deep Learning. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018.

[38] L. Schott, J. Rauber, M. Bethge, and W. Brendel. Towards the first adversarially robust neural network model on MNIST. In *International Conference on Learning Representations*, 2019.

[39] H. Sohn, D. Narain, N. Meirhaeghe, and M. Jazayeri. Bayesian computation through cortical latent dynamics. *Neuron*, 103(5):934–947.e5, Sept. 2019. ISSN 0896-6273, 1097-4199. doi: 10.1016/j.neuron.2019.06.012.

[40] C. Stringer, M. Pachitariu, N. Steinmetz, M. Carandini, and K. D. Harris. High-dimensional geometry of population responses in visual cortex. *Nature*, page 1, 2019.

[41] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

[42] N. Tishby and N. Zaslavsky. Deep learning and the information bottleneck principle. In *2015 IEEE Information Theory Workshop, ITW 2015, Jerusalem, Israel, April 26 - May 1, 2015*, pages 1–5. IEEE, 2015. doi: 10.1109/ITW.2015.7133169.

[43] V. N. Vapnik. *The Nature of Statistical Learning Theory*. Springer, 1st ed. 1995. corr. 2nd printing edition, Dec. 1998. ISBN 9780387945590.

[44] H. Wang, X. Wu, P. Yin, and E. P. Xing. High Frequency Component Helps Explain the Generalization of Convolutional Neural Networks. *arXiv:1905.13545 [cs]*, May 2019.

A Additional Figures for Section 4.1

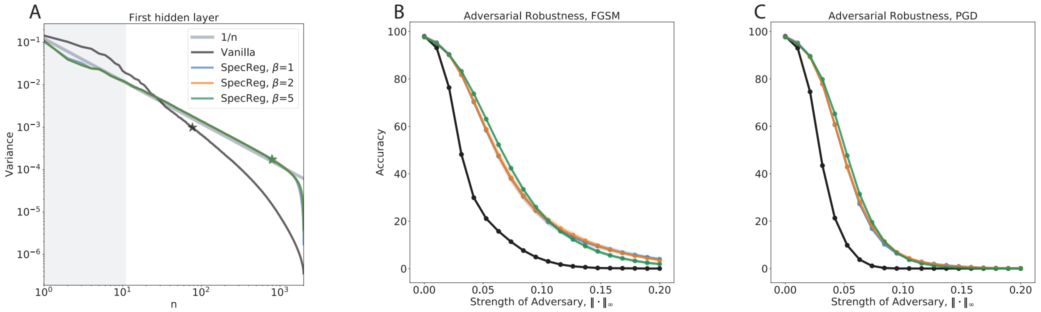

Figure A6: A) Eigenspectrum of a regularly trained network, Vanilla, and a spectrally regularized network, SpecReg, for various values of β. The shaded green area are the eigenvalues that are not regularized. B & C) Comparison of adversarial robustness between vanilla and spectrally regularized networks where the shaded region is ± 1 standard deviation.

B Additional Figures for Section 4.2

B.1 MLP

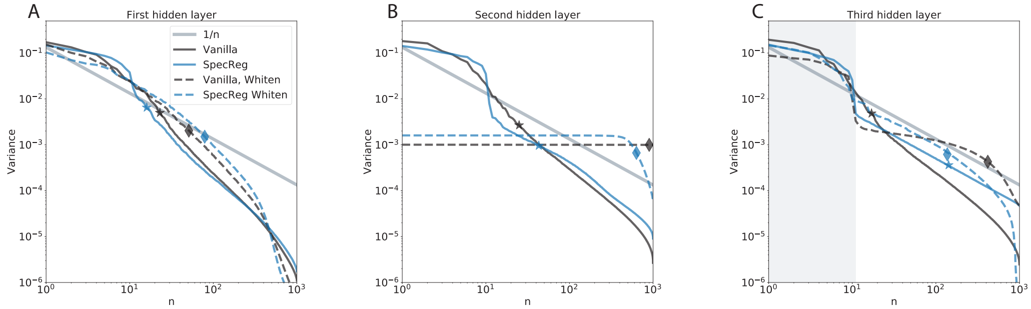

Figure A7: Spectra of the neural activity of the MLP with intermediate layer output whitened. A) Eigenspectrum of the first hidden layer. B) Eigenspectrum of the second hidden layer. Whitening the neural representation leads to an approximately flat spectra. C) Eigenspectrum of the third hidden layer. The shaded green area are the eigenvalues that are not regularized.

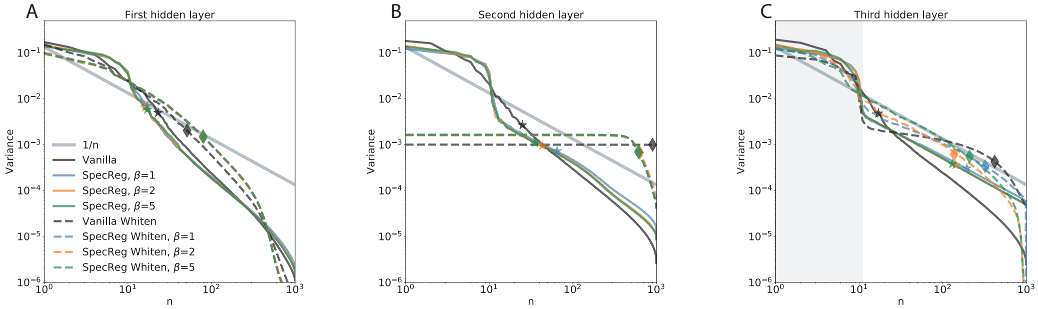

Figure A8: Spectra of the neural activity of the MLP for various values of β. A) Eigenspectrum of the first hidden layer. B) Eigenspectrum of the second hidden layer. Whitening the neural representation leads to an approximately flat spectra. C) Eigenspectrum of the third hidden layer. The shaded green area are the eigenvalues that are not regularized.

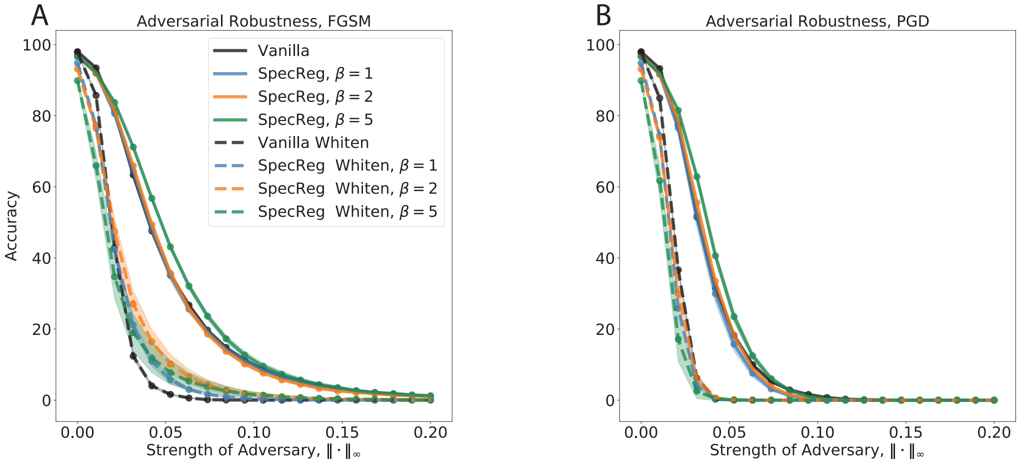

Figure A9: Adversarial robustness of the MLP for various values of β where the shaded region is $\pm$ 1 standard deviation.

B.2 CNN

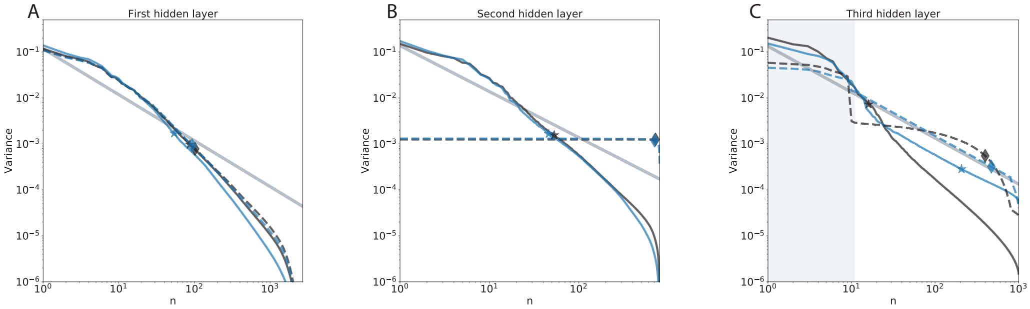

Figure A10: Spectra of the neural activity of the CNN. A) Eigenspectrum of the first hidden layer. B) Eigenspectrum of the second hidden layer. Whitening the neural representation leads to an approximately flat spectra. C) Eigenspectrum of the third hidden layer. The shaded green area are the eigenvalues that are not regularized.

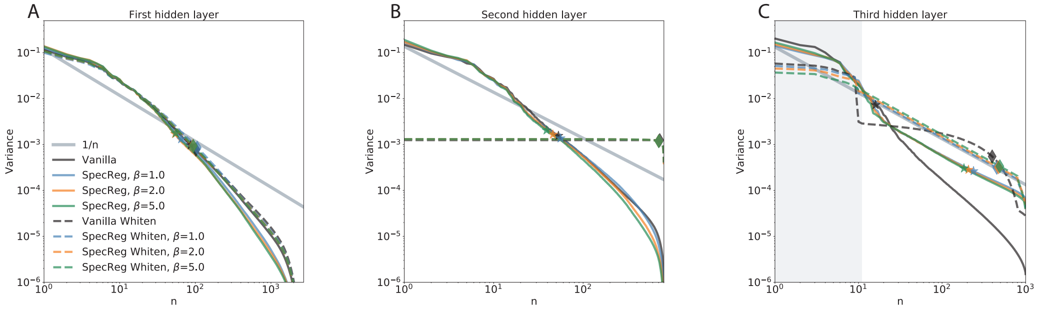

Figure A11: Spectra of the neural activity of the CNN for various values of β. A) Eigenspectrum of the first hidden layer. B) Eigenspectrum of the second hidden layer. Whitening the neural representation leads to approximately flat spectra. C) Eigenspectrum of the third hidden layer. The shaded green area are the eigenvalues that are not regularized.

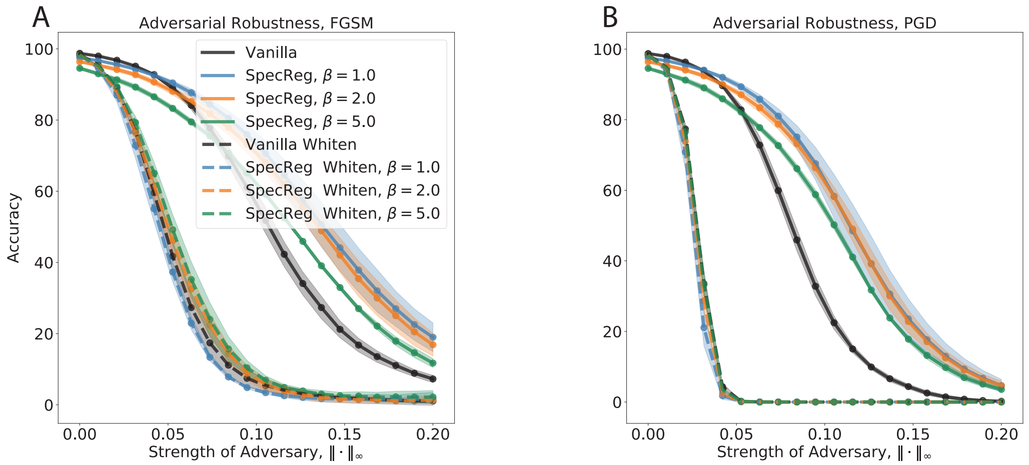

Figure A12: Adversarial robustness of the CNN for various values of β where the shaded region is ± 1 standard deviation.

C Additional Figures for Section 4.3

C.1 MLP

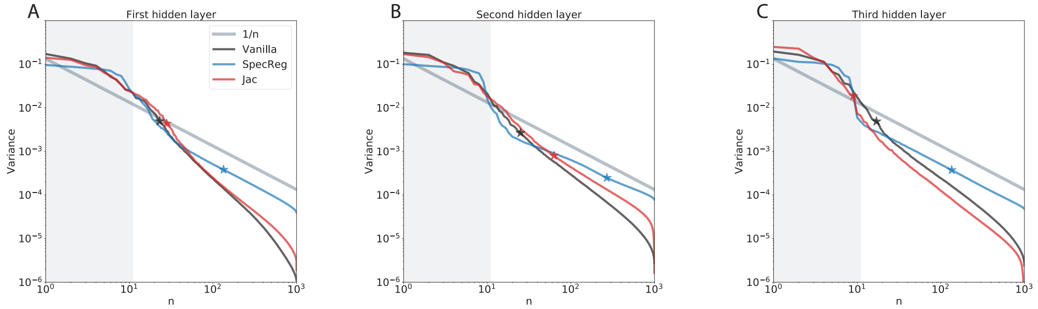

Figure A13: Spectra of the neural activity of the MLP. The shaded green area are the eigenvalues that are not regularized.

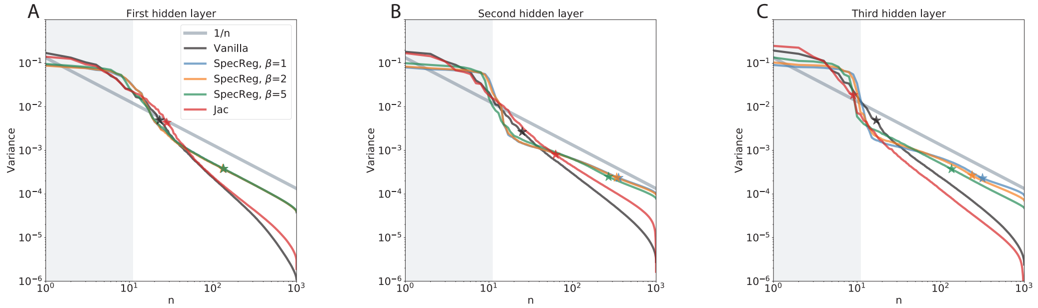

Figure A14: Spectra of the neural activity of the MLP for various values of β. The shaded green area are the eigenvalues that are not regularized.

C.2 CNN

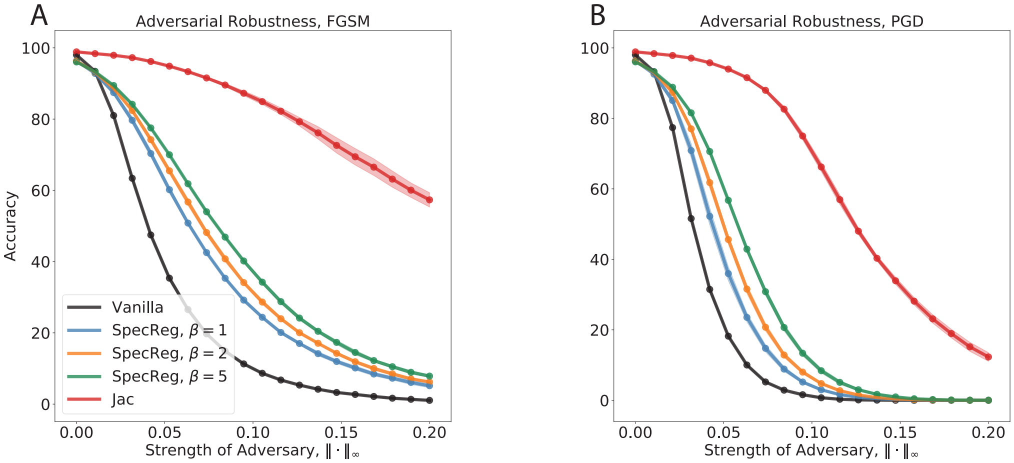

Figure A15: Adversarial robustness of the MLP for various values of β where the shaded region is ± 1 standard deviation.

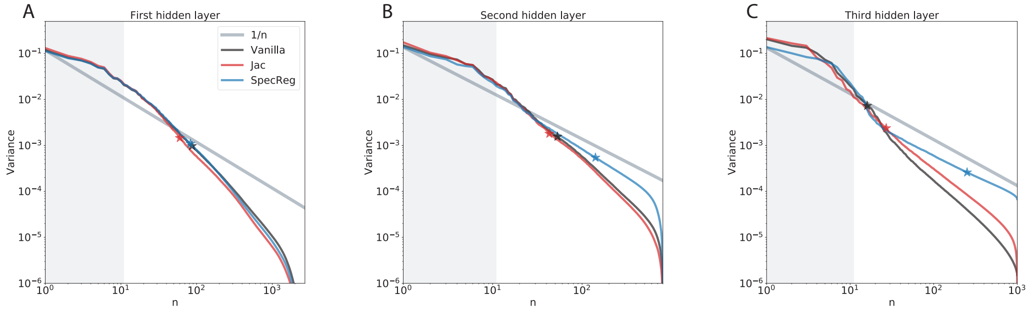

Figure A16: Spectra of the neural activity of the CNN. The shaded green area are the eigenvalues that are not regularized.

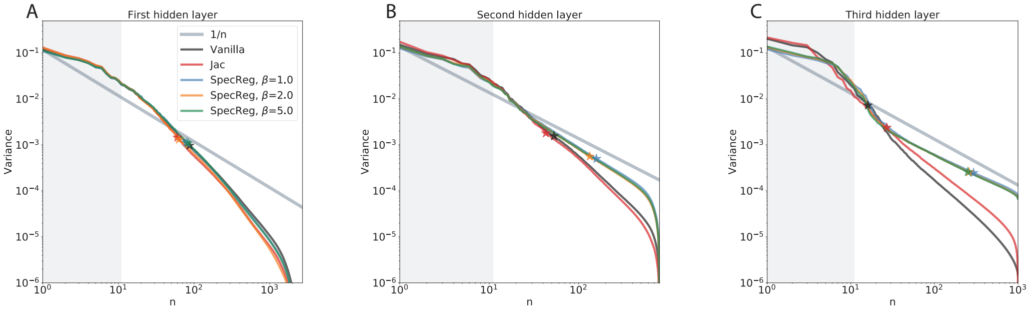

Figure A17: Spectra of the neural activity of the CNN for various values of β. The shaded green area are the eigenvalues that are not regularized.

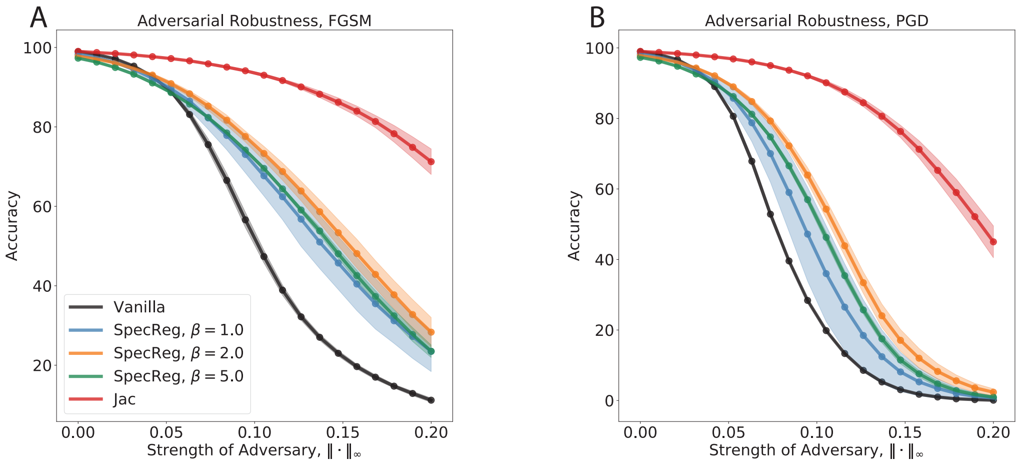

Figure A18: Adversarial robustness of the CNN for various values of β where the shaded region is $\pm$ 1 standard deviation.