Correlation of the Jacobian and Probability Flow

John Y. Shin
Department of Computer Science
New York University
New York, NY 10033
jys308@nyu.edu

Abstract

1 Introduction

Over the past decade, Deep Neural Networks (DNNs) have proven remarkably effective on a wide range of tasks in computer vision (CV), natural language processing (NLP), as well as other domains. Moreover, larger and deeper DNN models, with hundreds to thousands of layers, perform tasks seemingly impossible just a few years ago. For example, the CV architecture ResNet [1] has been successfuly trained with over 1000 layers, showing excellent generalization performance on a wide range of data sets (CIFAR10, CIFAR100, SVHN, ImageNet, etc.). Most recently, OpenAI released the NLP Language model GPT-3 [2], which has been trained on nearly a half trillion words, using 175 billion parameters, and achieving state-of-the-art (SOTA) performance on several NLP benchmarks.

The incredible size and depth of these models poses a new and deep theoretical challenges. [blah blah blah] Discuss Energy Landscape and ruggedly convexity

[What has been done before]

[Cross Sections] We do have some insight into how the Energy Landscape behaves by visualizing 2-dimensional cross-sections of small models during training, such as ResNet25. Maybe can run ourselves ?

[Analysis of the Hessian] Not really informative. Hessian only provides local information.

There has been past work on showing that the smoothness of the Hessian as a function of the weights of the final trained model is correlated with good generalization [3, 4]. Recently, there has been work on leveraging stochastic methods for the computation of the empirical spectral density of the Hessian [5, 6], and further studies of the spectrum of the Hessian have brought some contention to this claim [7, 8, 9].

[5, 7, 8, 4, 3, 9, 6]

[Past Studies of the Jacobian]

The Jacobian of the Neural Network, or the derivative of the Neural Network function with respect to either the data (input/output map) or the weights, has also been studied extensively, covering a wide range of topics, such as it's initialization [10, 11], as a measure of generalization [12, 13, 14, 15, 15, 16] as a way to regularize the network [17, 18], it's spectrum [10, 11, 19, 20, 21, 22], as a learning objective itself [23], as a measure of robustness [24], as well as it's connections to information geometry [25].

[12, 10, 13, 11, 17, 19, 14, 24, 18, 15, 25, 23, 15, 20, 16, 21, 22]

33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

[Empirical Generalization Metrics] Norm-based metrics such as WeightWatcher Correlated with generalization / test accuracy. Best metric is based on power law / heavy tailed. Not explicitly data dependent

[What needs to be done] Cross-Section is not a generalization metric, is not global

Importance of unsupervised metrics: self training

In order to characterize the Energy Landscape, traditional approaches attempt to count the number of local minima (i.e the complexity). And while this is well for theoretical analysis (such as spin glass theory, random matrix theory, etc), numerically this is quite hard. Especially for the massive production size DNNs in use today.

Here, we suggest an new, alternative approach–to study the Empirical Spectral Density (ESD) of the data-dependent Jacobian, which is readily calculated with a single epoch of Backprop using any off-the-shelf toolkit such as TensorFlow, PyTorch, etc.

Similar to the weightwatcher studies..

Show picture: compare relatively random / flat vs a deeply funneled convex Landscape ESDs

random: real world data, randomly labeled

Here is a summary of our main results:

-
-
-

2 Computing the Jacobian

Suppose we have a Neural Network classifier $f : \mathbb{R}^d \to \mathbb{R}^k$, where d is the dimensionality of the input and k is the number of classes. We define the Jacobian matrix of the Neural Network as:

$$J(\mathbf{x}) = \nabla_{\mathbf{x}} f(\mathbf{x}) \in \mathbb{R}^k \times \mathbb{R}^d \tag{1}$$

Or the derivative of the Neural Network with respect to the input. Suppose we have a training dataset of size n, with examples given as $(\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n)$, where $\mathbf{x}_i \in \mathbb{R}^d$. We can concatenate the $J(\mathbf{x})$ for each example into a vector:

$$\mathcal{J}_i = \text{flatten}(J(\mathbf{x}_i)) \in \mathbb{R}^{k \times d} \tag{2}$$

We can then form a matrix of these $\mathcal{J}_i$'s into a matrix $\mathcal{J} \in \mathbb{R}^n \times \mathbb{R}^{k \times d}$, which we will call the Jacobian matrix of the training dataset. Each row corresponds to a training example and the columns are a concatenation of the derivatives of each dimension of the output with each dimension of the input.

As a classifier trained with a cross-entropy loss and with a softmax as the final output of the network, we can interpret the output of the network as a vector of probabilities:

$$f(\mathbf{x}) = (p(y_1|\mathbf{x}), p(y_2|\mathbf{x}), \cdots, p(y_k|\mathbf{x})) \in \mathbb{R}^k \tag{3}$$

The Jacobian matrix of the neural network then becomes:

$$J(\mathbf{x}) = \nabla_{\mathbf{x}} f(\mathbf{x}) = (\nabla_{\mathbf{x}} p(y_1|\mathbf{x}), \nabla_{\mathbf{x}} p(y_2|x), \cdots, \nabla_{\mathbf{x}} p(y_k|\mathbf{x})) \in \mathbb{R}^k \times \mathbb{R}^d \tag{4}$$

For conciseness (and abuse) of notation, we can write the the flattened $\mathcal{J}_i$ in vector notation as:

$$\mathcal{J}_i = \nabla_{\mathbf{x}_i} p(\mathbf{y}|\mathbf{x}_i) \in \mathbb{R}^{k \times d} \tag{5}$$

We can then build the correlation matrix of the Jacobian over the training dataset as:

$$M_{ij} = (\mathcal{J}\mathcal{J}^T)_{ij} = \nabla_{\mathbf{x}_i} p(\mathbf{y}|\mathbf{x}_i) \cdot \nabla_{\mathbf{x}_j} p(\mathbf{y}|\mathbf{x}_j) \in \mathbb{R}^{n\times n} \tag{6}$$

Theorem 1. *testing*

References

[1] 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.

[2] Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

[3] Levent Sagun, Utku Evci, V Ugur Guney, Yann Dauphin, and Leon Bottou. Empirical analysis of the hessian of over-parametrized neural networks. *arXiv preprint arXiv:1706.04454*, 2017.

[4] Pratik Chaudhari, Anna Choromanska, Stefano Soatto, Yann LeCun, Carlo Baldassi, Christian Borgs, Jennifer Chayes, Levent Sagun, and Riccardo Zecchina. Entropy-sgd: Biasing gradient descent into wide valleys. *Journal of Statistical Mechanics: Theory and Experiment*, 2019(12):124018, 2019.

[5] Behrooz Ghorbani, Shankar Krishnan, and Ying Xiao. An investigation into neural net optimization via hessian eigenvalue density. *arXiv preprint arXiv:1901.10159*, 2019.

[6] Zhewei Yao, Amir Gholami, Kurt Keutzer, and Michael Mahoney. Pyhessian: Neural networks through the lens of the hessian. *arXiv preprint arXiv:1912.07145*, 2019.

[7] Micah Goldblum, Jonas Geiping, Avi Schwarzschild, Michael Moeller, and Tom Goldstein. Truth or backpropaganda? an empirical investigation of deep learning theory. *arXiv preprint arXiv:1910.00359*, 2019.

[8] Wesley J Maddox, Gregory Benton, and Andrew Gordon Wilson. Rethinking parameter counting in deep models: Effective dimensionality revisited. *arXiv preprint arXiv:2003.02139*, 2020.

[9] Diego Granziol. Flatness is a false friend. *arXiv preprint arXiv:2006.09091*, 2020.

[10] Jeffrey Pennington, Samuel Schoenholz, and Surya Ganguli. Resurrecting the sigmoid in deep learning through dynamical isometry: theory and practice. In *Advances in neural information processing systems*, pages 4785–4795, 2017.

[11] Jeffrey Pennington, Samuel S Schoenholz, and Surya Ganguli. The emergence of spectral universality in deep networks. *arXiv preprint arXiv:1802.09979*, 2018.

[12] Jure Sokolić, Raja Giryes, Guillermo Sapiro, and Miguel RD Rodrigues. Robust large margin deep neural networks. *IEEE Transactions on Signal Processing*, 65(16):4265–4280, 2017.

[13] 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.

[14] Yiding Jiang, Dilip Krishnan, Hossein Mobahi, and Samy Bengio. Predicting the generalization gap in deep networks with margin distributions. *arXiv preprint arXiv:1810.00113*, 2018.

[15] Samet Oymak, Zalan Fabian, Mingchen Li, and Mahdi Soltanolkotabi. Generalization, adaptation and low-rank representation in neural networks. In *2019 53rd Asilomar Conference on Signals, Systems, and Computers*, pages 581–585. IEEE, 2019.

[16] Amartya Sanyal, Philip HS Torr, and Puneet K Dokania. Stable rank normalization for improved generalization in neural networks and gans. *arXiv preprint arXiv:1906.04659*, 2019.

[17] Fredrik Bagge Carlson, Rolf Johansson, and Anders Robertsson. Tangent-space regularization for neural-network models of dynamical systems. *arXiv*, pages arXiv–1806, 2018.

[18] Judy Hoffman, Daniel A Roberts, and Sho Yaida. Robust learning with jacobian regularization. *arXiv preprint arXiv:1908.02729*, 2019.

[19] Zenan Ling and Robert C Qiu. Spectrum concentration in deep residual learning: a free probability approach. *IEEE Access*, 7:105212–105223, 2019.

[20] Matthew Shunshi Zhang and Bradly Stadie. One-shot pruning of recurrent neural networks by jacobian spectrum evaluation. *arXiv preprint arXiv:1912.00120*, 2019.

[21] Wojciech Tarnowski, Piotr Warchoł, Stanisław Jastrzbski, Jacek Tabor, and Maciej Nowak. Dynamical isometry is achieved in residual networks in a universal way for any activation function. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 2221–2230, 2019.

[22] Shengjie Wang, Abdel-rahman Mohamed, Rich Caruana, Jeff Bilmes, Matthai Plilipose, Matthew Richardson, Krzysztof Geras, Gregor Urban, and Ozlem Aslan. Analysis of deep neural networks with extended data jacobian matrix. In *International Conference on Machine Learning*, pages 718–726, 2016.

[23] Jonathan Lorraine and Safwan Hossain. Jacnet: Learning functions with structured jacobians.

[24] Fuxun Yu, Chenchen Liu, Yanzhi Wang, Liang Zhao, and Xiang Chen. Interpreting adversarial robustness: A view from decision surface in input space. *arXiv preprint arXiv:1810.00144*, 2018.

[25] Piotr A Sokol and Il Memming Park. Information geometry of orthogonal initializations and training. *arXiv preprint arXiv:1810.03785*, 2018.