The Jacobian Spectral Density: A Complexity Metric for Funneled Energy Landscapes in Deep Neural Networks

ABSTRACT

Abstract

ACM Reference Format:

. 2020. The Jacobian Spectral Density: A Complexity Metric for Funneled Energy Landscapes in Deep Neural Networks. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Over the past decade, Deep Neural Networks (DNNs) have proven remarkably effective on a wide range of computer vision (CV), natual language processing (NLP), and 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 has been successfuly trained with over 1000 layers, showing excellent generqlization performance on a wide range of data sets (CIFAR10, CIFAR100, SVHN, ImagaeNet, etc. Most recently, openAI released the NLP Language model GPT3, 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.

[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 matric 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:

-
-
-

[john: The story so far: we are interested in metrics pertaining to neural networks in which we can predict their ability to generalize. Recent work by Google [ICML 17] showed that MLPs can converge to 100% training accuracy with the "random label transform" or shuffling of all the class labels over the training set.

First, we performed experiments where we computed the spectral density of the Hessian of MLPs trained normally on CIFAR10 as well as those trained under the random label transformation. For shallow MLPs, the spectral density of the Hessian showed a larger index (number of negative eigenvalues), as well as more symmetry about zero and overall more roughness (need to quantify this better). For deeper MLPs, this difference between the spectral density of the random label transformed network and the normal network went away.

In addition, we looked at 10 random initializations of MLPs trained normally and trained on the random label transform. We looked at cross-correlations between these initializations, between normal-normal, random-random, and normal-random (more about these experiments).

One motivation for our work was to find a metric that would be able to discriminate between models trained normally and ones trained on the random label transform.]

Funneled Enegy Landscapes. [where to put this ?]

Organization of this paper.

2 METHODS

[start repeated section] Let us write the Energy Landscape (or optimization function, parameterized by $\mathbf{W}_l$s and $\mathbf{b}_l$s) for a DNN with L layers, activation functions $h_l(\cdot)$, and $N \times M$ weight matrices $\mathbf{W}_l$ and biases $\mathbf{b}_l$, as:

$$E_{DNN} = h_L(\mathbf{W}_L \times h_{L-1}(\mathbf{W}_{L-1} \times h_{L-2}(\cdots) + \mathbf{b}_{L-1}) + \mathbf{b}_L). \quad (1)$$

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

Conference'17, July 2017, Washington, DC, USA

© 2020 Association for Computing Machinery.

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00

https://doi.org/10.1145/nnnnnnn.nnnnnnn

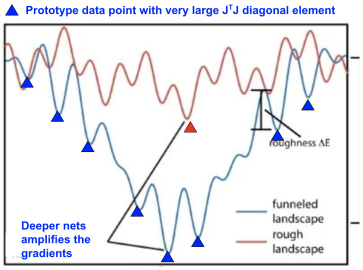

Prototype data point with very large J^TJ diagonal element
roughness ΔE
funneled landscape
rough landscape
Deeper nets amplifies the gradients

Figure 1: Sample figure caption.

Each DNN layer contains one or more layer 2D $N \times M$ weight matrices, $\mathbf{W}_l$, or pre-activation maps, $\mathbf{W}_{i,l}$, extracted from 2D Convolutional layers, and where $N > M$.[1] (We may drop the i and/or i, l subscripts below.) See Appendix A for how we define the Conv2D layer matrixes and for our choices of normalization.

Assume we are given several pretrained DNNs, e.g., as part of an architecture series. The models have been trained and evaluated on labeled data $\{d_i, y_i\} \in \mathcal{D}$, using standard techniques. The pretained pytroch model files are publicly-available, and the test accuracies have been reported online. [end repeated section]

In this study, we have acces to the training and test data, but we do not train the models ourselves;

Data-Dependent Jacobian.
previous work - regularizartion by diagonal elements - Ganguli and Pennington

Ganguli et al. chatacterize the ESD of the Jacobian using a (free) cumulant expansion. No need to compute the cumulants we can simplly compute the complete ESD, either using the test data, or on the training data using RandNLA methods. (Moreover, the (free) cumulants wiuld be different for a truly heavy tailed $\mathbf{J}$ vs a Gsussian random matrix. Gotta think carefully on that but I beleive this is the case, depending on what is evaluated) And earlier work suggest this is the case

[move this below?] We simply fit the ESD to a Power Law, and characterize it by

- Power Law exponent α
- Maximum eigenvalue λ_{max} or Spectral Norm
- (Effective) Rank (hard rank, number of zero eigenvalues, etc)

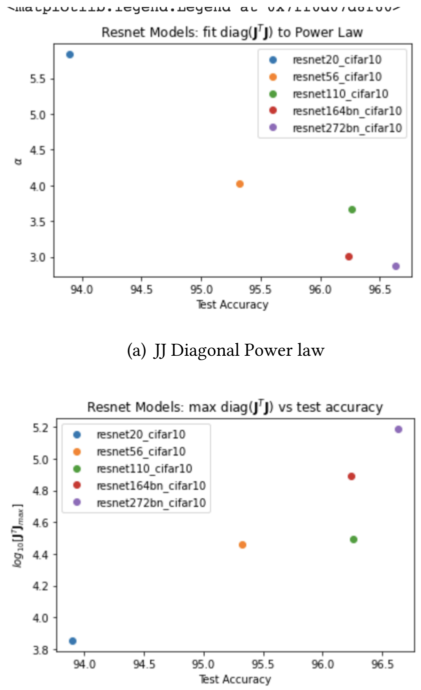
Empirical Spectral Density. Evaluate over test set (training set too large)
Dependence on batch size Jacobian during training is batch size dependent because of Batch Norm, other layers. Set model.eval(): do not include bathc norm. Final Jacobian is not-batch size dependent

$$\rho(\lambda) \sim \lambda^{\alpha}, \quad \lambda \leq \lambda_{max}, \tag{2}$$

where λ_{max} *is* the largest eigenvalue of $\mathbf{X} = \mathbf{J}^T\mathbf{J}$. Each of these quantities is defined for sample Jacobian $\mathbf{J}$ matrix.

Compute diagonal first, Jacobian diagonally dominant

Both diagonal elements and ES are Heavy Tailed, but can be very different. Correlations matter.



(a) JJ Diagonal Power law

(b) JJ Diagonal Max Element

Figure 2: Sample caption

3 METHODS

Diagonal $J^T\mathbf{J}$.

4 CONCLUSION

CONCLUSION

[1] We do not use intra-layer information from the models in our quality metrics, but (as we will describe) our metrics can be used to learn about intra-layer model properties.

A APPENDIX

In this appendix, we provide more details on several issues that are important for the reproducibility of our results.