




Delve: Neural Network Feature Variance Analysis

Justin Shenk^{*1,2}, Mats L. Richter^{†2}, and Wolf Byttner³

¹ VisioLab, Berlin, Germany ² Institute of Cognitive Science, University of Osnabrueck, Osnabrueck, Germany ³ Rapid Health, London, England, United Kingdom

DOI: [10.21105/joss.03992](https://doi.org/10.21105/joss.03992)

Software

- [Review](#) 
- [Repository](#) 
- [Archive](#) 

Editor: [Rachel Kurchin](#) 

Reviewers:

- [@sambitdash](#)
- [@dionhaefner](#)

Submitted: 08 October 2021

Published: 14 December 2021

License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC BY 4.0](#)).

Summary

Designing neural networks is a complex task. Deep neural networks are often referred to as “black box” models - little insight in the function they approximate is gained from looking at the structure of layer outputs. Delve is a tool for looking at how a neural network represents data, and how these representations, or features, change throughout training. This tool enables deep learning researchers to understand the limitations and suggest improvements for the design of their networks, such as removing or adding layers.

Several tools exist which allow analyzing neural networks after and during training. These techniques can be characterized by their focus on either data or model as well as their level of abstractness. Examples for abstract model-oriented techniques are tools for analyzing the sharpness of local optima ([Keskar et al., 2016](#); [Novak et al., 2018](#)), which can be an indicator for the generalizing capabilities of the trained models. In these scenarios the complex of dataset and model is reduced to the error surface, allowing for insights into the differences between different setups. A less abstract data-centric technique GradCam by Selvaraju et al. ([Chattopadhyay et al., 2018](#); [Selvaraju et al., 2019](#)), reduces the model to a set of class-activation maps that can be overlayed over individual data points to get an intuitive understanding of the inference process. SVCCA ([Morcos et al., 2018](#); [Raghu et al., 2017](#)) can be considered model-centric and a middle ground in terms of abstractness, since it allows the comparative analysis on the features extracted by specific layers. SVCCA is also relevant from a functional perspective for this work, since it uses singular value decomposition as a core technique to obtain the analysis results. Another model-centric tool that allows for a layer-by-layer analysis is logistic regression probes ([Alain & Bengio, 2016](#)), which utilize logistic regressions trained on the output of a hidden layer to measure the linear separability of the data and thus the quality of the intermediate solution quality of a classifier model.

The latter is of great importance for this work, since logistic regression probes are often used to compare models and identify the contribution of layers to overall performance ([Richter, Byttner, et al., 2021](#); [Richter, Schöning, et al., 2021](#); [Shenk et al., 2020](#)) and to demonstrate that the saturation metric is capable of showing parameter-inefficiencies in neural network architectures.

However, the aforementioned tools have significant limitation in terms of their usefulness in practical application scenarios, where these tools are to be used to improve the performance of a given model. In case of data-centric tools like GradCam the solution propagates back to the data, which makes it hard to derive decisions regarding the neural architecture. However, the biggest concern in all aforementioned tools are the cost in computational resources and the integration of the analysis into the workflow of a deep learning practitioner. Tools like SVCCA and logistic regression probes require complex and computationally expensive procedures that need to be conducted after training. This naturally limits these techniques to small benchmarks

*co-first author

†co-first author

and primarily academic datasets like Cifar10 (Shenk et al., 2020). A analysis tool that is to be used during the development of a deep learning based model needs to be able to be used with little computational and workflow overhead as possible. Ideally the analysis can be done while the training is in progress, allowing the researcher to interrupt potentially long running training session to improve the model. Saturation was proposed in 2018 (Shenk, 2018) and later refined (Shenk et al., 2020) and is the only known analysis technique known to the authors that has this capability while also allowing to identify parameter-inefficiencies in the setup (Richter, Byttner, et al., 2021; Richter, Schöning, et al., 2021; Shenk et al., 2020). In order to make saturation usable an application scenario, it is necessary to provide a easy-to-use framework that allows for an integration of the tool into the normal training and inference code with only minimally invasive changes. It is also necessary that the computation and analysis can be done online as part of the regular forward pass of the model, to make the integration as seamless as possible.

DeIve provides a framework for allowing a seamless and minimal overhead integration for saturation and other statistical analysis of neural network layer eigenspaces. It hooks into PyTorch (Paszke et al., 2019) models and allows saving statistics via TensorBoard (Abadi et al., 2015) events or CSV writers. A comprehensive source of documentation is provided on the home page (<http://delve-docs.readthedocs.io>).

Statement of Need

Research on changes in neural network representations has exploded in the past years (Alain & Bengio, 2016; Montavon et al., 2010; Morcos et al., 2018; Raghu et al., 2017; Selvaraju et al., 2019; Zhou et al., 2016). Furthermore, researchers who are interested in developing novel algorithms must implement from scratch much of the computational and algorithmic infrastructure for analysis and visualization. By packaging a library that is particularly useful for extracting statistics from neural network training, future researchers can benefit from access to a high-level interface and clearly documented methods for their work. DeIve has already been used in a number of scientific publications (Richter, Byttner, et al., 2021; Richter, Schöning, et al., 2021; Shenk et al., 2020). The combination of ease of usage and extensibility in DeIve enables exciting scientific explorations for machine learning researchers and engineers. The source code for DeIve has been archived to Zenodo with the linked DOI: (Shenk et al., 2021)

Overview of the Library

The software is structured into several modules which distribute tasks. Full details are available at <https://delve-docs.readthedocs.io/>.

The TensorBoardX SummaryWriter (Abadi et al., 2015) is used to efficiently save artifacts like images or statistics during training with minimal interruption. A variety of layer feature statistics can be observed:

Statistic

intrinsic dimensionality

layer saturation (intrinsic dimensionality divided by feature space dimensionality)

the covariance-matrix

the determinant of the covariance matrix (also known as generalized variance)

the trace of the covariance matrix, a measure of variance of the data

the trace of the diagonal matrix, another way of measuring the dispersion of the data.

layer saturation (intrinsic dimensionality divided by feature space dimensionality)

Several layers are currently supported:

- 79 ▪ Convolutional
- 80 ▪ Linear
- 81 ▪ LSTM

82 Additional layers such as PyTorch's ConvTranspose2D are planned for future development (see
83 issue #43).

84 Eigendecomposition of the feature covariance matrix

85 Saturation is a measure of the rank of the layer feature eigenspace introduced by (Shenk,
86 2018; Shenk et al., 2019) and further explored in (Shenk et al., 2020).

87 Covariance matrix of features is computed online as described in (Shenk et al., 2020):

$$Q(Z_l, Z_l) = \frac{\sum_{b=0}^B A_{l,b}^T A_{l,b}}{n} - (\bar{A}_l \otimes \bar{A}_l)$$

88 for B batches of layer output matrix A_l and n number of samples.

89 References

- 90 Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis,
91 A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia,
92 Y., Jozefowicz, R., Kaiser, L., Kudlur, M., ... Zheng, X. (2015). *TensorFlow: Large-scale
93 machine learning on heterogeneous systems*. <https://www.tensorflow.org/>
- 94 Alain, G., & Bengio, Y. (2016). Understanding intermediate layers using linear classifier
95 probes. *ArXiv, abs/1610.01644*.
- 96 Chattopadhyay, A., Sarkar, A., Howlader, P., & Balasubramanian, V. N. (2018). Grad-
97 CAM++: Generalized gradient-based visual explanations for deep convolutional networks.
98 *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*. [https://doi.
99 org/10.1109/wacv.2018.00097](https://doi.org/10.1109/wacv.2018.00097)
- 100 Keskar, N. S., Mudigere, D., Nocedal, J., Smelyanskiy, M., & Tang, P. T. P. (2016). On
101 large-batch training for deep learning: Generalization gap and sharp minima. *CoRR,
102 abs/1609.04836*.
- 103 Montavon, G., Müller, K.-R., & Braun, M. L. (2010). Layer-wise analysis of deep net-
104 works with gaussian kernels. In J. D. Lafferty, C. K. I. Williams, J. Shawe-Taylor,
105 R. S. Zemel, & A. Culotta (Eds.), *Advances in neural information processing sys-
106 tems 23* (pp. 1678–1686). Curran Associates, Inc. [http://papers.nips.cc/paper/
107 4061-layer-wise-analysis-of-deep-networks-with-gaussian-kernels.pdf](http://papers.nips.cc/paper/4061-layer-wise-analysis-of-deep-networks-with-gaussian-kernels.pdf)
- 108 Morcos, A. S., Raghu, M., & Bengio, S. (2018). *Insights on representational similarity in
109 neural networks with canonical correlation*. <http://arxiv.org/abs/1806.05759>
- 110 Novak, R., Bahri, Y., Abolafia, D. A., Pennington, J., & Sohl-Dickstein, J. (2018). Sensitivity
111 and generalization in neural networks: An empirical study. *International Conference on
112 Learning Representations*. <https://openreview.net/forum?id=HJC2SzZCW>
- 113 Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z.,
114 Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M.,
115 Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., ... Chintala, S. (2019). PyTorch: An
116 imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A.
117 Beygelzimer, F. dAlché-Buc, E. Fox, & R. Garnett (Eds.), *Advances in neural information*

- 118 *processing systems* 32 (pp. 8024–8035). Curran Associates, Inc. [http://papers.neurips.](http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf)
119 [cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf](http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf)
- 120 Raghu, M., Gilmer, J., Yosinski, J., & Sohl-Dickstein, J. (2017). *SVCCA: Singular vector*
121 *canonical correlation analysis for deep learning dynamics and interpretability*. [http://](http://arxiv.org/abs/1706.05806)
122 arxiv.org/abs/1706.05806
- 123 Richter, M. L., Byttner, W., Krumnack, U., Schallner, L., & Shenk, J. (2021). Size matters.
124 *CoRR*, *abs/2102.01582*. https://doi.org/10.1007/978-3-030-86340-1_11
- 125 Richter, M. L., Schöning, J., & Krumnack, U. (2021). Should you go deeper? Optimizing
126 convolutional neural network architectures without training by receptive field analysis.
127 *CoRR*, *abs/2106.12307*. <https://arxiv.org/abs/2106.12307>
- 128 Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., & Batra, D.
129 (2019). Grad-CAM: Visual explanations from deep networks via gradient-based
130 localization. *International Journal of Computer Vision*, 128(2), 336–359. [https:](https://doi.org/10.1007/s11263-019-01228-7)
131 [//doi.org/10.1007/s11263-019-01228-7](https://doi.org/10.1007/s11263-019-01228-7)
- 132 Shenk, J. (2018). *Spectral Decomposition for Live Guidance of Neural Network Architecture*
133 *Design* [Master's Thesis]. University of Osnabrück.
- 134 Shenk, J., Richter, M. L., Arpteg, A., & Huss, M. (2019). Spectral analysis of latent repre-
135 sentations. *CoRR*, *abs/1907.08589*. <http://arxiv.org/abs/1907.08589>
- 136 Shenk, J., Richter, M. L., Byttner, W., Arpteg, A., & Huss, M. (2020). Feature space
137 saturation during training. *CoRR*, *abs/2006.08679*. <https://arxiv.org/abs/2006.08679>
- 138 Shenk, J., Richter, M. L., Byttner, W., & Marcinkiewicz, M. (2021). *Delve-team/delve:*
139 *v0.1.45* (Version v0.1.45) [Computer software]. Zenodo. [https://doi.org/10.5281/zenodo.](https://doi.org/10.5281/zenodo.5233860)
140 [5233860](https://doi.org/10.5281/zenodo.5233860)
- 141 Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., & Torralba, A. (2016). Learning deep features
142 for discriminative localization. *2016 IEEE Conference on Computer Vision and Pattern*
143 *Recognition (CVPR)*, 2921–2929. <https://doi.org/10.1109/cvpr.2016.319>