Caveats of neural persistence in deep neural networks

Leander Girrbach [1] **Anders Christensen** [1 2] **Ole Winther** [2 3] **Zeynep Akata** [1 4] **A. Sophia Koepke** [1]

Abstract

Neural Persistence is a prominent measure for quantifying neural network complexity, proposed in the emerging field of topological data analysis in deep learning. In this work, however, we find both theoretically and empirically that the variance of network weights and spatial concentration of large weights are the main factors that impact neural persistence. First, we prove tighter bounds on neural persistence that motivate this claim theoretically. Then, we confirm that our interpretation holds in practise by calculating neural persistence for synthetic weight matrices and for trained deep neural networks. This raises the question if the benefits of neural persistence can be achieved by simpler means, since already calculating 0-order persistent homology for large matrices is costly.

1. Introduction

Analysing deep neural networks to gain a better understanding of their inner workings is crucial, given their now ubiquitous use and practical success for a wide variety of applications. However, this is a notoriously difficult problem. *Topological Data Analysis (TDA)* has gained popularity for analysing machine learning models, and in particular deep learning models. TDA investigates data in terms of its scale-invariant topological properties, which are robust to perturbations (Cohen-Steiner et al., 2007).

Recent works consider neural networks as weighted graphs, which allows for analysis with tools from TDA developed for such data structures (Rieck, 2023). This is possible by considering the intermediate feature activations as vertices, and parameters as edges. The corresponding network weights are then interpreted as edge weights. Using this perspective, Rieck et al. (2019) define *neural persistence*, a popular measure[1] of neural network complexity which is calculated on the weights of trained neural networks, i.e. the edges of the computational graph. Neural persistence does not take the data and intermediate or output activations into account. Nevertheless, Rieck et al. (2019) show that neural persistence can be used as an early-stopping criterion in place of a validation set.

2. Understanding neural persistence

In this section, we aim at providing a deeper understanding of neural persistence. In particular, we identify the *variance* of weights and *spatial concentration* of large weights as important factors that impact neural persistence. We formalise these insights by deriving tighter bounds on neural persistence in terms of max-over-row and max-over-column values of weight matrices.

Before stating the definition of neural persistence, we recall the definition of complete bipartite graphs.

Definition 2.1 (Bipartite graph)**.** A graph $G = (V, E)$ is called *bipartite* if vertices V can be separated into two disjoint subsets A, B with $V = A \cup B$ and $A \cap B = \emptyset$, and all edges $e \in E$ are of the form $e = (a, b)$ with $a \in A$ and $b \in B$, i.e. every edge connects a vertex in A to one in B.

Definition 2.2 (Complete bipartite)**.** G is *complete bipartite* if edges between all vertices in A and all vertices in B exist.

Remark 2.3. Any matrix $W \in \mathbb{R}^{n \times m}$ can be interpreted as the adjacency matrix of an undirected weighted complete bipartite graph. In this case, rows and columns correspond to vertices in A and B, respectively. The matrix entries then resemble edge weights between all vertices in A and B.

Rieck et al. assert that neural persistence can be defined in terms of the maximum spanning tree (MST) of a complete bipartite graph instead of persistent homology.

Definition 2.4 (Neural persistence)**.** Let $W \in [0; 1]^{n \times m}$ be a matrix with n rows, m columns, and entries bounded below by 0 and above by 1. Throughout this paper, we denote entries in W with row index i and column index j as $W_{i,j}$. As in Remark 2.3, W can be interpreted as the adjacency

[1]University of Tübingen [2]Technical University of Denmark [3]University of Copenhagen [4]MPI for Intelligent Systems. Correspondence to: Leander Girrbach <leander.girrbach@uni-tuebingen.de>.

Presented at the 2^{nd} Annual Workshop on Topology, Algebra, and Geometry in Machine Learning (TAG-ML) at the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. 2023.

[1]The here considered measures of network complexity do not comply with the measure-theoretic definition of a measure.

matrix of an undirected weighted complete bipartite graph $G_W = (V_W, E_W)$. Let $\mathrm{MST}(G_W) = (V_W, E_{\mathrm{MST}(G_W)})$ with $E_{\mathrm{MST}(G_W)} \subset E_W$ be the unique MST of G_W. In general, uniqueness is not guaranteed, but can be achieved by infinitesimal perturbations. Then, let $\mathrm{MST}^w(G_W)$ be the set of weights of edges contained in the MST, i.e.

$$\mathrm{MST}^w(G_W) := \{W_{v,v'} \mid (v, v') \in E_{\mathrm{MST}(G_W)}\}. \quad (1)$$

The neural persistence $\mathrm{NP}_p(W)$ is defined as

$$\mathrm{NP}_p(W) := \left(1 + \sum_{w \in \mathrm{MST}^w(G_W)} (1-w)^p\right)^{\frac{1}{p}}, \quad (2)$$

and subsequently neural persistence for an entire neural network is defined as the average neural persistence of all layers. Weights, which can have arbitrary values, are mapped to the range $[0, 1]$ by taking the absolute value and then dividing by the largest value in the neural network. For the remainder of this paper, we assume that all weights are in $[0, 1]$.

Normalised neural persistence. To make neural persistence values of matrices with different sizes comparable, Rieck et al. propose to divide neural persistence by the theoretical upper bound $(n + m - 1)^{\frac{1}{p}}$ (see Theorem 1 in (Rieck et al., 2019)). This normalisation maps neural persistence values to the range $[0, 1]$. We follow Rieck et al. and use the proposed normalisation and $p = 2$ in all experiments.

Variance and spatial concentration of large weights impact neural persistence. Defining neural persistence in terms of the MST of a complete bipartite graph provides the interesting insight that the neural persistence value is closely related to max-over-rows and max-over-columns values in the matrix. This characterisation provides intuitions about properties of weight matrices that influence the neural persistence. These intuitions are formalised in Theorem 2.5 as bounds on neural persistence which are tighter than the bounds given in Theorem 2 in (Rieck et al., 2019).

Theorem 2.5. *Let $G_W = (V_W, E_W)$ be a weighted complete bipartite graph as in Definition 2.4 with edge weights given by W and $V_W = A \cup B$, $A \cap B = \emptyset$. To simplify notation, we define*

$$\Phi_b := (1 - \max_{a \in A} W_{a,b})^p \quad \text{for } b \in B, \quad (3)$$

$$\Psi_a := (1 - \max_{b \in B} W_{a,b})^p \quad \text{for } a \in A. \quad (4)$$

Using these shortcuts, we define

$$L := \left(\sum_{b \in B} \Phi_b + \sum_{a \in A} \Psi_a\right)^{\frac{1}{p}}, \quad (5)$$

$$U := \left(|B \setminus B_{\not\sim A}| + \sum_{b \in B_{\not\sim A}} \Phi_b + \sum_{a \in A} \Psi_a\right)^{\frac{1}{p}}, \quad (6)$$

where

$$B_{\not\sim A} := \{b \in B \mid \forall a \in A : b \neq \operatorname*{argmax}_{b' \in B} W_{a,b'}\}. \quad (7)$$

$B_{\not\sim A} \subset B$ can be thought of as the set of columns whose maximal element does not coincide with the maximal element in any row of W.

Then, the following inequalities hold:

$$0 \leq L \leq \mathrm{NP}_p(W) \leq U \leq (n+m)^{\frac{1}{p}}. \quad (8)$$

Proof (sketch). For the lower bound, using properties of spanning trees, we construct a bijection between vertices V (with one vertex excluded) and edges in $\mathrm{MST}(G_W)$. Each vertex v is mapped to an edge that is connected to v. Using this bijection, we can bound the weight of each edge in the MST by the maximum weight of any edge connected to the respective vertex. Since maximum weights of edges connected to vertices correspond to max-over-rows and max-over-columns values, we obtain the formulation of L. For the upper bound, we observe that all max-over-rows and max-over columns values are necessarily included in $\mathrm{MST}^w(G_W)$. However, in some cases max-over-rows values and max-over-columns values coincide. Therefore, this observation leaves some values in $\mathrm{MST}^w(G_W)$ undetermined. For these, we choose the value that maximises neural persistence, i.e. 0, to obtain an upper bound.

Interpretation of bounds on neural persistence. As already mentioned, the bounds on neural persistence derived in Theorem 2.5 mostly depend on max-over-columns and max-over-rows values in W and thus identify additional factors that impact neural persistence, in particular the *variance* and *spatial concentration* of weights.

The lower bound L is tighter when the variance of weights is smaller. It then is more likely that the actual weight in the MST chosen for any vertex v is close to the maximum weight connected to v. If mean weight values and variances of weights are highly correlated, which is the case in practise, neural persistence increases with lower variance. The reason is that the lower variance causes the expected maximum value of a sample to be closer to the mean weight value, i.e. the max-over-rows and max-over-columns values will also decrease together with lower mean and variance.

The upper bound U is tighter when $B_{\not\sim A}$ is smaller. This is the case when large weights are concentrated on edges connected to few or even a single vertex, i.e. when there is relevant spatial concentration of large weights on certain rows or columns. In the extreme case, all edges with maximum weight for any vertex in A are connected to the same vertex $b \in B$. Then, we know that edges with maximum weight, i.e. max-over-column values, for all vertices in $B \setminus \{b\}$ will be part of the MST. In this case, we have

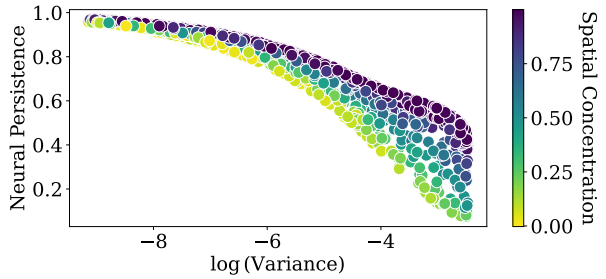

Figure 1: Correspondence of neural persistence with log-transformed variance and spatial concentration of entries in (individual) synthetic matrices with entries in $[0, 1]$.

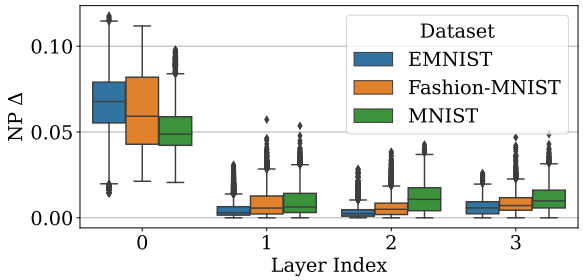

Figure 2: Impact of random permutations in weight matrices of trained deep neural networks on neural persistence.

equality of $\mathrm{NP}_p(W) = U$. Also, when large weights are concentrated on fewer rows or columns, max-over-rows and max-over-columns values will be generally lower, which leads to higher neural persistence values.

In Figure 1, we demonstrate that this interpretation is in agreement with empirical results on synthetic weight matrices. Here, we sample matrices of shape 100×100 with entries in $[0, 1]$ from truncated Pareto distributions and truncated normal distributions with varying skew. Empirically, these families of distributions approximate the (normalised) weights in trained neural networks (see Section 3) well. Spatial concentration is controlled by treating the matrix as a vector and approximately sorting entries by magnitude in increasing order. Approximate sorting is achieved by adding Gaussian noise with suitable variance (to achieve the desired spatial concentration of large entries) before sorting and subsequently again removing the noise. We measure the effect by the (normalised) number of inversions (Estivill-Castro, 2004), which yields a score between -1 and 1, where 0 indicates random dispersion and 1 indicates perfect sorting in increasing order. As predicted by our theoretical analysis, neural persistence increases monotonically both with lower variance and also with higher spatial concentration of large weights. The correspondence appears roughly linear as the R^2 score of a linear regression fit is $\approx 89\%$, but flattens for neural persistence values close to one.

3. Experimental analysis

Motivated by our theoretical analysis, we investigate the impact of the variance of weights and the spatial concentration of large weights in deep neural networks on neural persistence. In particular, we find that no relevant spatial structures is present in later layers of deep neural networks, and therefore neural persistence corresponds roughly linearly to the variance of weights, as effects of spatial structure become irrelevant.

Setup. To study neural persistence of deep neural networks in a controlled setting, we train DNNs with exhaustive combinations of the following hyperparameters: Number of hidden layers $\in \{1, 2, 3\}$, hidden size $\in \{50, 100, 250, 650\}$, and activation function $\in \{\tanh, \mathrm{relu}\}$. We use the Adam optimizer (Kingma & Ba, 2015) with the same hyperparameters as (Rieck et al., 2019), i.e. with a learning rate of 0.003, no weight decay, $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-8}$. Models are trained for 40 epochs with batch size 32 on three datasets, namely MNIST (LeCun et al., 1998), EMNIST (Cohen et al., 2017), and Fashion-MNIST (Xiao et al., 2017). For EMNIST, we use the *balanced* setting. EMNIST has more classes, namely 49 instead of 10 for MNIST and Fashion-MNIST. For each combination of hyperparameters and dataset, we train 20 copies with different initialisation and minibatch trajectories. Here, we analyse the models after training for 40 epochs.

Additionally, we train 20 linear classifiers (perceptrons) for each dataset using the same hyperparameters (optimiser, batch size, number of epochs) as for deep models.

Linear classifiers. Linear classifiers are similar to the synthetic weight matrices evaluated in Figure 1, as they also only contain one linear transformation. In Table 1, we show that linear classifiers trained on different datasets indeed exhibit relevant spatial structure. We find this by randomly permuting entries in weight matrices, which destroys any spatial structure that may have been present. In many cases, this causes large changes in neural persistence. Furthermore, linear regression fits with log-transformed variance of entries in weight matrices of trained linear classifiers as independent variable and neural persistence as dependent variable yield R^2 scores close to 1, which indicates a strong correspondence of the log-variance of trained weight matrices and the neural persistence. Taken together, these results show that the factors we identified to impact neural persistence are actually relevant in practise for linear classifiers, in the sense that they vary, in this case across datasets.

Spatial structure in deep networks. To analyse the spatial structure in trained deep neural networks, we again shuffle

	MNIST	Fashion-MNIST	EMNIST
Δ NP (Avg.)	0.07	0.17	0.18
R^2	0.994	0.993	0.996

Table 1: Δ NP (Avg.): Average difference in neural persistence when shuffling entries of weight matrices of linear classifiers trained on different datasets. R^2: R^2 scores of linear regression fits with log-transformed variance of entries in weight matrices of trained linear classifiers as independent variable and neural persistence as dependent variable.

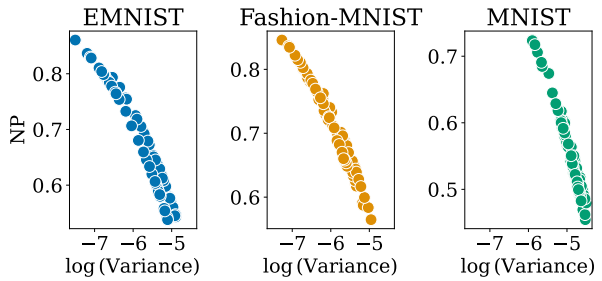

Figure 3: Roughly linear correspondence of log-variance of weights and neural persistence for deep neural networks.

entries in weight matrices and compare the resulting neural persistence values to those for the original weight matrices. Results for (absolute) changes in neural persistence are provided in Figure 2 and show that, irrespective of the dataset, the neural persistence of later layers in the network is insensitive to random permutation of entries. The difference between neural persistence of permuted matrices and the true neural persistence is mostly less than 0.02, which is a good bound for the variation of neural persistence values resulting from different initialisation and minibatch trajectories. These findings indicate the absence of any spatial structure in the large weights in later layers of trained neural networks that would be relevant for neural persistence.

Variance of weights of deep networks. Unlike spatial structure, differences in the variance of trained weights also exist in the case of DNNs. Therefore, our theoretical results suggest that, in the absence of relevant spatial structure, the variance of weights becomes the main factor that corresponds to changes in neural persistence. Indeed, we observe a roughly linear correspondence of neural persistence with the log-transformed global variance of all weights in the trained network, which is shown in Figure 3.

Due to effects of matrix size on neural persistence which are still present in normalised neural persistence, for better readability we only include neural networks with two or three hidden layers and hidden size $\in \{250, 650\}$ in Figure 3, but similar results can be observed for all models.

As further consequence, variance and neural persistence are in most cases highly correlated throughout training: The median Pearson correlation (among all models) is ≈ -0.98 (mean is ≈ -0.83). This implies that variance is similarly useful for the applications proposed by Rieck et al., especially using neural persistence as early stopping criterion.

4. Conclusion

In this work, we presented an analysis of the neural persistence measure. We showed both theoretically and empirically that the variance of weights and spatial concentration of large weights are the main factors impacting neural persistence. First, we derived new bounds on neural persistence which motivated the above mentioned factors. In practise, we found that later layers in trained deep feed-forward neural networks do not exhibit relevant spatial structure. Therefore, neural persistence is highly related to the variance of network weights, which is significantly cheaper to compute.

Acknowledgements: This work was supported by BMBF FKZ: 01IS18039A, DFG project number 276693517, by the ERC (853489 - DEXIM), and by EXC number 2064/1 – project number 390727645. Anders Christensen thanks the European Laboratory for Learning and Intelligent Systems (ELLIS) PhD program for support.

References

Cohen, G., Afshar, S., Tapson, J., and Van Schaik, A. Emnist: Extending mnist to handwritten letters. In *International joint conference on neural networks*, 2017.

Cohen-Steiner, D., Edelsbrunner, H., and Harer, J. Stability of persistence diagrams. *Discret. Comput. Geom.*, 2007.

Estivill-Castro, V. Generating nearly sorted sequences - the use of measures of disorder. In *The Australasian Theory Symposium*, 2004.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. In *ICLR*, 2015.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE*, 1998.

Rieck, B. On the expressivity of persistent homology in graph learning. *arXiv preprint arXiv:2302.09826*, 2023.

Rieck, B., Togninalli, M., Bock, C., Moor, M., Horn, M., Gumbsch, T., and Borgwardt, K. M. Neural persistence: A complexity measure for deep neural networks using algebraic topology. In *ICLR*, 2019.

Xiao, H., Rasul, K., and Vollgraf, R. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.