a higher degree of per-layer redundancy is expected, resulting in a different structural complexity. Overall, these results indicate that for a fixed architecture approaches targeted at increasing the neural persistence during the training process may be of particular interest.

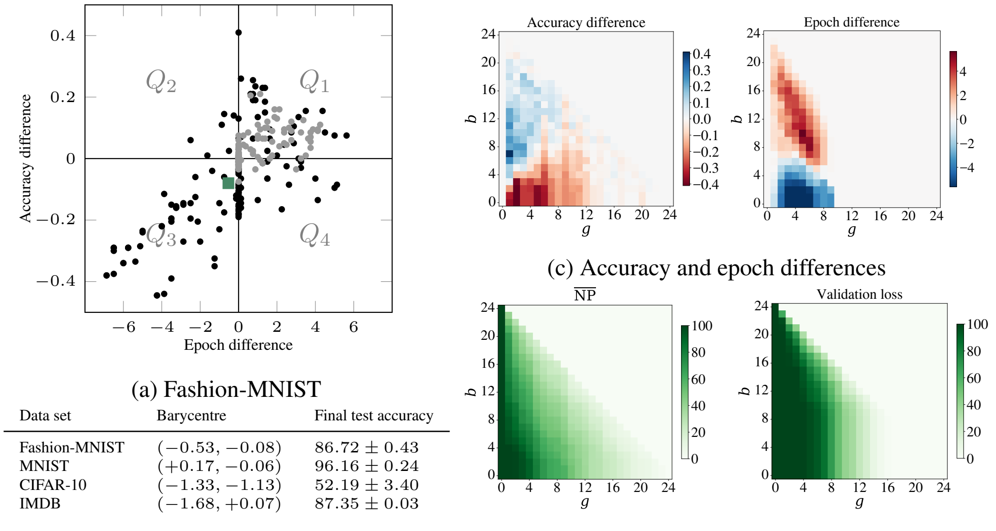
4.2 Early stopping based on neural persistence

Neural persistence can be used as an *early stopping* criterion that does not require a validation data set to prevent overfitting: if the mean normalized neural persistence does not increase by more than $\Delta_{\min}$ during a certain number of epochs g, the training process is stopped. This procedure is called 'patience' and Algorithm 2 describes it in detail. A similar variant of this algorithm, using validation loss instead of persistence, is the state-of-the-art for early stopping in training (Bengio, 2012; Chollet et al., 2015). To evaluate the efficacy of our measure, we compare it against validation loss in an extensive set of scenarios. More precisely, for a training process with at most G epochs, we define a $G \times G$ parameter grid consisting of the 'patience' parameter g and a burn-in rate b (both measured in epochs). b defines the number of epochs after which an early stopping criterion starts monitoring, thereby preventing underfitting. Subsequently, we set $\Delta_{\min} = 0$ for all measures to remain comparable and scale-invariant, as non-zero values could implicitly favour one of them due to scaling. For each data set, we perform 100 training runs of the same architecture, monitoring validation loss and mean normalized neural persistence every quarter epoch. The early stopping behaviour of both measures is simulated for each combination of b and g and their performance over all runs is summarized in terms of median test accuracy and median stopping epoch; if a criterion is not triggered for one run, we report the test accuracy at the end of the training and the number of training epochs. This results in a scatterplot, where each point (corresponding to a single parameter combination) shows the difference in epochs and the absolute difference in test accuracy (measured in percent). The quadrants permit an intuitive explanation: Q_2, for example, contains all configurations for which our measure stops *earlier*, while achieving a *higher* accuracy. Since b and g are typically chosen to be small in an early stopping scenario, we use grey points to indicate uncommon configurations for which b or g is larger than half of the total number of epochs. Furthermore, to summarize the performance of our measure, we calculate the barycentre of all configurations (green square).

Figure 4a depicts the comparison with validation loss for the 'Fashion-MNIST' (Xiao et al., 2017) data set; please refer to Section A.3 in the appendix for more data sets. Here, we observe that most common configurations are in Q_2 or in Q_3, i.e our criterion stops earlier. The barycentre is at $(-0.53, -0.08)$, showing that out of 625 configurations, on average we stop half an epoch earlier than validation loss, while losing virtually no accuracy (0.08%). Figure 4c depicts detailed differences in accuracy and epoch for our measure when compared to validation loss; each cell in a heatmap corresponds to a single parameter configuration of b and g. In the heatmap of accuracy differences, blue, white, and red represent parameter combinations for which we obtain *higher, equal, or lower* accuracy, respectively, than with validation loss for the same parameters. Similarly, in the heatmap of epoch differences, green represents parameter combinations for which we stop *earlier* than validation loss. For $b \leq 8$, we stop earlier (0.62 epochs on average), while losing only 0.06% accuracy. Finally,

Algorithm 2 Early stopping based on mean normalized neural persistence

```
Require: Weighted neural network $\mathcal{N}$, patience $g$, $\Delta_{\min}$
$P \leftarrow 0, G \leftarrow 0$                          ▷ Initialize highest observed value and patience counter
procedure EarlyStopping($\mathcal{N}$, $g$, $\Delta_{\min}$)    ▷ Callback that monitors training at every epoch
    $P' \leftarrow \overline{\mathrm{NP}}(\mathcal{N})$
    if $P' > P + \Delta_{\min}$ then        ▷ Update mean normalized neural persistence and reset counter
        $P \leftarrow P', G \leftarrow 0$
    else                                                              ▷ Update patience counter
        $G \leftarrow G + 1$
    end if
    if $G \geq g$ then                                      ▷ Patience criterion has been triggered
        return $P$                                   ▷ Stop training and return highest observed value
    end if
end procedure
```



(a) Fashion-MNIST

Data set	Barycentre	Final test accuracy
Fashion-MNIST	$(-0.53, -0.08)$	86.72 ± 0.43
MNIST	$(+0.17, -0.06)$	96.16 ± 0.24
CIFAR-10	$(-1.33, -1.13)$	52.19 ± 3.40
IMDB	$(-1.68, +0.07)$	87.35 ± 0.03

(b) Summary

(c) Accuracy and epoch differences

(d) Number of triggers

Figure 4: The visualizations depict the differences in accuracy and epoch for all comparison scenarios of mean normalized neural persistence versus validation loss, while the table summarizes the results on other data sets. Final test accuracies are shown irrespectively of early stopping to put the accuracy differences into context.

Figure 4d shows how often each measure is triggered. Ideally, each measure should consist of a dark green triangle, as this would indicate that *each* configuration stops all the time. For this data set, we observe that our method stops for more parameter combinations than validation loss, but not as frequently for all of them. To ensure comparability across scenarios, we did not use the validation data as additional training data when stopping with neural persistence; we refer to Section A.7 for additional experiments in data scarcity scenarios. We observe that our method stops earlier when overfitting can occur, and it stops later when longer training is beneficial.

5 Discussion

In this work, we presented *neural persistence*, a novel topological measure of the structural complexity of deep neural networks. We showed that this measure captures topological information that pertains to deep learning performance. Being rooted in a rich body of research, our measure is theoretically well-defined and, in contrast to previous work, generally applicable as well as computationally efficient. We showed that our measure correctly identifies networks that employ best practices such as dropout and batch normalization. Moreover, we developed an early stopping criterion that exhibits competitive performance while not relying on a separate validation data set. Thus, by saving valuable data for training, we managed to boost accuracy, which can be crucial for enabling deep learning in regimes of smaller sample sizes. Following Theorem 2, we also experimented with using the p-norm of *all* weights of the neural network as a proxy for neural persistence. However, this did not yield an early stopping measure because it was never triggered, thereby suggesting that neural persistence captures salient information that would otherwise be hidden among all the weights of a network. We extended our framework to convolutional neural networks (see Section A.4) by deriving a closed-form approximation, and observed that an early stopping criterion based on neural persistence for convolutional layers will require additional work. Furthermore, we conjecture that assessing dissimilarities of networks by means of persistence diagrams (making use of higher-dimensional topological features), for example, will lead to further insights regarding their generalization and learning abilities. Another interesting avenue for future research would concern the analysis of the 'function space' learned by a neural network. On a more general level, *neural persistence* demonstrates the great potential of topological data analysis in machine learning.

REFERENCES

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems: Simple, end-to-end, LeNet-5-like convolutional MNIST model example, 2015. URL `https://github.com/tensorflow/models/blob/master/tutorials/image/mnist/convolutional.py`.

Alessandro Achille and Stefano Soatto. Emergence of invariance and disentanglement in deep representations. *Journal of Machine Learning Research*, 18:1–34, 2018.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *International Conference on Learning Representations (ICLR)*, 2015.

Yoshua Bengio. Practical recommendations for gradient-based training of deep architectures. In Grégoire Montavon, Geneviève B. Orr, and Klaus-Robert Müller (eds.), *Neural Networks: Tricks of the Trade*, volume 7700 of *Lecture Notes in Computer Science*, pp. 437–478. Springer, Heidelberg, Germany, 2012.

Monica Bianchini and Franco Scarselli. On the complexity of neural network classifiers: A comparison between shallow and deep architectures. *IEEE Transactions on Neural Networks and Learning Systems*, 25(8):1553–1565, 2014.

Peter Bubenik. Statistical topological data analysis using persistence landscapes. *Journal of Machine Learning Research*, 16:77–102, 2015.

Gunnar Carlsson and Facundo Mémoli. Characterization, stability and convergence of hierarchical clustering methods. *Journal of Machine Learning Research*, 11:1425–1470, 2010.

Gunnar Carlsson, Tigran Ishkhanov, Vin de Silva, and Afra Zomorodian. On the local behavior of spaces of natural images. *International Journal of Computer Vision*, 76(1):1–12, 2008.

Corrie J. Carstens and Kathy J. Horadam. Persistent homology of collaboration networks. *Mathematical Problems in Engineering*, 2013:815035, 2013.

Travers Ching, Daniel S. Himmelstein, Brett K. Beaulieu-Jones, Alexandr A. Kalinin, Brian T. Do, Gregory P. Way, Enrico Ferrero, Paul-Michael Agapow, Michael Zietz, Michael M. Hoffman, Weil Xie, Gail L. Rosen, Benjamin J. Lengerich, Johnny Israeli, Jack Lanchantin, Stephen Woloszynek, Anne E. Carpenter, Avanti Shrikumar, Jinbo Xu, Evan M. Cofer, Christopher A. Lavender, Srinivas C. Turaga, Amr M. Alexandri, Zhiyong Lu, David J. Harris, Dave DeCaprio, Yanjun Qi, Anshul Kundaje, Yifan Peng, Laura K. Wiley, Marwin H.S. Segler, Simina M. Boca, S. Joshua Swamidass, Austin Huang, Anthony Gitter, and Casey S. Greene. Opportunities and obstacles for deep learning in biology and medicine. *Journal of The Royal Society Interface*, 15 (141):20170387, 2018.

François Chollet et al. Keras. `https://keras.io`, 2015.

David Cohen-Steiner, Herbert Edelsbrunner, and John Harer. Extending persistence using Poincaré and Lefschetz duality. *Foundations of Computational Mathematics*, 9(1):79–103, 2009.

David Cohen-Steiner, Herbert Edelsbrunner, John Harer, and Yuriy Mileyko. Lipschitz functions have L_p-stable persistence. *Foundations of Computational Mathematics*, 10(2):127–139, 2010.

Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to algorithms*. MIT Press, Cambridge, MA, USA, 3rd edition, 2009.

Herbert Edelsbrunner and John Harer. *Computational topology: An introduction*. American Mathematical Society, Providence, RI, USA, 2010.