TopoHyperDrive: Accelerating Meta-Search in Hyperparameter Optimization through Topological Analysis

Vladislav Zhuzhel [1]

Abstract

Neural networks have become an important part of the toolbox of data scientists, addressing a wide range of applications from text generation and image recognition to medical data analysis. As these networks increase in complexity and size, selecting optimal hyperparameters and architectures that is crucial for maximizing model performance becomes less feasible. Traditional methods for hyperparameter optimization often treat neural networks as black boxes, leading to lengthy and computationally expensive searches. In this paper, we introduce a novel approach that is called TopoHyperDrive[1] that leverages the topological structure of the model embeddings to accelerate the hyperparameter search process.

1. Introduction

Neural networks have experienced remarkable growth in their capabilities and applications over the last years, becoming an important part in a variety of fields such as text generation(Brown et al., 2020), facial recognition(Serengil & Ozpinar, 2020), financial forecasting(Cheng et al., 2022), and personalized healthcare(Ali et al., 2020). This expansion is fueled by substantial advancements in computational power, the accessibility of extensive datasets, and innovations in machine learning algorithms. As these networks grow in size and complexity, they push the boundaries of what artificial intelligence can accomplish in both research and real-world applications.However, as these models expand in scale, the demand on computational resources intensifies correspondingly. Tasks such as text generation or text-to-image synthesis now require extensive training periods, often spanning days or even months, even when employing multi-GPU setups equipped with state-of-the-art accelerators (Chen et al., 2023). This increasing complexity highlights the need for advanced meta-search techniques that can efficiently navigate the expansive hyperparameter space to optimize performance and reduce computational overhead.

Most existing meta-search methods treat neural networks as black boxes—systems that process inputs and yield outputs, which are then optimized based on specific metrics. This approach encompasses a broad range of techniques, from the basic and often impractical Grid Search to Random Search, and extends to more advanced strategies like Bayesian Optimization (Bergstra et al., 2011), Multi-fidelity Optimization (Li et al., 2018), and Gradient-based Approaches (Pedregosa, 2016). Except for gradient-based methods, these techniques generally do not account for the internal workings of the neural networks. Gradient-based methods, however, are applicable only when hyperparameter gradients can be explicitly defined, restricting their use to specific scenarios. Moreover, all these method rely on the metric quality and do not incorporate the inner structure of the obtained embeddings.

Recent advancements in foundation models using self-supervised learning (Brown et al., 2020; Luo et al., 2023; He et al., 2022) have shifted focus towards the quality of the embeddings themselves, rather than performance on specific tasks. This shift underscores the importance of studying the intrinsic properties of embeddings. Motivated by these developments and the limitations of existing methods, we introduce TopoHyperDrive, a topology-based method inspired by the Representation Topology Divergence (RTD) (Barannikov et al., 2021), designed to accelerate hyperparameter search. The contribution of this paper is as follows:

- Introduced a method that incorporates the topological features of model embeddings using the Representation Topology Divergence (RTD).
- Proposed several ways to use RTD in both metric-based and metric-free setups.
- Evaluated the convergence rate of hyperparameter searches using RTD and compared these with existing meta-parameter search techniques.

[1] Skolkovo Institute of Science and Technology, Moscow, Russia. Correspondence to: Vladislav Zhuzhel <vladislav.zhuzhel@skoltech.ru>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021. Copyright 2021 by the author(s).

[1] GitHub: https://github.com/VladislavZh/TopoHyperDrive

2. Related Works

2.1. Black-Box Optimization

As previously noted, the predominant approach to hyperparameter optimization involves black-box optimization. In this context, the neural network is assumed to be a black box that processes input and produces an output, which is then evaluated using a specific metric.

The simplest methods for identifying optimal hyperparameters include grid search, random search, or Sobol sequences (Bergstra & Bengio, 2012). These approaches do not leverage prior knowledge about the performance of previously trained models. They explore a predefined search space and ultimately select the best-performing configuration. Despite its straightforward implementation, this method is highly time-consuming, as it necessitates training models on setups that could be deemed suboptimal based on insights from past iterations.

The desire to leverage the performance data of previously trained models has led to the development of various methods designed to accelerate hyperparameter search. One of the most well-known is the Tree-structured Parzen Estimator (TPE) approach (Bergstra et al., 2013). This Bayesian-based method predicts the point with the greatest expected improvement in the objective metric at each step, following a few initial random steps to start the optimization process.

However, the TPE approach requires full training at each step of the hyperparameter optimization, which can be extremely time-consuming (Chen et al., 2023). To address this, one could consider eliminating underperforming candidates at the early stages of training. This led to the development of the Hyperband method (Li et al., 2018), which begins with a group of candidates and progressively discards the least effective ones based on performance, continuing the training process until only the optimal configuration remains. The BOHB method (Falkner et al., 2018) merges the principles of both the TPE and Hyperband approaches, addressing key shortcomings such as the extensive time requirements of the TPE method and the dependency on random initialization inherent in Hyperband.

2.2. Neural Network Representation Similarity Scores

Despite the effectiveness and robustness of black-box optimization methods, one might question whether it is possible to leverage the intrinsic structure of the neural network. Furthermore, in representation learning setups, choosing an appropriate objective metric can be challenging, and these metrics may not adequately reflect the true quality of the obtained embeddings. Therefore, we will explore methods to compare neural network representations. Specifically, we aim to develop a score that quantifies the similarity between two representations of a given dataset.

One possible metric for comparing neural network representations involves using summary statistics derived from Canonical Correlation Analysis (CCA) (Thompson, 2000). This method identifies bases for two matrices such that the projections of the datasets onto these bases yield maximized correlation. To assess similarity between representations, one can use the mean CCA correlation or apply CCA to the truncated singular value decomposition, which is less sensitive to perturbations, offering a more robust measure of similarity (Raghu et al., 2017).

The primary limitation of the CCA-based approach is its design to be invariant to any invertible linear transformation. Consequently, representations whose width exceeds the dataset size become indistinguishable (Kornblith et al., 2019). However, this limitation can be mitigated by restricting the invariance property to orthogonal transformations and isotropic scaling. This modification forms the basis for the Centered Kernel Alignment (CKA) score (Kornblith et al., 2019), which offers a more nuanced measure of representation similarity.

However, both of these approaches focus on the correlation between representations and fail to directly consider their inherent topological structures. This limitation has been addressed in (Barannikov et al., 2021) through the introduction of Representation Topology Divergence (RTD). This method evaluates the emergence and dissolution of topological features in a joint graph relative to one of the graphs within the simplicial approximation of the manifold, with edge lengths not exceeding α (Niyogi et al., 2008). In (Barannikov et al., 2021), it has been demonstrated that Representation Topology Divergence (RTD) surpasses Centered Kernel Alignment (CKA) in terms of identifying clusters as topological features and correlating with the measures of disagreement (Kuncheva & Whitaker, 2003).

3. Methods

Topological features are critically important for the quality of embeddings. In scenarios such as image classification, where the final layer atop the backbone is typically simplistic—often just a single linear layer—no new topological features are introduced. Consequently, it is essential for the embeddings themselves to encapsulate all topological features intrinsic to the data. For instance, in classification tasks, the embeddings should naturally form clusters corresponding to each class, a goal actively pursued in contrastive learning (Khosla et al., 2020; Chen et al., 2020; Deng et al., 2019).

Consequently, we will employ the Representation Topology Divergence (RTD) (Barannikov et al., 2021) to assess the quality of embeddings and accelerate the hyperparameter search process introducing TopoHyperDrive.

3.1. Representation Topology Divergence

Representation Topology Divergence (RTD) quantitatively assesses the dissimilarity between two neural network embeddings, represented as point clouds P and P', which each comprise an equal number of data points but may reside in different ambient spaces. This comparison leverages topological features identified through a Topological Data Analysis (TDA) tool known as the R-Cross-Barcode (Barannikov et al., 2021). The R-Cross-Barcode represents the emergence and disappearance of topological features by analyzing differences between the Vietoris-Rips filtered complexes of the individual and joined graphs for the point clouds.

The RTD score is computed using the following formula:

$$\mathrm{RTD}(P, P') = \sum_i \mathrm{length}(\mathrm{Bar}_i)$$

Here, Bar_i denotes the barcode intervals, which encapsulate the persistence of topological features across scales. A higher RTD score indicates a greater degree of topological dissimilarity between the embeddings, reflecting significant structural variances.

This methodology not only enables a nuanced exploration of the structural differences or similarities between model representations but also provides a robust framework for hyperparameter optimization. By exploiting the underlying topological properties of model embeddings, the RTD approach enhances the capacity to discern and utilize complex patterns within data, thereby improving model training and performance evaluation.

3.2. TopoHyperDrive variants

Since RTD is a pairwise score, it raises the question of how one can effectively leverage it to advance the optimization process. We propose three possible methods. All these methods can be equipped with any other multi-objective hyperparameters search methods and the ”One to Noise” can be applied in sole objective setup.

3.2.1. ”One to Noise” Divergence

A straightforward approach is to compare the obtained embeddings with random noise. Ideally, we aim for our embeddings to be as distinct as possible, thereby introducing numerous topological features. In this scenario, one could either use this value as the sole objective or optimize the model based on two objectives simultaneously. However, a potential drawback of this approach is that the model might introduce new topological features that fail to generalize across the data, thereby leading to overfitting.

3.2.2. ”One to Top Performer” Divergence

An alternative approach involves identifying a model that is as divergent as possible from the current best model for the next hyperparameters optimization step. The underlying assumption is that models with similar configurations are likely to yield similar performance levels. However, employing this metric as the sole objective is impractical, as it may result in oscillating between different variants of topological features without a clear method to determine the optimal variant. Consequently, it becomes necessary to optimize the model based on dual objectives.

3.2.3. ”One to All” Divergence

The final approach we propose seeks to configure hyperparameters such that the resulting model diverges maximally from all previous iterations, based on the mean value of all scores. This strategy aims to explore a broad range of representation topologies, facilitating the discovery of the most effective one in relation to a specific objective function. Similar to the ”One to Top Performer” approach, identifying the optimal topology is contingent upon the presence of a well-defined objective function.

4. Experiments

4.1. Data

In our study, the experimental validation of the TopoHyperDrive method was carried out using the CIFAR-100 dataset. This dataset is a well-established benchmark in the machine learning community, containing 60,000 color images categorized into 100 classes, with 600 images per class. Each image is of 32x32 pixels, representing various objects and living entities, making it a challenging dataset for image recognition tasks.

4.2. Baseline Methods

In our comparative analysis, we employed several baseline methods alongside the proposed TopoHyperDrive approach to demonstrate its effectiveness in optimizing neural network hyperparameters. These baseline methods, commonly used in hyperparameter optimization, include Random Search, TPE (Tree-structured Parzen Estimator) Sampler (Bergstra et al., 2013), Hyperband (Li et al., 2018), and BOHB (Bayesian Optimization and Hyperband) (Falkner et al., 2018).

- Random Search serves as a straightforward baseline, where hyperparameters are selected randomly from a predefined search space.

- TPE Sampler, or Tree-structured Parzen Estimator, represents a more sophisticated approach. It models the probability of achieving more promising results and uses these probabilistic models to select the most promising hyperparameters to evaluate in subsequent trials. This method typically outperforms Random Search as it builds on prior observations to refine the search dynamically.

- Hyperband is an optimization framework that accelerates the evaluation process by adaptively allocating resources to more promising configurations. It uses a bandit-based approach to allocate a budget and aggressively stops poor-performing configurations early, which makes it highly efficient, especially in scenarios with limited computational resources.
- BOHB combines the strengths of Bayesian optimization and Hyperband, utilizing TPE to guide the selection of configurations and Hyperband to determine the number of resources allocated to each configuration. This hybrid method leverages the efficiency of Hyperband and the guided search capability of Bayesian optimization, making it particularly powerful for finding optimal hyperparameters quickly and effectively.

These baseline methods provided a comprehensive backdrop for evaluating the efficacy of our RTD-based approach, highlighting its advantages in leveraging the topological properties of model embeddings to guide the search process more insightfully.

4.3. Search Space

For our search space, we adopt a VGG-like architectural model, drawing inspiration from (Simonyan & Zisserman, 2014). Our configuration consists of a five-block structure, where each block comprises several convolutional layers equipped with a variable number of filters. Max pooling layers are interspersed between these blocks to reduce spatial dimensions progressively. The architecture is further extended with fully connected layers positioned before the softmax output layer. The hyperparameters optimized in this setup include:

- The number of layers within each convolutional block.
- The number of convolutional filters in each layer of every block. These number are fixed within every block.
- The size of the hidden units in the fully connected layers.

This approach allows us to explore a wide range of configurations and to fine-tune the network architecture for optimal performance on the given task.

4.4. Experimental Setup

In our experiments, apart from the hyperparameters specified in the previous section, we standardized several other parameters to ensure consistency across different model configurations:

- **Batch Size:** We set the batch size to 1024 for training all models.
- **Optimizer:** We used the Adam optimizer with a learning rate of 0.001 and a weight decay of 1×10^{-6}.
- **Learning Rate Scheduler:** A Reduce-on-Plateau scheduler was implemented to adjust the learning rate by a factor of 0.1 if there was no improvement in performance for 5 epochs, aiding in finer adjustments towards the latter stages of training.
- **Early Stopping:** To prevent overfitting and to save computational resources, an early stopping mechanism was employed, which halts training if there is no improvement in the validation metrics for 15 consecutive epochs.
- **Number of epochs:** The maximum number of epochs for all the experiments was set to 100.

These settings were chosen to provide a robust framework for evaluating the effectiveness of different hyperparameter configurations while maintaining efficient use of computational resources. All the experiments were conducted on RTX-4090.

4.5. Results

The results can be found in Figure 1.

References

Ali, F., El-Sappagh, S., Islam, S. R., Kwak, D., Ali, A., Imran, M., and Kwak, K.-S. A smart healthcare monitoring system for heart disease prediction based on ensemble deep learning and feature fusion. *Information Fusion*, 63: 208–222, 2020.

Barannikov, S., Trofimov, I., Balabin, N., and Burnaev, E. Representation topology divergence: A method for comparing neural network representations. *arXiv preprint arXiv:2201.00058*, 2021.

Bergstra, J. and Bengio, Y. Random search for hyper-parameter optimization. *Journal of machine learning research*, 13(2), 2012.

Bergstra, J., Bardenet, R., Bengio, Y., and Kégl, B. Algorithms for hyper-parameter optimization. *Advances in neural information processing systems*, 24, 2011.

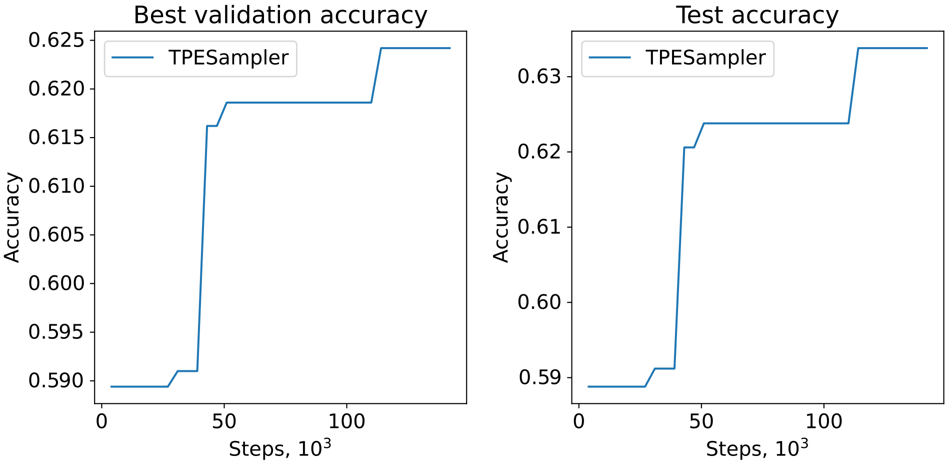

Figure 1. Best validation accuracy of the hyper parameter optimization methods on the number of steps and the corresponding test accuracy.

Bergstra, J., Yamins, D., and Cox, D. Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures. In *International conference on machine learning*, pp. 115–123. PMLR, 2013.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901, 2020.

Chen, J., Yu, J., Ge, C., Yao, L., Xie, E., Wu, Y., Wang, Z., Kwok, J., Luo, P., Lu, H., et al. Pixart-α: Fast training of diffusion transformer for photorealistic text-to-image synthesis. *arXiv preprint arXiv:2310.00426*, 2023.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pp. 1597–1607. PMLR, 2020.

Cheng, D., Yang, F., Xiang, S., and Liu, J. Financial time series forecasting with multi-modality graph neural network. *Pattern Recognition*, 121:108218, 2022.

Deng, J., Guo, J., Xue, N., and Zafeiriou, S. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4690–4699, 2019.

Falkner, S., Klein, A., and Hutter, F. Bohb: Robust and efficient hyperparameter optimization at scale. In *International conference on machine learning*, pp. 1437–1446. PMLR, 2018.

He, K., Chen, X., Xie, S., Li, Y., Dollár, P., and Girshick, R. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 16000–16009, 2022.

Khosla, P., Teterwak, P., Wang, C., Sarna, A., Tian, Y., Isola, P., Maschinot, A., Liu, C., and Krishnan, D. Supervised contrastive learning. *Advances in neural information processing systems*, 33:18661–18673, 2020.

Kornblith, S., Norouzi, M., Lee, H., and Hinton, G. Similarity of neural network representations revisited. In *International conference on machine learning*, pp. 3519–3529. PMLR, 2019.

Kuncheva, L. I. and Whitaker, C. J. Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. *Machine learning*, 51:181–207, 2003.

Li, L., Jamieson, K., DeSalvo, G., Rostamizadeh, A., and Talwalkar, A. Hyperband: A novel bandit-based approach to hyperparameter optimization. *Journal of Machine Learning Research*, 18(185):1–52, 2018.

Luo, D., Cheng, W., Wang, Y., Xu, D., Ni, J., Yu, W., Zhang, X., Liu, Y., Chen, Y., Chen, H., et al. Time series contrastive learning with information-aware augmentations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 4534–4542, 2023.

Niyogi, P., Smale, S., and Weinberger, S. Finding the homology of submanifolds with high confidence from random samples. *Discrete & Computational Geometry*, 39:419–441, 2008.

Pedregosa, F. Hyperparameter optimization with approximate gradient. In *International conference on machine learning*, pp. 737–746. PMLR, 2016.

Raghu, M., Gilmer, J., Yosinski, J., and Sohl-Dickstein, J. Svcca: Singular vector canonical correlation analysis for deep learning dynamics and interpretability. *Advances in neural information processing systems*, 30, 2017.

Serengil, S. I. and Ozpinar, A. Lightface: A hybrid deep face recognition framework. In *2020 innovations in intelligent systems and applications conference (ASYU)*, pp. 1–5. IEEE, 2020.

Simonyan, K. and Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Thompson, B. Canonical correlation analysis. 2000.