A Model Zoo Generation Details

In our model zoos, we use three architectures. Two of them rely on a general CNN architecture, the third is a common ResNet-18[20]. For the first two architectures, use the general CNN architecture in two sizes, detailed in Table 4. By varying different generating factors listed in Table 1, we create a grid of configurations, where each node represents a model. Each node is instantiated as a model and trained with the exact same training protocol. We chose the hyperparameters with diversity in mind. The ranges for each of the generating factors are chosen such that they can lead to functioning models with a corresponding set of other generating factors. Nonetheless, that leads to some nodes with uncommon and less than promising configurations.

The code to generate the models can be found on www.modelzoos.cc. With that code, the model zoos can be replicated, changed or extended. We trained our model zoos on CPU nodes with up to 64 CPUs. Training a zoo takes between 3h (small models, small configuration and small dataset) and 3 days (large models, large configuration and large dataset). Overall, the generation of the zoos took around 30'000 CPU hours.

Table 4: CNN architecture details for the models in model zoos.

Layer	**Component**	**CNN small**	**CNN large**
Conv 1	input channels	1 or 3	3
	output channels	8	16
	kernel size	5	3
	stride	1	1
	padding	0	0
Max Pooling	kernel size	2	2
Activation			
Conv 2	input channels	8	16
	output channels	6	32
	kernel size	5	3
	stride	1	1
	padding	0	0
Max Pooling	kernel size	2	2
Activation			
Conv 3	input channels	6	32
	output channels	4	15
	kernel size	2	3
	stride	1	1
	padding	0	0
Activation			
Linear 1	input channels	36	60
	output channels	20	20
Activation			
Linear 2	input channels	20	20
	output channels	10	10
Total Parameters		2464 or 2864	10853

B Data Management and Accessibility of Model Zoos

Data Management and Documentation: To ensure that every zoo is reproducible, expandable, and understandable, we document each zoo. For each zoo, a Readme file is generated, displaying basic information about the zoo. The exact search pattern and the training protocol used to train the zoo is saved in a in a machine-readable json file. To make the zoos expandable, the dataset used to train the zoo and a file describing the model architecture are included. The model class definition in pytorch is included with the zoo. Each model is saved along with a json file containing its exact hyperparameter combination. A second json file contains the the performance metrics during training. Model checkpoints are saved for every epoch. To enable further training of the models in the zoo, a checkpoint recording the optimizer state is saved for the final epoch of each model. All data can be found on the model zoo website as well directly from Zenodo.

Accessibility: We ensure the technical accessibility of the data by hosting it on Zenodo, where the data will be hosted for at least 20 years. Further, we take steps to reduce access barriers by providing code for data loading and preprocessing. With that we reduce the friction associated with analyzing of the raw zoo files. Further, it improves consistency by reducing errors associated with extracting information from the zoo. To that end, we provide a PyTorch dataset class encapsulating all model zoos for easy and quick access within the PyTorch framework. A Tensorflow counterpart will follow. All code can be found on the model zoo website as well as a code repository on github. To ensure conceptional accessibility, we include detailed insights, visualizations and the analysis of the model zoo (Sec. 4) with each zoo. Mode details can be found on the dataset website www.modelzoos.cc.

C Dataset Documentation and Intended Uses

The main dataset documentation can be found at www.modelzoos.cc and is detailed in the paper in Section 3.4. There, we provide links to the zoos, which are hosted on Zenodo as well as analysis of the zoos. In the future, the analysis will be systematically extended. The documentation includes code to reproduce, adapt or extend the zoos, code to reproduce the benchmark results, as well as code to load and preprocess the datasets. Dataset Metadata and DOIs are automatically provided by Zenodo, which also guarantees the long-term availability of the data. Files are stored as `zip`, `json` and `pt` (pytorch) files. All libraries to read and use the files are common and open source. We provide the code necessary to read and interpret the data.

The datasets are synthetic and intended to investigate populations of neural network models, i.e., to develop or evaluate model analysis methods, progress the understanding of learning dynamics, serve as datasets for representation learning on neural network models, or as a basis for new model generation methods. More information regarding the usage is given in the paper.

D Author Statement

The dataset is publicly available under www.modelzoos.cc and licensed under the Creative Commons Attribution 4.0 International license (CC-BY 4.0). The authors state that they bear responsibility under the CC-BY 4.0 license.

E Hosting, Licensing, and Maintenance Plan

The dataset is publicly available under www.modelzoos.cc and licensed under the Creative Commons Attribution 4.0 International license (CC-BY 4.0). The landing page contains documentation, code and references to the datasets, as detailed in the paper in Section 3.4. The datasets are hosted on Zenodo, to ensure (i) long-term availability (at least 20 years), (ii) automatic searchable dataset meta data, (iii) DOIs for dataset, and (iv) dataset versioning. The authors will maintain the datasets, but invite the community to engage. Code to recreate, correct, adapt, or extend the datasets is provided, s.t. maintenance can be taken over by the community at need. The github repository allows the community to discuss, interact, add or change code.