Model-invariant weight distribution descriptors for visual exploration of neural networks en masse

Supplementary material

G. Eilertsen, J. Jönsson, J. Unger, and A. Ynnerman

Linköping University, Sweden

1. Introduction

Here, we provide some complementary experiments for assessing the invariance of weight descriptors under different types of variations, as well as examples of descriptors for different variations in networks and configurations. We refer to the main paper for details on how the descriptors are formulated.

2. Permutation invariance

Neural networks experience permutation symmetries, such that the mapping performed by a network can be represented by different parameterizations [HN90]. One of the central properties of a successful weight representation is that it should be robust to different parameterizations of the same model – it is the mapping we want to represent, irrespective of its specific parameterization. To test this property, we produce artificial permutations of weights, by randomly swapping neurons in fully connected layers or channels in convolutional layers. By compensating for this swapping in the subsequent layer, the exact mapping of the network is maintained but with a different parameterization. We refer to [NSA*23] for a formal description of the permutation operation on MLPs, although we here also extend this to include convolutional layers. We perform this permutation operation for each neuron/channel throughout a network. To evaluate a representation's robustness to permutations, we generate 10 different parameterizations of each network, and measure the average ratio between representations of permuted weights and weights from different models. That is, a perfectly permutation invariant representation will have ratio 0 while a representation that have similar distances between permutations as between different models will have a ratio of 1. The results are shown in Table 1 for different dimensionality of the representations. The global statistics have best invariance, but do not encode enough information. Raw and random projection have overall highest numbers, while PCA has a ratio that increases with the number of components utilized. The MI-WD descriptor has the best ratio for higher dimensionality.

Dim.	Global stat.	Raw	PCA	Rand. proj.	MI-WD desc.
16	**0.024**	0.315	0.049	0.347	0.125
64	-	0.320	**0.076**	0.320	0.084
256	-	0.287	0.145	0.329	**0.082**
1024	-	0.247	0.317	0.326	**0.113**

Table 1: *Ratio between intra and inter distances of different weight representations, where intra distances measure the dissimilarity between different parameterizations of the same model.*

3. Examples

Fig. 1-4 show examples of weight descriptors for networks trained with the same architecture and hyperparameters but different random seeds. We expect these to yield similar representations, although not exactly the same due to the stochastic differences in the optimization. Also included are examples of the weight permutations used in Section 2. Optimally, we want different permutations of the same network resulting in the same descriptor. As seen in the figures, all descriptors are highly similar for a certain setting, although different seed generally generates slightly larger variations, which is expected. The closer similarity of the descriptors in Fig. 2 and Fig. 4, compared to Fig. 1 and Fig. 3, points to how their similar settings in terms of optimizer, activation function, and initialization yields higher similarity (although there are distinct differences).

Fig. 5 shows two examples of how the descriptors change during training. The examples start from a similar initialization point (Glorot normal), but converge to different weights during optimization. Fig. 6 shows the same trained models as in Fig. 5, at different levels of descriptor size. While larger descriptors reveal more details of the weight statistics, the default size (32×32) used in our experiments provides a good trade-off between details and descriptor size which captures the higher level properties of the weights.

References

[HN90] HECHT-NIELSEN R.: On the algebraic structure of feedforward network weight spaces. In *Advanced Neural Computers*. Elsevier, 1990, pp. 129–135. 1

[NSA*23] NAVON A., SHAMSIAN A., ACHITUVE I., FETAYA E., CHECHIK G., MARON H.: Equivariant architectures for learning in deep weight spaces. In *International Conference on Machine Learning* (2023), PMLR, pp. 25790–25816. 1

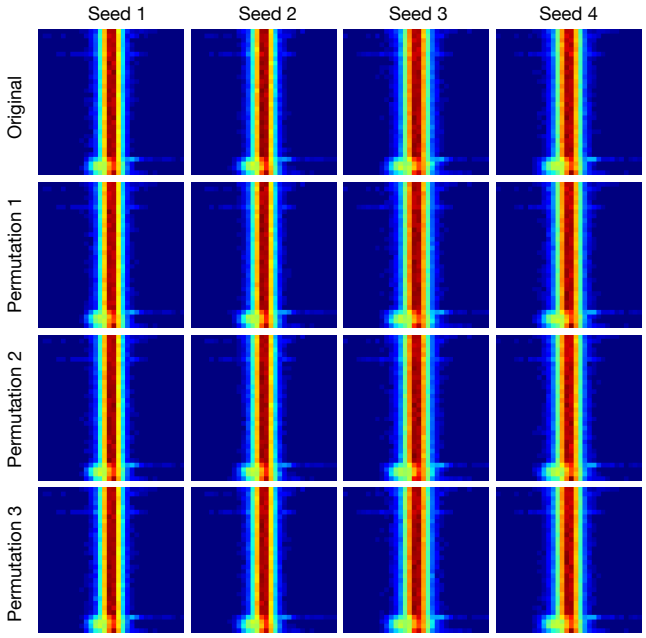

Figure 1: *Descriptors for networks trained with the same hyperparameters (dataset: **SVHN**, batchsize: **256**, optimizer: **RMSProp**, activation function: **ELU**, augmentations: **off**, initialization: **Glorot uniform**), but with different random seeds (columns). For each seed, the network weights have been randomly permuted for different parameterizations of the same function (rows).*

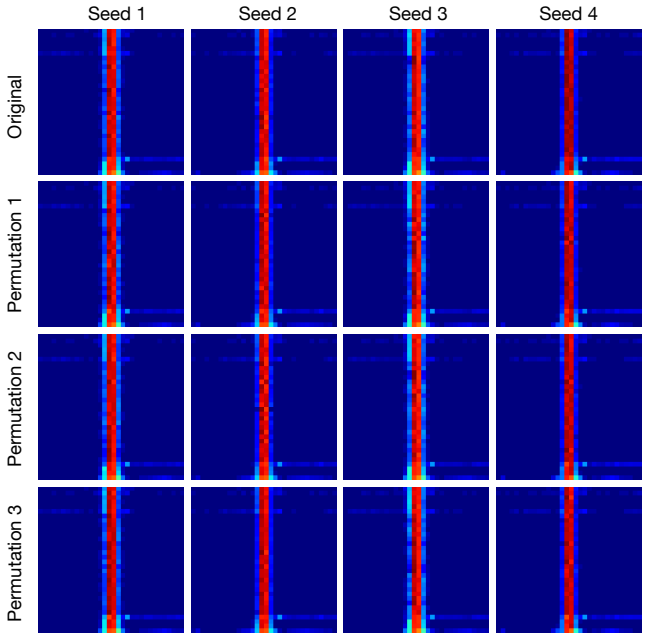

Figure 2: *Descriptors for networks trained with the same hyperparameters (dataset: **STL-10**, batchsize: **128**, optimizer: **RMSProp**, activation function: **ReLU**, augmentations: **on**, initialization: **Glorot uniform**), but with different random seeds (columns). For each seed, the network weights have been randomly permuted for different parameterizations of the same function (rows).*

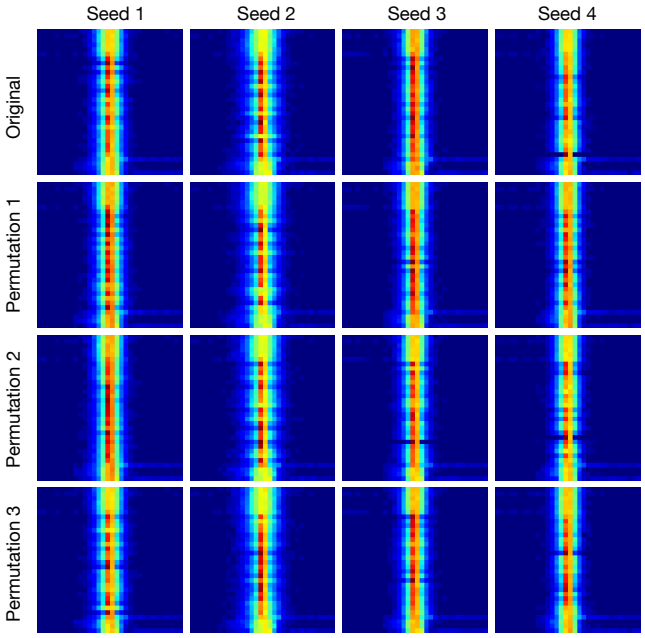

Figure 3: *Descriptors for networks trained with the same hyperparameters (dataset: **CIFAR-10**, batchsize: **32**, optimizer: **ADAM**, activation function: **ReLU**, augmentations: **off**, initialization: **Glorot uniform**), but with different random seeds (columns). For each seed, the network weights have been randomly permuted for different parameterizations of the same function (rows).*

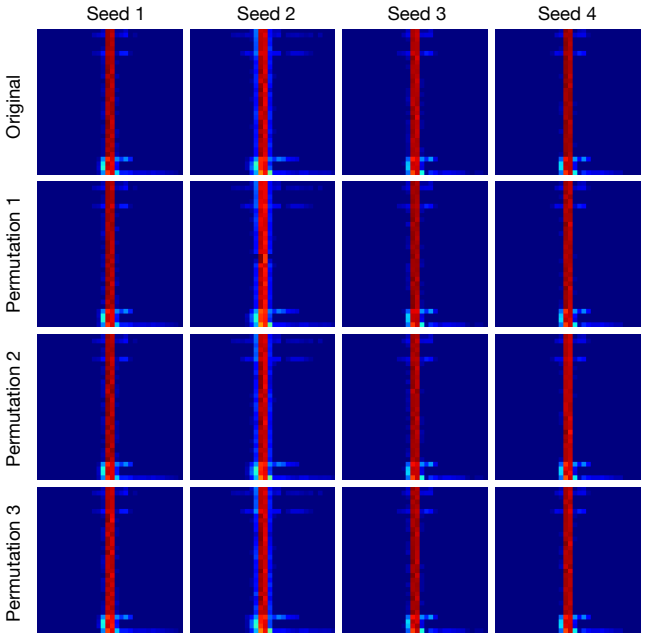

Figure 4: *Descriptors for networks trained with the same hyperparameters (dataset: **SVHN**, batchsize: **64**, optimizer: **RMSProp**, activation function: **ReLU**, augmentations: **off**, initialization: **Glorot uniform**), but with different random seeds (columns). For each seed, the network weights have been randomly permuted for different parameterizations of the same function (rows).*

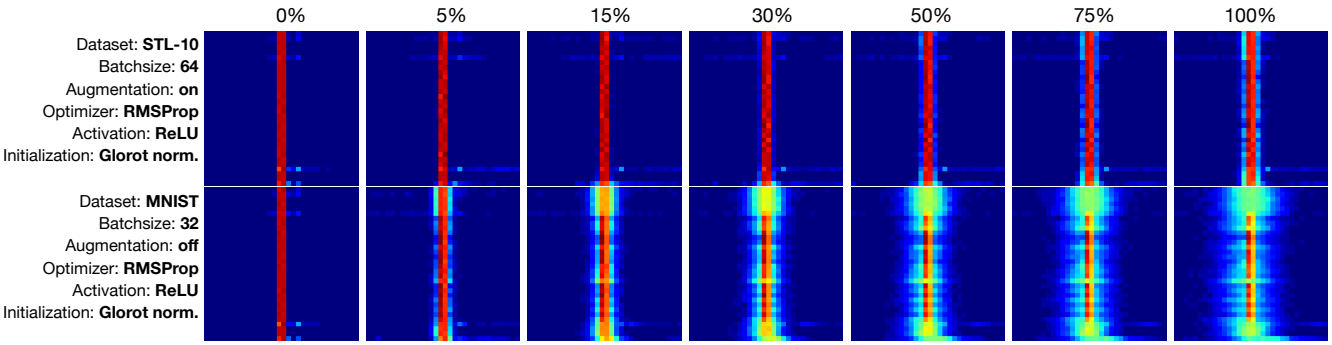

Figure 5: *Descriptors for networks during training, from initialization point (0%) to convergence (100%). The two examples use the same network architecture and start from similar initialization, but optimize toward different models.*

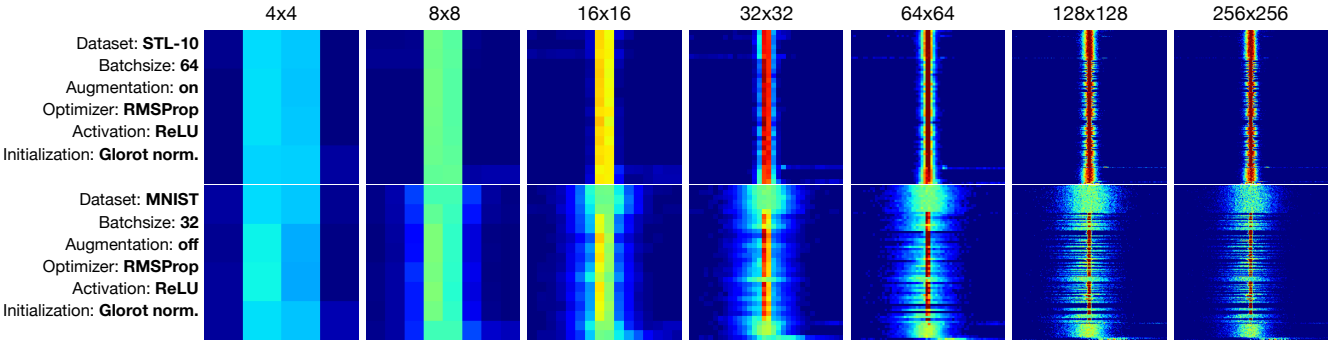

Figure 6: *Descriptors for different granularity $S \times B$, where S is the local window of weights and B is the number of histogram bins, i.e. the top and bottom of each image shows the weight distribution for shallow and deep layers, respectively. The networks are the same as in Fig. 5.*