Feature Learning and Random Features in Standard Finite-Width Convolutional Neural Networks: An Empirical Study (Supplementary Material)

Maxim Samarin[1] **Volker Roth**[1] **David Belius**[1]

[1]Department of Mathematics and Computer Science, University of Basel, Switzerland

A SUPPLEMENTARY MATERIAL

A.1 SNAKES DATASET

For a challenging classification task, we chose a subset of ImageNet 2012 [Russakovsky et al., 2015] comprised of ten snake categories illustrated in Fig. A.1. The extracted dataset contains 1300 training and 50 test images per class, resulting in 13000 train and 500 test images in total. As a benchmark performance results, we evaluate a standard pre-trained AlexNet on this dataset, achieving 47.6% test accuracy. Training our implementation of AlexNet with cross-entropy loss on the snakes dataset provides 98.5% train and 51.4% test accuracy (single run). In our experiments, we used a mean squared error loss, which in comparison led to 99.1% train and 53.8% test accuracy (single run). These results indicate that both loss functions lead to comparable performance and outperform a standard pre-trained AlexNet (trained on full ImageNet) with respect to generalization.

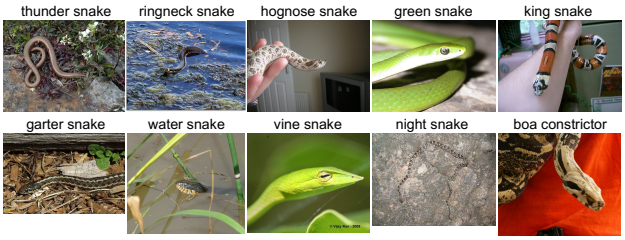

Figure A.1: Ten snake categories from ImageNet.

A.2 EARLY TRAINING TRAJECTORY

Following Lee et al. [2019], we study training trajectories for data samples x of the MNIST test set during training. For illustration, we plot the iteration $t = 0, 1, \ldots$ against the standard LeNet output $f^l(w_t, x)$ and the linearization $f^l_{\text{lin}}(u_t, x)$ for different widths. Note that w_t and u_t are the weights after t gradient updates for LeNet and LinLeNet trained on MNIST. As we use one-hot encoding, output l denotes the predicted output for the correct class of the data point x. The same hyperparameters as for the other MNIST experiments are used (see Sec. 4.1). A fixed random seed ensures that both LeNet and LinLeNet at a particular width factor are initialized exactly the same and receive the same mini-batches during training. Exemplary results are shown in Fig. A.2 for a digit 8 of the test set, with similar results being obtained for other samples, too. At small widths, training trajectories immediately diverge. With increasing width, the curves behave more similar; they are however not close in a path-wise sense, but the statistics of training trajectories become more alike.

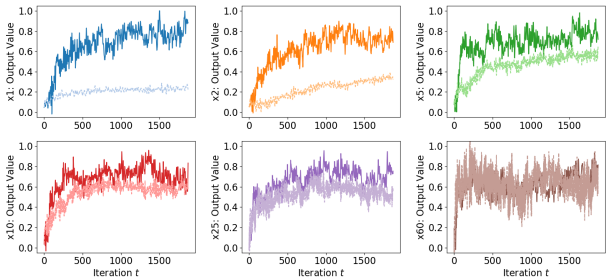

Figure A.2: Training trajectories of LeNet (dark) and LinLeNet (light) do not stay close for small width factors. Shown are the output values during training for the same MNIST input example from the test set at different widths.

A.3 EFFECTIVE RANK

The effective rank was introduced by Roy and Vetterli [2007] and can be viewed as the exponential entropy of normalized singular values. We restate the main definition in the following.

Definition. Let A be a complex-valued non-all-zero matrix of size $M \times N$ with (real positive) singular values $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_Q \geq 0$, where $Q = \min\{M, N\}$. Let $\sigma =$

Accepted for the 38th Conference on Uncertainty in Artificial Intelligence (UAI 2022).

$(\sigma_1, \sigma_2, ..., \sigma_Q)^\top$ and the singular value distribution be

$$p_k = \frac{\sigma_k}{\sum_{j=1}^{Q} \sigma_j} \quad \text{with } k = 1, 2, ..., Q. \tag{1}$$

The effective rank of matrix A is then defined as

$$\text{erank}(A) := \exp\left(H(p_1, p_2, ..., p_Q)\right) \tag{2}$$

where $H(p_1, p_2, ..., p_Q)$ is the Shannon entropy

$$H(p_1, p_2, ..., p_Q) = -\sum_{k=1}^{Q} p_k \log p_k. \tag{3}$$

In comparison to the usual notion of rank, an important property of the effective rank is that $erank(A) \leq rank(A)$ [Roy and Vetterli, 2007].

A.4 SINGULAR VALUES LINALEXNET

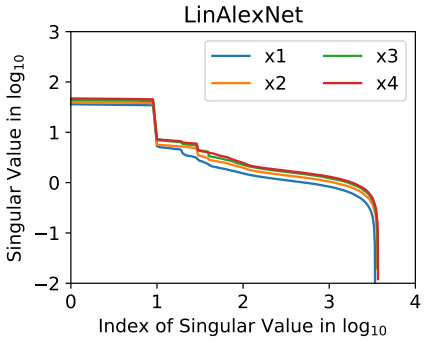

Figure A.3: Singular value distribution of LinAlexNet for 600 samples of the snakes dataset.

A.5 TRAIN AND TEST ACCURACY VALUES

Tables A.1 and A.2 provide the values for train and test accuracy in the AlexNet and LeNet experiments, respectively. The sample mean accuracy as well as sample standard deviation for 5 independent reruns of models are shown.

Table A.1: Mean accuracy and standard deviation for 5 independent reruns of the AlexNet experiment (see Fig. 5).

		×1	×2	×3	×4
Test	Lin. 0.1	30.48±1.68	33.52±0.63	33.48±1.14	33.88±1.68
	Lin. 1.0	33.64±1.26	35.2±1.53	36.76±1.52	36.6±2.05
	LeNet	54.04±1.13	54.72±1.27	54.72±0.98	55.16±1.54
Train	Lin. 0.1	36.75±0.5	42.62±0.8	46.64±0.54	49.84±0.95
	Lin. 1.0	51.05±2.17	64.45±3.12	72.63±3.02	79.45±3.12
	LeNet	99.15±0.05	99.19±0.03	99.16±0.05	99.16±0.04

Table A.2: Mean accuracy and standard deviation for 5 independent reruns of the LeNet experiments

		×1	×2	×5
MNIST (see Fig. 2)				
Test	Lin.	94.48±1.04	96.42±0.17	97.86±0.02
	LeNet	99.15±0.1	99.29±0.05	99.38±0.06
Train	Lin.	94.3±1.07	96.56±0.33	98.14±0.07
	LeNet	99.86±0.02	99.94±0.01	99.95±0.01
MNIST with translation (see Fig. 3)				
Test	Lin.	69.95±5.86	80.91±1.77	89.57±0.49
	LeNet	97.6±0.16	98.35±0.12	98.61±0.03
Train	Lin.	68.93±5.44	80.54±1.64	88.89±0.29
	LeNet	97.55±0.05	98.28±0.06	98.59±0.03
CIFAR-10 (see Fig. 4)				
Test	Lin.	42.98±0.53	48.07±1.12	54.42±0.4
	LeNet	63.2±0.58	69.58±0.48	75.76±0.22
Train	Lin.	43.99±0.97	50.3±1.52	60.45±1.14
	LeNet	92.07±0.24	98.53±0.07	99.76±0.04

		×10	×25	×60
MNIST (see Fig. 2)				
Test	Lin.	98.36±0.04	98.72±0.05	98.91±0.1
	LeNet	99.4±0.03	99.42±0.04	99.39±0.06
Train	Lin.	98.82±0.05	99.45±0.02	99.83±0.03
	LeNet	99.97±0.00	99.97±0.01	99.97±0.01
MNIST with translation (see Fig. 3)				
Test	Lin.	92.13±0.11	94.25±0.48	94.66±0.58
	LeNet	98.63±0.13	98.63±0.33	98.57±0.12
Train	Lin.	91.76±0.13	94.0±0.1	94.42±0.29
	LeNet	98.71±0.05	98.76±0.05	98.77±0.03
CIFAR-10 (see Fig. 4)				
Test	Lin.	58.23±0.43	62.47±0.26	65.8±0.23
	LeNet	77.56±0.21	78.83±0.1	78.97±0.13
Train	Lin.	68.32±0.92	81.24±0.74	93.84±0.33
	LeNet	99.92±0.01	99.96±0.00	99.98±0.00