Measuring the Data Efficiency of Deep Learning Methods


Hlynur Davíð Hlynsson
Ruhr-Universität Bochum


Alberto N. Escalante B.
Ruhr-Universität Bochum


Laurenz Wiskott
Ruhr-Universität Bochum

Measuring the Data Efficiency of Deep Learning Methods

Hlynur Davíð Hlynsson, Alberto N. Escalante-B. and Laurenz Wiskott

Institut für Neuroinformatik, Ruhr-University Bochum
{firstname.lastname}@ini.rub.de

Main Idea

How would you measure the data efficiency — performance as a function of training set size — of a learning algorithm? It seems natural to:

- Vary the size of homogeneous data and measure performance.
- Next, ramp up the variability of the training data.

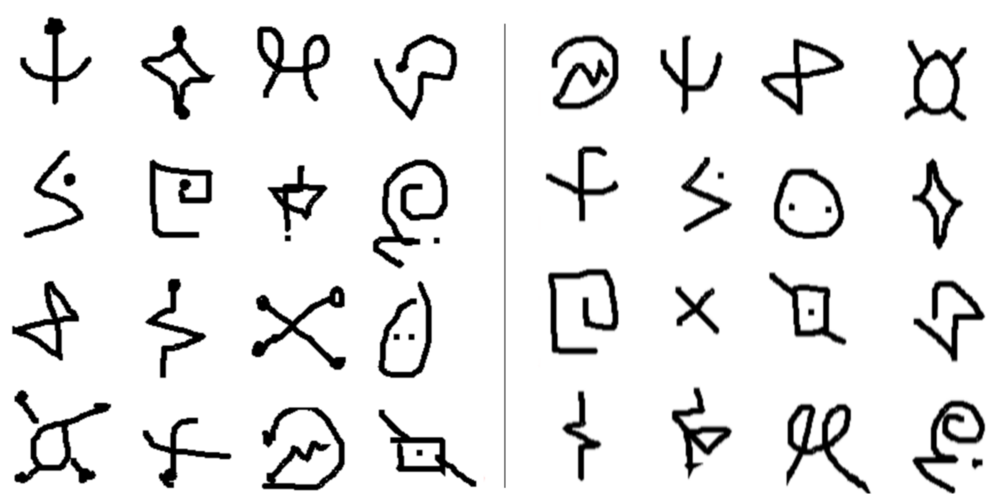
This is exactly what we do, with a simple set of challenges.

More Specifically

- The performance of different hypotheses is compared on a classification task. The learning curves are plotted as a function of training set size.
- Alternatively, alter the relationship between training and test set distributions; the task ranges from classification to transfer learning.

Experimental Protocol

Different challenges based on how the samples are placed in probe set P and target set S during testing.

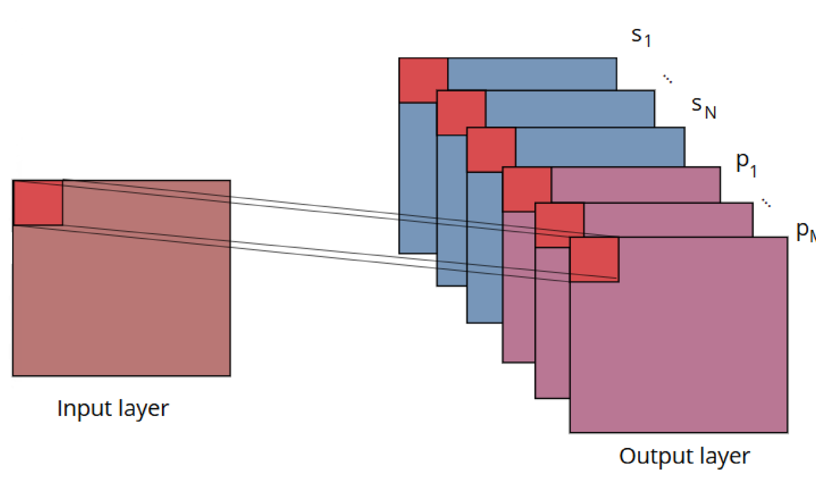


The algorithm sees a symbol on the left (probe set), and find the same character from the right (target set). Extract features from each image and do nearest-neighbor classification.

- **Challenge 0** P and S samples are from the training set.
- **Challenge 1** P and S samples are taken from new samples of characters that were trained on.
- **Challenge 2** P and S samples belong to completely unseen characters.

HiGSFA

Hierarchical **i**nformation-preserving **G**raph-based **S**low **F**eature **A**nalysis



- Hierarchical feature extraction, similar to Convolutional Neural Networks.
- Layers output N channels of slow features, M channels of PCA features.
- M is either fixed beforehand or determined via slowness threshold.

Learn features in a hierarchical manner by solving:

$$\begin{aligned}
\underset{y_j}{\text{minimize}}\ & \mathbb{E}\left[(y_j(x) - y_j(x'))^2\right] && \text{slowness} \\
\text{subject to}\ & \mathbb{E}\left[y_j(x)\right] = 0 && \text{zero mean} \\
& \mathbb{E}\left[y_j(x)^2\right] = 1 && \text{unit variance} \\
& \mathbb{E}\left[y_j(x) y_i(x)\right] = 0 && \text{decorrelation}
\end{aligned} \tag{1}$$

Preserve information by replacing too fast (noisy) features with PCA features.

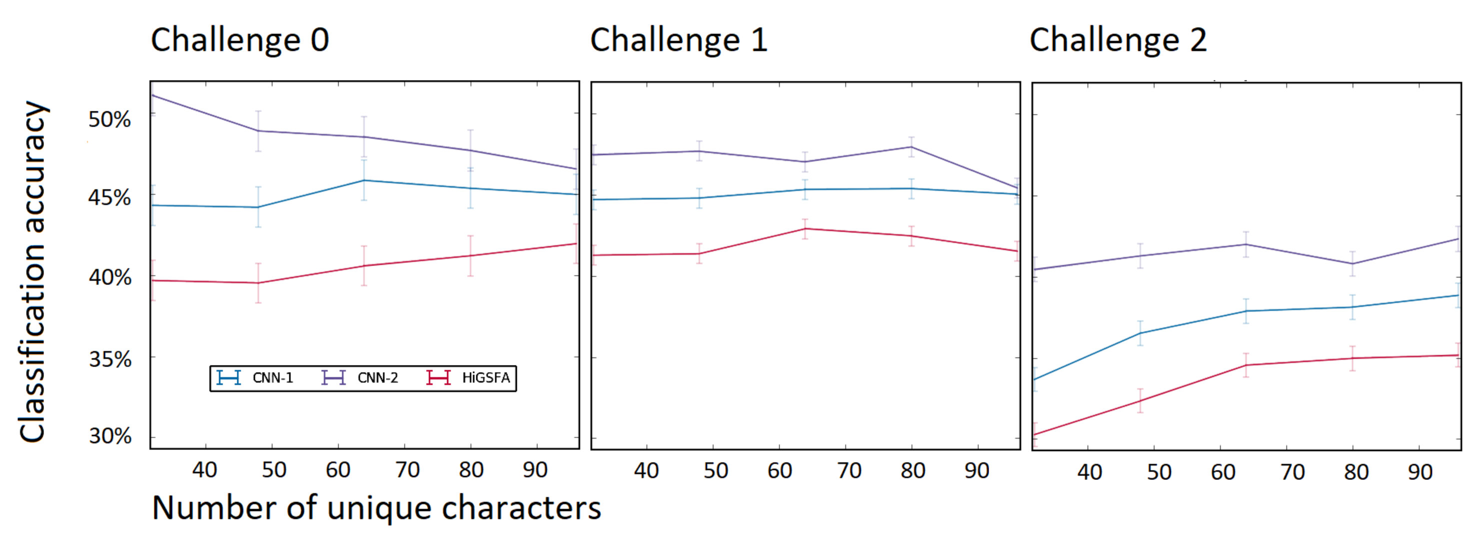
Results

Classification: MNIST, with a varying number of samples per digit.

Samples	HiGSFA		CNN-1		CNN-2	
	Acc.	Std.	Acc.	Std.	Acc.	Std.
5	35.68	± 0.43	**72.36**	± 0.37	72.32	± 0.09
10	75.74	± 0.22	**80.39**	± 0.24	79.55	± 0.18
50	**92.97**	± 0.05	90.32	± 0.10	91.46	± 0.07
200	**96.25**	± 0.03	94.67	± 0.06	95.65	± 0.05
500	97.19	± 0.01	96.58	± 0.05	**97.31**	± 0.05
2000	97.89	± 0.01	98.25	± 0.02	**98.57**	± 0.02
4000	98.13	± 0.01	98.69	± 0.01	**98.95**	± 0.02

MNIST. Average percentage of correctly classified samples on the test set from 100 runs.

Transfer learning: We fix either the number of alphabets, or characters-per-alphabet, to be 8 and vary the other number from 4 to 12.



Omniglot. The average of all the runs, with 16 training samples per character.

Future Work

- Invent more benchmarks for sample or data efficiency.
- Compare a wider variety of methods on increasingly heterogeneous data.
- Instead of comparisons: Define absolute measures of data efficiency.

References

[1] A. N. Escalante-B and L. Wiskott. Improved graph-based SFA: Information preservation complements the slowness principle. *CoRR*, 2016.

[2] B. M. Lake, R. Salakhutdinov, and J. B. Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338, 2015.

[3] S. Lawrence, C. L. Giles, and A. C. Tsoi. What size neural network gives optimal generalization? convergence properties of backpropagation. Technical report, 1998.

[4] M. Schüler, H. D. Hlynsson, and L. Wiskott. Gradient-based training of slow feature analysis by differentiable approximate whitening. *arXiv preprint arXiv:1808.08833*, 2018.