DEEP NETS DON'T LEARN VIA MEMORIZATION

David Krueger[1,⋆], **Nicolas Ballas**[1,⋆], **Stanislaw Jastrzebski**[2,⋆], **Devansh Arpit**[1,⋆],
Maxinder S. Kanwal[1], **Tegan Maharaj**[3], **Emmanuel Bengio**[4],
Asja Fischer[5], **Aaron Courville**[1]
[1] MILA, Université de Montréal, firstname.lastname@umontreal.ca.
[2] Jagiellonian University, staszek.jastrzebski@gmail.com.
[3] MILA, École Polytechnique de Montréal, tegan.maharaj@polymtl.ca.
[4] McGill University, emmanuel.bengio@mcgill.ca.
[5] University of Bonn, fischera@iai.uni-bonn.de.
⋆ Equal contributions.

ABSTRACT

We use empirical methods to argue that deep neural networks (DNNs) do not achieve their performance by *memorizing* training data, in spite of overly-expressive model architectures. Instead, they learn a simple available hypothesis that fits the finite data samples. In support of this view, we establish that there are qualitative differences when learning noise vs. natural datasets, showing that: (1) more capacity is needed to fit noise, (2) time to convergence is longer for random labels, but *shorter* for random inputs, and (3) DNNs trained on real data examples learn simpler functions than when trained with noise data, as measured by the sharpness of the loss function at convergence. Finally, we demonstrate that for appropriately tuned explicit regularization, e.g. dropout, we can degrade DNN training performance on noise datasets without compromising generalization on real data.

1 INTRODUCTION

Zhang et al. (2017) show that deep neural networks (DNNs) can be trained to fit random labels, and note that this poses a challenge for traditional theories of generalization based on measures of capacity. Based on their findings, they suggest that "brute-force memorization" may be part of an effective learning strategy for DNNs on real data, implying that generalization and memorization are not necessarily opposed. We argue this is not the case, and support our view by demonstrating qualitative differences in learning random noise vs. learning data.

What does it mean to 'memorize' a training set? In the context of learning, memorization means a failure to generalize. Zhang et al. (2017), however, claim that memorization may be a component of learning real tasks requiring generalization. This brings us to another, fuzzier definition of memorization: not learning patterns from data. Phrases like "brute-force memorization" (Zhang et al., 2017) connote fitting a dataset without capitalizing on any patterns in the data. In contrast, we believe that DNNs first learn and then refine simple patterns, which are shared across examples, in order to quickly drive down training loss, and only incorporate more case-by-case memorization as a later resort. Consider this popular mnemonic for remembering a English spellings: "i before e". This simple pattern is refined, as the mnemonic continues: "except after c", and further: "or when sounded as 'A' as in 'neighbor' and 'weigh"', with many further exceptions, such as "eigenvalue", existing and sometimes being incorporated into further rhymes.

2 EXPERIMENTS AND DISCUSSION

While random noise datasets of a fixed size do contain *some* patterns, we expect the patterns in most real-world deep learning datasets to be simpler and more general. To probe whether and how learning differs for real versus random data, we randomize the inputs or targets for some subset of data points, as in Zhang et al. (2017). Specifically, for some fraction of training examples, we

replace either the labels (with random labels), or the inputs (with i.i.d. Gaussian noise matching the real dataset's mean and variance).

2.1 LEARNING FROM DATA VS. NOISE

For these experiments, we train 2-layer ReLU-MLPs on MNIST for 1000 epochs using SGD with learning rate 0.01, using 5 random seeds for each hyper-parameter setting. Our first finding (see Figure 1), is that as more examples are replaced with noise, more capacity is needed in order to reach maximal validation performance. This indicates that networks are able to explain real data in terms of simpler patterns, which can be represented by fewer parameters.

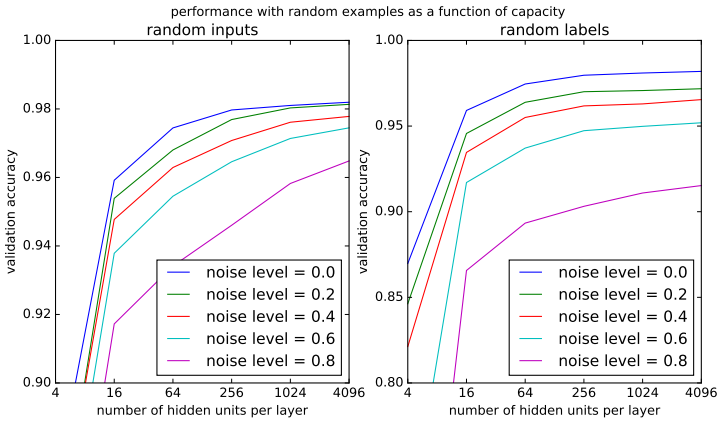

Figure 1: Performance as a function of capacity in 2-layer MLPs. For real data, performance is already very close to maximal with 4096 hidden units, but when there is noise in the dataset, higher capacity is needed.

Our next experiments measure time-to-convergence, i.e. how many epochs it takes to reach 100% training accuracy. Reducing the capacity or increasing the size of the dataset slows down training for real data as well as noise. However, the effect is more severe for datasets containing noise, as our experiments (Figure 2) show.

The reduced dependence of training time on dataset size demonstrates that the functions learned by a neural net capture patterns in the data. Since these patterns are consistent across data examples, adding more real data examples requires less changes to the parameter values. In contrast, adding more noise examples continues to require large changes to the parameters, because there are no consistent patterns in the noise data.

The reduced dependence of training time on capacity suggests that hypotheses learned by DNNs trained on real data are simpler than those learned on noise data, reinforcing the results in Figure 1.

2.2 COMPLEXITY OF THE LEARNED FUNCTION

Hochreiter & Schmidhuber (1997) argue that flat minima of the loss function have better generalization properties, using a Bayesian MDL-based argument. Keskar et al. (2017) argue that SGD learns flatter minima when the batch-size is smaller, explaining the experimental finding that small batch-sizes yield better generalization, and correlating this with a novel measure of flatness. In Figure 3 (left), we investigate the sharpness/flatness of learned function around the local minima given different amount of noise in the dataset. To characterize the sharpness, we approximate the norm of the Hessian's largest eigenvalue using the method of (LeCun et al., 1993). As we increase the amount of noise in the training data, we observe an increase in this sharpness around the local minima of the learned function, indicating that fitting noise results in a more complex function.

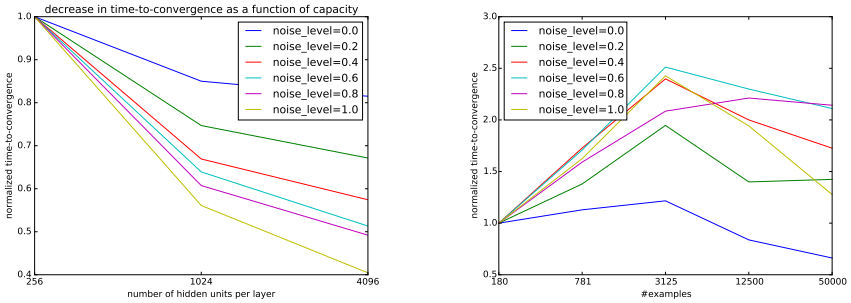

Figure 2: Change in time to convergence: as a function of capacity with dataset size fixed to 50000 (left), or dataset size with capacity fixed at 4096 units (right). Noise level is the proportion of the dataset whose inputs are replaced with Gaussian noise. Because there is are patterns underlying real data, having more capacity/data doesn't help/hurt performance as much as it does for noise data.

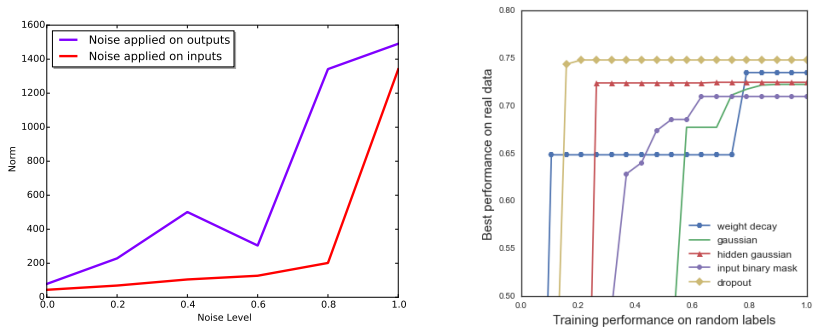

Figure 3: Left: Approximate norm of the Hessian's largest eigenvalue for MLPs trained on the MNIST dataset with different levels of input and label noise. Right: Effect of regularization on train accuracy for noise data vs. generalization on real data.

2.3 Effect of regularization on learning

We now assess the ability of regularization to degrade training performance on random labels, while maintaining generalization performance on real data. Specifically, we train a small Alexnet-style CNN (following Zhang et al. (2017)) on CIFAR-10 with either real or random labels. We compare the following regularizers: dropout(0-.9), input dropout(0-.9), Gaussian noise(0-5), and weight decay (0-1), and find that dropout is most able to hinder memorization without reducing the model's ability to learn. We train for 50 epochs and a learning rate schedule of 0.01, dropping by half every 10 epochs.

Results are shown in Figure 3 (right). Each curve represents a different regularizer. We plot the (random label) train accuracy achieved with a particular value of the regularizer on the x-axis, and the corresponding value of (real label) validation accuracy on the y-axis. Higher curves indicate better performance overall (on real data). Flat curves further indicate that the corresponding regularization technique differentially targets memorization-style learning (of random labels).

3 Conclusion

Our empirical exploration demonstrates qualitative differences in DNN training for noise vs. real data, all of which support the claim that DNNs use simple hypotheses, not memorization, to fit real data. Nonetheless, since DNNs have the effective capacity to fit noise, it is unclear why they recover generalizable solutions on real data. We hypothesize this is because finding patterns is *easier* than brute-force memorization, and hope that this can be formalized in future work.

ACKNOWLEDGMENTS

We thank Akram Erraqabi, Simon Lacoste-Julien, and Jason Jo for helpful discussions. Third author was supported by Grant No. DI 2014/016644 from Ministry of Science and Higher Education, Poland.

REFERENCES

Sepp Hochreiter and Jürgen Schmidhuber. Flat minima. *Neural Computation.*, pp. 1–42, 1997.

Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On large-batch training for deep learning: Generalization gap and sharp minima. *International Conference on Learning Representations (ICLR)*, 2017.

Y. LeCun, P. Simard, and B. Pearlmutter. Automatic learning rate maximization by on-line estimation of the hessian's eigenvectors. In S. Hanson, J. Cowan, and L. Giles (eds.), *Advances in Neural Information Processing Systems (NIPS 1992)*, volume 5. Morgan Kaufmann Publishers, San Mateo, CA, 1993.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. *International Conference on Learning Representations (ICLR)*, 2017.