A Closer Look at Memorization in Deep Networks

Presenter: Ceyer Wakilpoor

Devansh Arpit[1,2] Stanislaw Jastrzebski[3] Nicolas Ballas[1,2] David Krueger[1,2] Emmanuel Bengio[4] Maxinder S. Kanwal [5] Tegan Maharaj [1,6] Asja Fischer [7] Aaron Courville [1,2,8] Yoshua Bengio [1,2,9] Simon Lacoste-Julien [1,2]

[1]Montral Institute, [2]Universit de Montral

[3]Jagiellonian University, [4]McGill University

[5]University of California, [6]Polytechnique Montral

[7]University of Bonn, [8]CIFAR Fellow

[9]CIFAR Senior Fellow

ICML, 2017

Outline

Outline

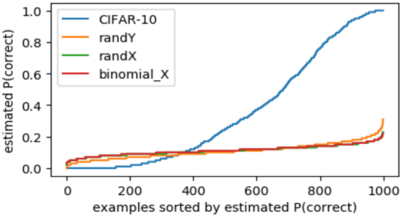
Measuring Memorization

- Old notions of generalization state that models with sufficient capacity can "memorize" a data-set
- DNNs have been observed to break these notions despite high *representational capacity*
- Important to consider training method, *Effective Capacity:* $EC(A) = \{\, h \,|\, \exists D \ such\, that\ \ h \,\epsilon\, A(D) \,\}$
- DNNs can still fit random noise
- Define "memorization" as differences in behavior of DNNs when trained on noise vs real data

Outline

Main Observations

- Noise data has little to no differences between labels, indicating the examples are fit more independently, rendering the labels more difficult to learn
- When training a DNN on random data and real data, DNNs proved to learn "easy" patterns before "difficult" ones



Outline

Loss-Sensitivity

- Simulate measuring "memorization" by measuring how much the effect of each sample has on the average loss

$$g_x^t = ||\partial L_t / \partial x||_1 \qquad \bar{g}_x = \frac{\sum_{t \epsilon T} g_x^t}{|T|}$$

- In real data this value was high for a subset of the data, in random data it was high for all of the data

Loss-Sensitivity

- Using the gini coefficient to measure the g_x^t shows that in random data, the network learns all of the examples equally as needed in rote memorization

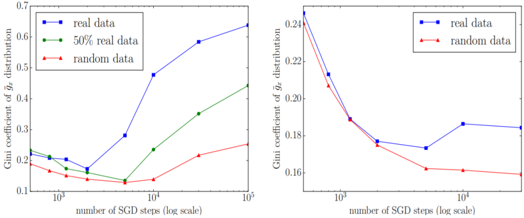


Figure 3. Plots of the Gini coefficient of $\bar{g}_{\mathbf{x}}$ over examples $\mathbf{x}$ (see section 3.2) as training progresses, for a 1000-example real dataset (14x14 MNIST) versus random data. On the left, Y is the normal class label; on the right, there are as many classes as examples, the network has to learn to map each example to a unique class.

Loss-Sensitivity

- Class specific loss sensitivity, where $L_{t(y=i)}$ is the cross-entropy sum corresponding to class i: $\bar{g}_{i,j} = \mathbb{E}_{(x,y)} \frac{\sum_{t \epsilon T} |\partial L_t(y=i)/\partial x_{y=j}|}{|T|}$
- Note how concentrated the random data is for i = j:

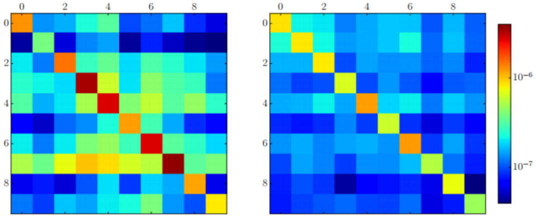


Figure 4. Plots of per-class g_x (see previous figure; log scale), a cell i, j represents the average $|\partial \mathcal{L}(y = i)/\partial x_{y=j}|$, i.e. the loss-sensitivity of examples of class i w.r.t. training examples of class j. Left is real data, right is random data.

Outline

Capacity and Effective Capacity

- On MNIST, validation accuracy improved with higher capacity when noise examples were present
- However, no differences were found on CIFAR10, which is contrary to traditional thoughts on limiting capacity being able to help generalization
- This suggests the DNN would have sufficient capacity regardless

Capacity and Effective Capacity

- When increasing training examples or decreasing capacity, training on each dataset slowed down, but especially so on those containing noise
- Effective capacity of a DNN as defined before, can increased by adding neurons or by training longer
- Increasing effective capacity gave larger diminishing returns with real data compared to data with noise
- On noise data, time-to-convegence is longer and increases substantially as a function of dataset size compared to real data
- All of these further support that DNNs will learn patterns before memorizing

Critical Sample Ratio

- Critical sample is a subset of data where there is an adversarial example in its proximity:

$$\arg\max_i f_i(x) \neq \arg\max_j f_j(\hat{x})$$
$$s.t. \, ||x - \hat{x}||_\infty \leq r$$

- A high number of critical samples would be indicative of a complex hypothesis
- The critical sample ratio would be the $\frac{\#critical\ samples}{\#data\ points}$

Critical Sample Ratio

- Use LASS (Langevin Adversarial Sample Search) to determine critical sample
- LASS uses the gradient of the networks output vector and adds noise to avoid getting stuck at training points where the gradient is zero

Algorithm 1 Langevin Adversarial Sample Search (LASS)

Require: $\mathbf{x} \in \mathbb{R}^n$, α, β, r, noise process η

Ensure: $\hat{\mathbf{x}}$

converged = FALSE
$\tilde{\mathbf{x}} \leftarrow \mathbf{x}$; $\hat{\mathbf{x}} \leftarrow \emptyset$
while not converged or max iter reached **do**
    $\Delta = \alpha \cdot \operatorname{sign}(\frac{\partial f_k(\mathbf{x})}{\partial \mathbf{x}}) + \beta \cdot \eta$
    $\tilde{\mathbf{x}} \leftarrow \tilde{\mathbf{x}} + \Delta$
    for $i \in [n]$ **do**
        $\tilde{\mathbf{x}}_i \leftarrow \begin{cases} \mathbf{x}_i + r \cdot \operatorname{sign}(\tilde{\mathbf{x}}_i - \mathbf{x}^i) & if |\tilde{\mathbf{x}}_i - \mathbf{x}_i| > r \\ \tilde{\mathbf{x}}_i & otherwise \end{cases}$
    end for
    if $\arg\max_i f(\mathbf{x}) \neq \arg\max_i f(\tilde{\mathbf{x}})$ **then**
        converged = TRUE
        $\hat{\mathbf{x}} \leftarrow \tilde{\mathbf{x}}$
    end if
end while

Critical Sample Ratio

- The number of critical samples is much higher when a deep CNN is trained on noise data, results recorded on validation set through training (they used an r of .3)

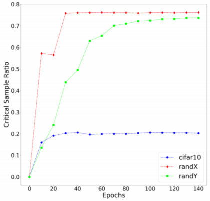


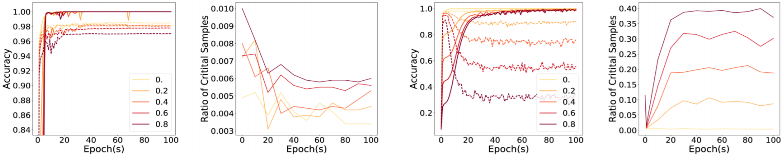
Figure 9. Critical sample ratio throughout training on CIFAR-10, random input (randX), and random label (randY) datasets.

- The higher number of CSRs on the noise data suggest a more complex learned decision surface
- The gradual increase and then plateau of the CSR suggests complex hypotheses are learned in later epochs

Critical Sample Ratio

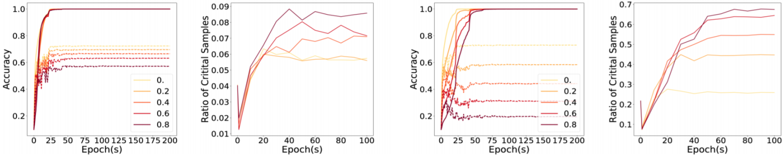
- A similar test was run with 20-80% of the training dataset replaced with either input or labeled noise
- The accuracy goes lower when in later epochs when the noise is higher
- Indicated how the network fits more complex non-target concepts

Critical Sample Ratio



(a) Noise added on classification inputs. (b) Noise added on classification labels.

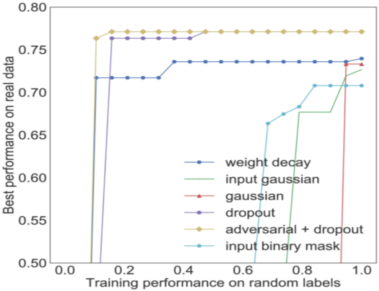
Figure 7. Accuracy (left in each pair, solid is train, dotted is validation) and Critical sample ratios (right in each pair) for MNIST.



(a) Noise added on classification inputs. (b) Noise added on classification labels.

Regularization

- Previous studies show that SGD has a bigger role in generalizing well compared to explicit regularization
- Flat curve would indicate good performance, where the validation accuracy increases with training accuracy, so adversarial training + dropout avoided memorization the best



Summary

- There are qualitative differences in DNN optimization behavior on real data vs. noise. In other words, DNNs do not just memorize real data.
- DNNs learn simple patterns first, before memorizing. In other words, DNN optimization is content-aware, taking advantage of patterns shared by multiple training examples.
- Regularization techniques can deferentially hinder memorization in DNNs while preserving their ability to learn about real data.

Future Work

Using data-dependent research, understand why DNNs have still been able to find generalizable solutions to real data