ADVERSARIAL ROBUSTNESS THROUGH OVERPARAMETERIZATION

AJ BARRY

1. Introduction

In classical statistics, one chooses their model to minimize some Risk function (mean squared error, etc.) or some approximation thereof. If their model is parametrized by p parameters, this means choosing some optimal number of parameters p^*. In such a regime one must be careful not to under fit the data (high model bias, low variance) or over fit the model (high variance, low bias). However, in modern machine learning there has been a recent explosion in the size of models. Many cutting edge models have hundreds of billions of parameters, much more than even the number of training examples. As shown in [1], if one has n data points and n parameters, one can perfectly fit their model to the data. These extremely overparameterized models appear to have yielded much better results than the smaller models. There are several hypothesies as to why these larger models seem to work better than predicted. One is the "Benign overfitting" hypothesis, which states that if both the number of data points, and the dimension of the data are large, then some over fitting is not problematic. However, Bubeck and Sellke in [5] provide an alternative explanation, that one in fact needs to have extreme overparameterization to have a robust Neural Network.

Theorem 1.1. *[5] Let $\mathcal{F}$ be a class of functions $\mathbb{R}^d \to \mathbb{R}$ and let $(x_i, y_i)_{i\in[n]}$ be i.i.d. input output pairs in $\mathbb{R}^n \times [-1, 1]$. If:*

1. *admits a Lipschitz parametrization by p real parameters, each of size at most poly(n, d).*
2. *The distribution μ of the covariates x_i satisfies Isoperimetry (or some mixture thereof).*
3. *The expected conditional variance of the output is strictly positive. Denoted $\sigma^2 \equiv_\mu [Var(y|x)] > 0$.*

Then w.h.p. over the sampling data, for all $f \in$:

$$\frac{1}{n}\sum_{i=1}^{n}(f(x_i) - y_i)^2 \leq \sigma^2 - \epsilon \Rightarrow Lip(f) \geq \tilde{\Omega}\left(\frac{\epsilon}{\sigma}\sqrt{\frac{nd}{p}}\right)$$

Date: 2023-4-22.

To interpret the theorem, 1.1 shows that if our data satisfies the quite general property of isoperimetry, and the output function has polynomial bounded weights, then one needs at least $p = nd$ parameters for there to be an output function which perfectly memorizes our data which also has a small Lipshitz constant. Since adversarial attacks generally are small perturbations of our training data, having a Lipshitz function with small constant guarantees that our adversarial output will not vary too much from that of our uncorrupted output. What we aim to do in this paper, is to provide some empirical experiments which lend creedence to Bubeck & Sellke's results. In our first experiment we will attempt to replicate the results of Experiment 2 from [4], and for our next experiment we will train two neural networks on the MNIST dataset, and test their robustness to adversarial attacks and perturbations.

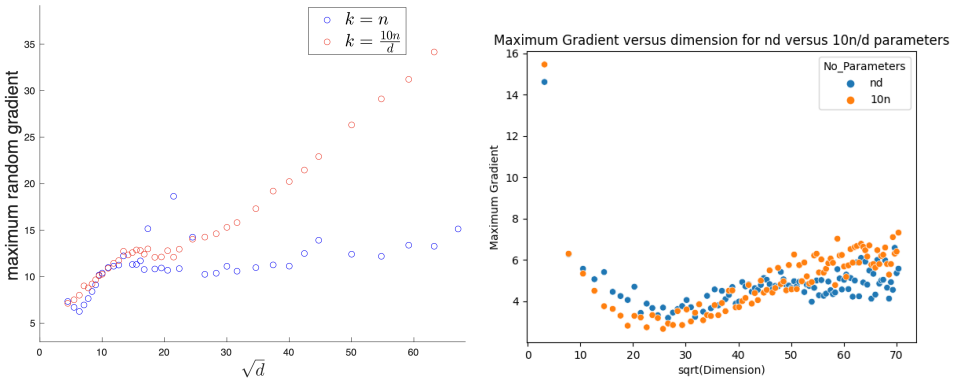
2. Experiment on Random Data

In this section I attempted to replicate the an experiment performed by Bubeck et.al. in [4][Experiment t2]. This experiment aims to provide evidence that if f is a perfect memorizer with $p = nd$ parameters, then f will have an approximately constant maximum gradient with respect to the dimension. Whereas for $p = n$ parameters, the maximum gradient will increase linearly with the square root of dimension.

For this experiment we consider a generic Gaussian data set generated $x_i \overset{iid}{\sim} N(0, \frac{1}{d} I_d)$ and uniformly generated from $y_i \overset{iid}{\sim} U(\{-1, +1\})$. We will train two-layer neural networks with ReLU activations, and using the Adam optimizer on the least squares loss. For the experiment in question, we fix $n = 10^4$ and vary d from 10-5000 in increments of 50. We will use $p = nd$ parameters for the larger model and $p = 10n$ parameters for the smaller model. We train both models until the loss function is at most $\epsilon = 0.1$.

To estimate the Lipschitz constant numerically we instead will compute the maximum random gradient of the output function. We will do this by generating 1000 random samples $z_j \overset{iid}{\sim} N(0, \frac{1}{d} I_d)$ and computing the maximum gradient among those samples $\max_{z_j} \|\nabla f(z_j)\|$ for each model and for each value of d. The only difference between our model and that of [4] is that their results employed batch normalization, while ours did not. When I noticed the error in my code, I attempted to re-run the models. However, the code appeared to take much longer to run, to the point that computing the figure with batch normalization was untenable given the hardware and time at my disposal. Unfortunately the code from [4] was not publicly available so I could not verify my results with respect to theirs. We contrast our results to those found in [4] below.

.



(A) Results from [4] (B) Our experimental results

The results found in this paper were not as clean as those found in [4]. While the models with nd parameters appear to have a lower maximum gradient (on average) for large values of d, and appear to be roughly constant after a certain point, our results appear to show a different overall pattern than those found by [4]. This could be due to our lack of batch normalization, however I was unable to verify due to the aforementioned issues.

3. MNIST Experiment

In [5] they prove that with high probability it is *necessary* for a neural network to have at least nd parameters for it to be robust to adversarial attacks (Lipschitz). However, they make no claims as to the sufficiency of this criterion. To this end, we aim to examine if simply having a (near) zero loss memorizer with greater than nd parameters provides us with robust performance relative to a smaller model ($p = 2n$) of the same architecture.

For this experiment we will train both models on the MNIST data set. The MNIST data set is somewhat of a canonical example in the machine learning literature, and consists of 70,000 images of handwritten digits (60,000 training, 10,000 test). Each image is a 28×28 pixels and we will normalize the values of the pixels so that they are in $[0, 1]$.

We will train two models. One with $p = 3 * 10^6$ parameters, and the other with $p = 12 * 10^4$ parameters. While MNIST has dimension $28^2 = 784$, it is estimated that it has effective dimension on the order of $[5 - 20]$. Therefore, our model with $3 * 10^6$ will approximate the case of $p = nd$ and the model with $12 * 10^4$ parameters will approximate the case of $p = n$. As such, we will refer to the model with $p = 3 * 10^6$ parameters as f_{nd} and the model with $p = 12 * 10^4$ parameters as f_n. Both models will be simple two layer neural networks with ReLU activation in the hidden layer and softmax activation for the output layer. The two models will be trained using the Adam optimizer and the categorical cross-entropy loss function. Both models were trained

until the loss was below a threshold of $\epsilon = 10^{-10}$. We can see the results of our model on the test data in 2.

Model	Test Accuracy	Test Loss
f_{nd}	0.985	0.587
f_n	0.980	0.211

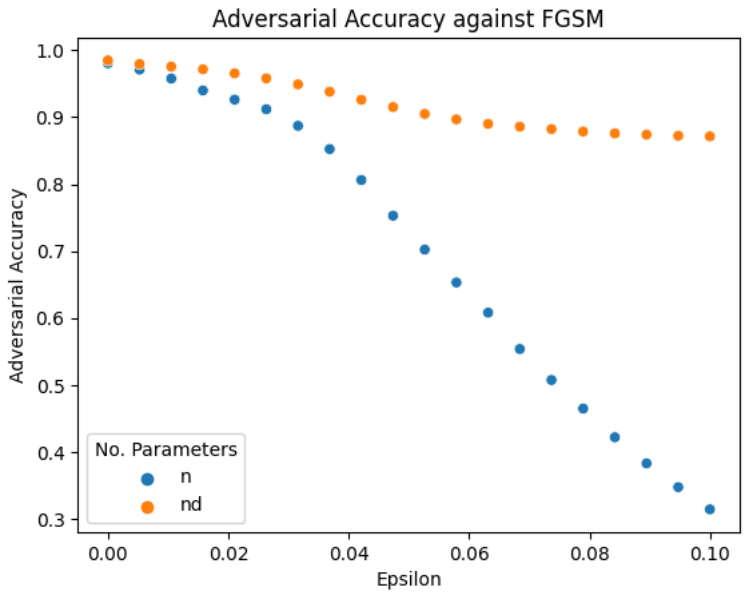
FIGURE 2. Test results of model f_{nd} and f_n

We notice that the model with more parameters has slightly better performance on the test set, however the difference is not particularly significant. We now move towards testing both models against adversarial attacks and corrupted data. Our first example of an adversarial attack will be the Fast Gradient Sign method.

Definition 3.1 (Fast Gradient Sign Method)**.** [6] For x being an input to a NN, y being the target associated with x, θ being the parameters of the model and L being the cost function used to train the Neural Network. The *Fast Gradient Sign Method* attack is:

$$x_{\text{adv}} = x + \epsilon \operatorname{sgn}(\nabla_x L(x, \theta, y))$$

Evaluating the FGSM adversaries for both models on every image in our test set yields the results found below.



We notice that even for lager values of ϵ, the larger model still maintains robust performance against FGSM. Even for larger values of ϵ such as $\epsilon =$

0.1, the model still classifies nearly 90% of the test images correctly. We can contrast this to the second model, which suffers substantially from the adversarial attack, reaching approximately 30% accuracy for $\epsilon = 0.1$. We can note some examples of the adversarial attacks generated for the two models below.

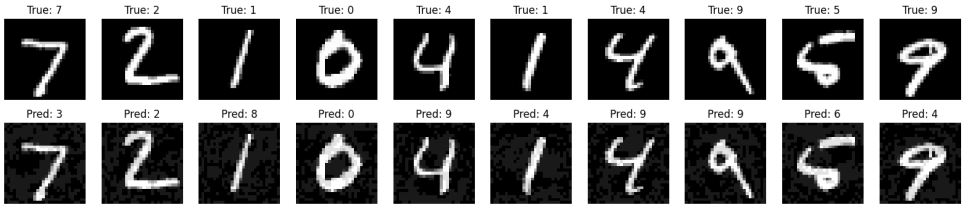


FIGURE 3. FGSM attacks for f_n

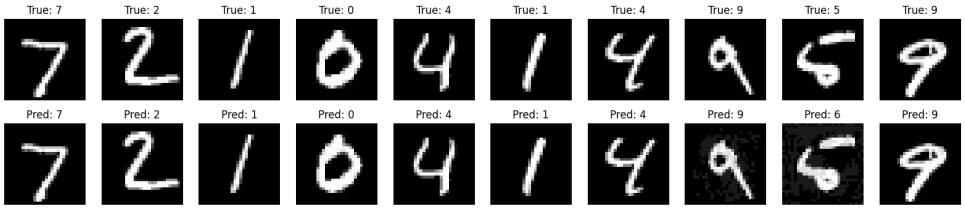


FIGURE 4. FGSM attacks for f_{nd}

From visual inspection, it seems clear that the images for the smaller model appear to be suffering more extreme corruptions from FGSM than those from the larger model. After some investigation, it appears that this is occurring because 86.3% images for the larger model were evaluated to have loss gradient of precisely zero (or at least sufficiently small that they were numerically evaluated to be zero). This indicates that FGSM or any other adversarial attack which uses the gradient of the loss function may not provide a fair comparison among the models. However, this peculiar result is notable and could say something about potential benefits of overparameterization.

We now move towards evaluating both models against a white noise attack. We will define a white noise attack as follows.

Definition 3.2 (White Noise Attack)**.** Let $\eta \sim N(0, \epsilon^2 I_d)$ for d being the dimension of x. The *White Noise Attack* is:

$$x_{adv} = x + \eta.$$

The results of both models against white noise attacks for every item in the test set can be found in Figure 5.

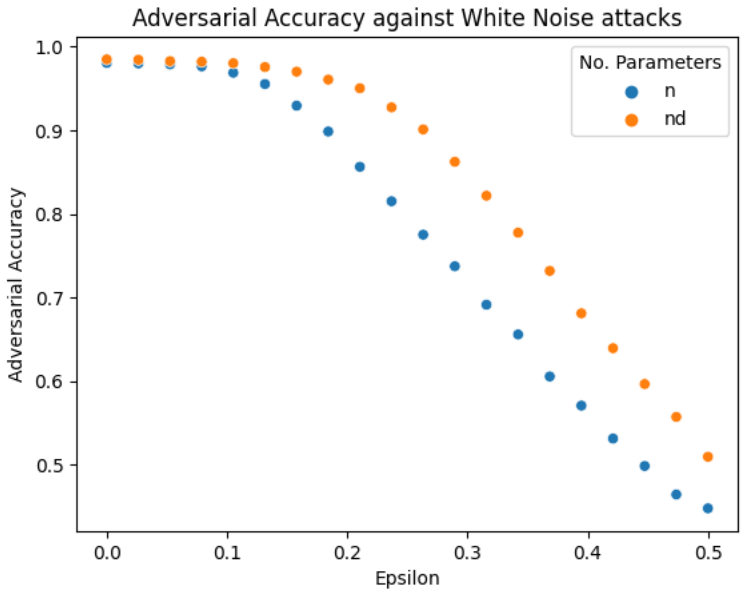


FIGURE 5

We can see from 5 that the overparametrized model continuously outperforms the smaller model in its robustness towards white noise attacks, oftentimes by over 10%.

The final test which we will conduct is to test our two models on the examples from the `mnist_corrupted` data set on tensorflow. the `mnist_corrupted` dataset consists of 15 different corruptions that are applied to the test images. We can see the performance of our respective models against the test sets for each respective corruption in Figure 6.

Corruption	Accuracy f_{nd}	Accuracy f_n
identity	0.9848	0.9805
shot_noise	0.9767	0.9711
impulse_noise	0.8987	0.7474
glass_blur	0.9417	0.9481
motion_blur	0.847	0.8388
shear	0.9533	0.9288
scale	0.7509	0.6788
rotate	0.8742	0.8511
brightness	0.1503	0.2295
translate	0.3614	0.3029
stripe	0.3695	0.1591
fog	0.128	0.1807
spatter	0.9613	0.9261
dotted_line	0.9575	0.8893
zigzag	0.8186	0.6801
canny_edges	0.6744	0.6533

FIGURE 6. Results for each data set in `minst_corrupted`

Once again we can see that the larger model generally outperforms the smaller one. While there are some instances where the smaller model appears to outperform, these cases have either minuscule performance differences (glass_blur) or both models are exceptionally poor classifiers (fog, brightness).

In sum, the larger model does seem to perform substantially better against adversarial attacks and on corrupted data, lending credence to Bubeck and Sellke's findings.

REFERENCES

[1] Eric B Baum. "On the capabilities of multilayer perceptrons". en. In: *Journal of Complexity* 4.3 (Sept. 1988), pp. 193–215. DOI: 10.1016/0885-064X(88)90020-9.

[2] Mikhail Belkin, Siyuan Ma, and Soumik Mandal. *To understand deep learning we need to understand kernel learning*. Tech. rep. arXiv:1802.01396. arXiv:1802.01396 [cs, stat] type: article. arXiv, June 2018.

[3] Mikhail Belkin et al. "Reconciling modern machine-learning practice and the classical bias–variance trade-off". en. In: *Proceedings of the National Academy of Sciences* 116.32 (Aug. 2019), pp. 15849–15854. DOI: 10.1073/pnas.1903070116.

[4] Sébastien Bubeck, Yuanzhi Li, and Dheeraj Nagaraj. *A law of robustness for two-layers neural networks*. Tech. rep. arXiv:2009.14444. arXiv:2009.14444 [cs, stat] type: article. arXiv, Nov. 2020.

[5] Sébastien Bubeck and Mark Sellke. *A Universal Law of Robustness via Isoperimetry*. Tech. rep. arXiv:2105.12806. arXiv:2105.12806 [cs, stat] type: article. arXiv, Dec. 2022.

[6] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. *Explaining and Harnessing Adversarial Examples*. Tech. rep. arXiv:1412.6572. arXiv:1412.6572 [cs, stat] type: article. arXiv, Mar. 2015.

[7] Yann LeCun, Corinna Cortes, and CJ Burges. "MNIST handwritten digit database". In: *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist* 2 (2010).

[8] Song Mei and Andrea Montanari. *The generalization error of random features regression: Precise asymptotics and double descent curve*. Tech. rep. arXiv:1908.05355. arXiv:1908.05355 [math, stat] type: article. arXiv, Dec. 2020.

[9] Norman Mu and Justin Gilmer. "MNIST-C: A Robustness Benchmark for Computer Vision". In: *arXiv preprint arXiv:1906.02337* (2019).

[10] Preetum Nakkiran et al. *Deep Double Descent: Where Bigger Models and More Data Hurt*. Tech. rep. arXiv:1912.02292. arXiv:1912.02292 [cs, stat] type: article. arXiv, Dec. 2019.

[11] Christian Szegedy et al. *Intriguing properties of neural networks*. Tech. rep. arXiv:1312.6199. arXiv:1312.6199 [cs] type: article. arXiv, Feb. 2014.