Understanding deep learning requires rethinking generalization

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, Oriol Vinyals

Presentation by Jason Hartford

This paper

see: https://openreview.net/forum?id=Sy8gdB9xx

The authors report the experimental findings of a fascinating inquiry on the ability of the deep neural networks to fit randomly labelled data. The investigation is sound, enlightening, and inspiring…

This is definitely groundbreaking work, which will inspire many works in the coming years.

- ***ICLR meta review (scores: 10, 9, 10)***
  Got ICLR best paper 2017

This paper

see: https://openreview.net/forum?id=Sy8gdB9xx

The authors report the experimental findings of a fascinating inquiry on the ability of the deep neural networks to fit randomly labelled data. The investigation is sound, enlightening, and inspiring…

This is definitely groundbreaking work, which will inspire many works in the coming years.
- *ICLR meta review (scores: 10, 9, 10)*
Got ICLR best paper 2017

I expected to like this paper, because I respect the authors, and many people have said good things about it.... I'm sorry to say I was very disappointed.

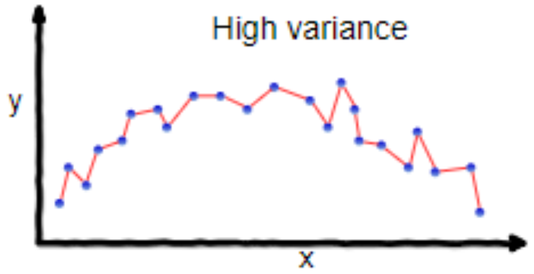
[T]he results in this paper are completely unsurprising. I'm surprised that the authors were surprised. I'm shocked that at least one reviewer thought this was ground breaking.
- Thomas G Dietterich

This paper

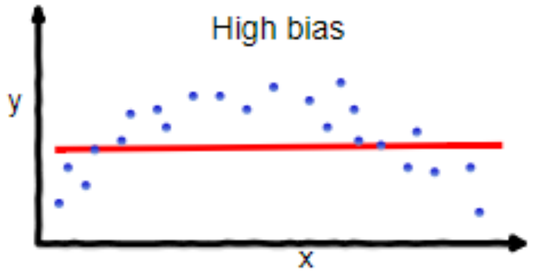
see: https://openreview.net/forum?id=Sy8gdB9xx

Bias - variance tradeoff

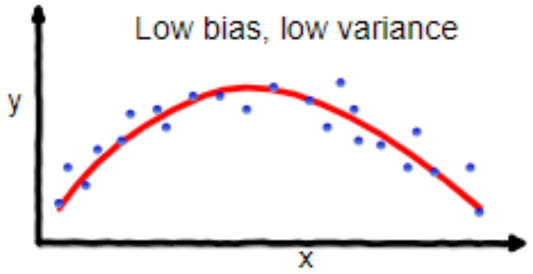
If you Google “bias variance tradeoff”, this is the first result…



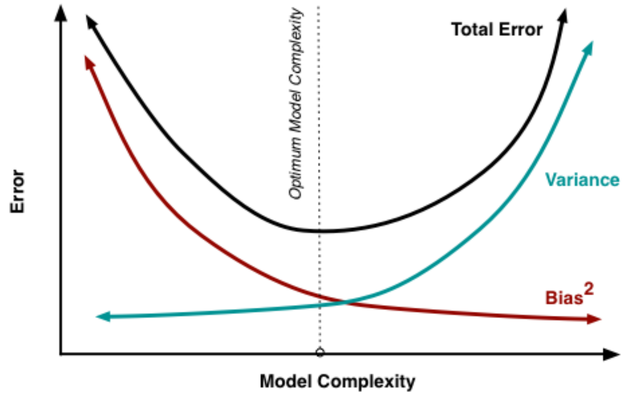
overfitting



underfitting

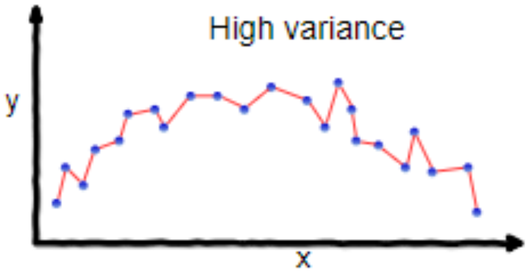


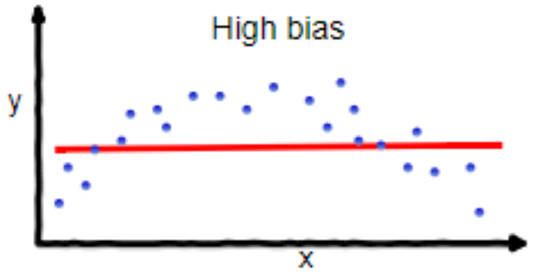
Good balance

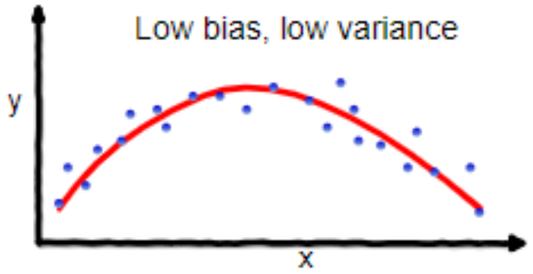


Bias - variance tradeoff

If you Google “bias variance tradeoff”, this is the first result…



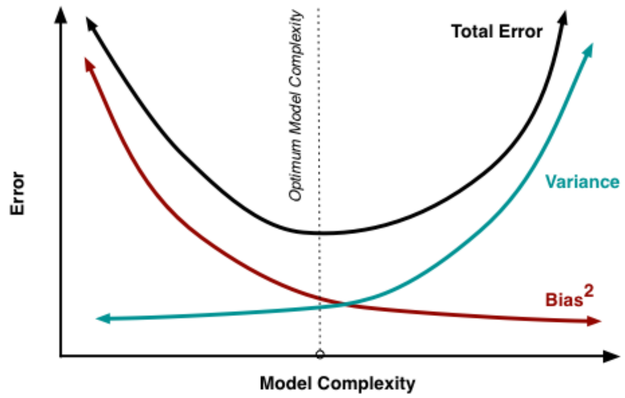




“State of the art deep nets do this…”

underfitting

Good balance



Key contributions

Empirically show that standard theory that explains generalization in IID settings can't distinguish between neural networks that have radically different generalization performance. They show…

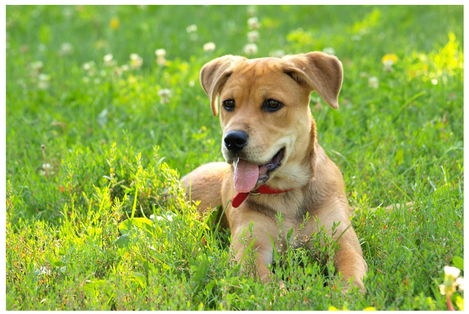
- **Deep networks can easily fit random labels** *(for image data)*

- Explicit regularization helps generalization, but is not sufficient to explain generalization performance (under standard generalization bounds).

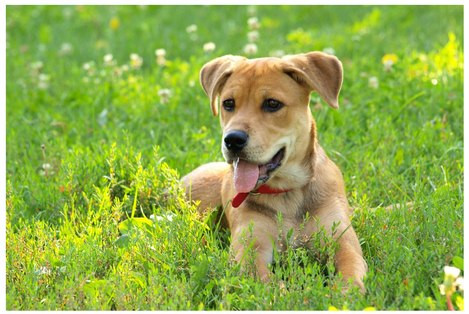
- Explicit construction showing a 2 layer ReLU network with 2n+d parameters can perfectly fit an n x d training matrix.

- SGD as an implicit regularizer in linear models

Randomization tests

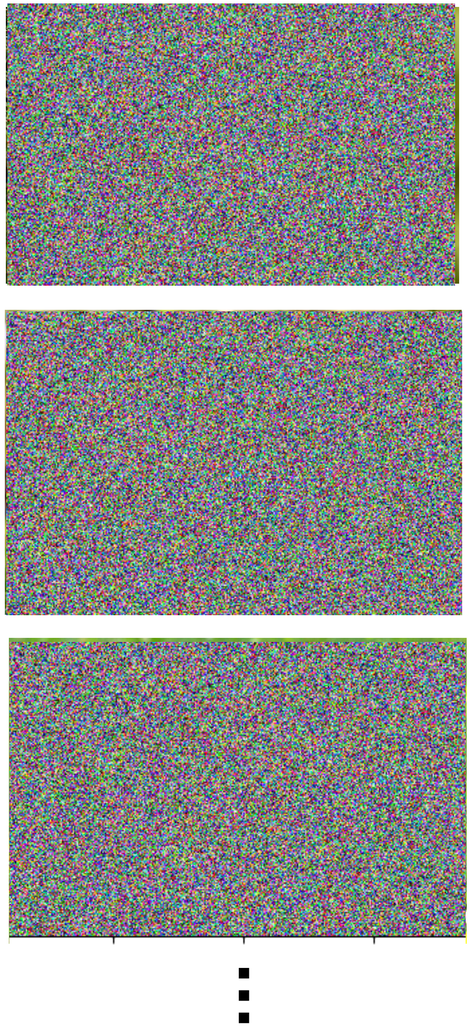
Input image	True label
	“cat”
	“dog”
	“dog”
⋮	

Randomization tests

Input image	True label	Random label
	“cat”	“airplane”
	“dog”	“automobile”
	“dog”	“cat”
⋮		

Randomization tests

Input image



True label

“cat”

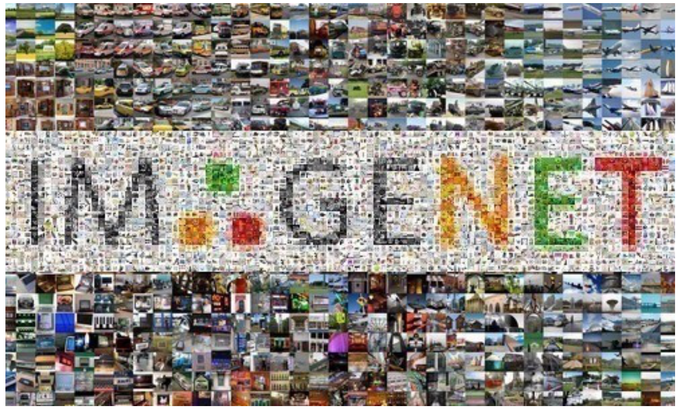
“dog”

“dog”

Also tested permuting pixels with a fixed and random permutation matrix, but didn’t include full results on those….

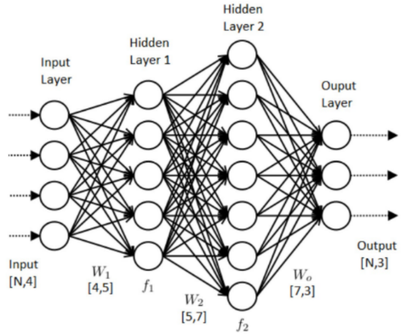
Datasets and models

CIFAR 10
n = 60,000 images
32 x 32 with 10 classes

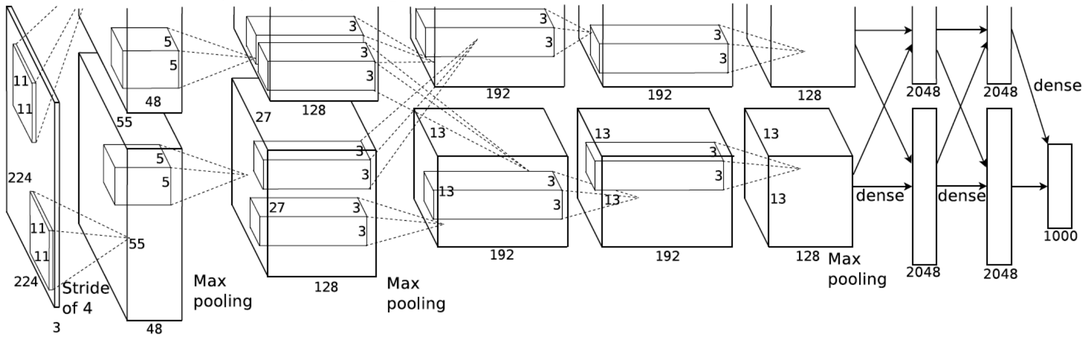


IMAGENET
n = 1,281,167 images
299 x 299 with 1000 classes

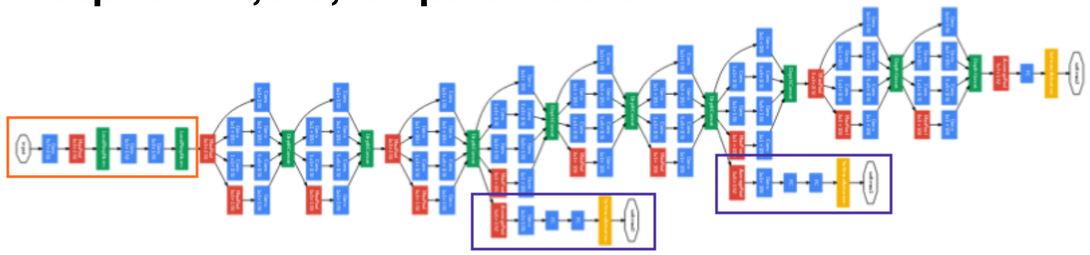
Multilayer perceptron
1,735,178 parameters $\approx 28n$



AlexNet - 1,387,786 parameters $\approx 23n$



Inception - 1,649,402 parameters $\approx 27n$



Trained with SGD + momentum with the same learning rates on both true & random labels. Explicit regularization turned off initially…

What do we expect?

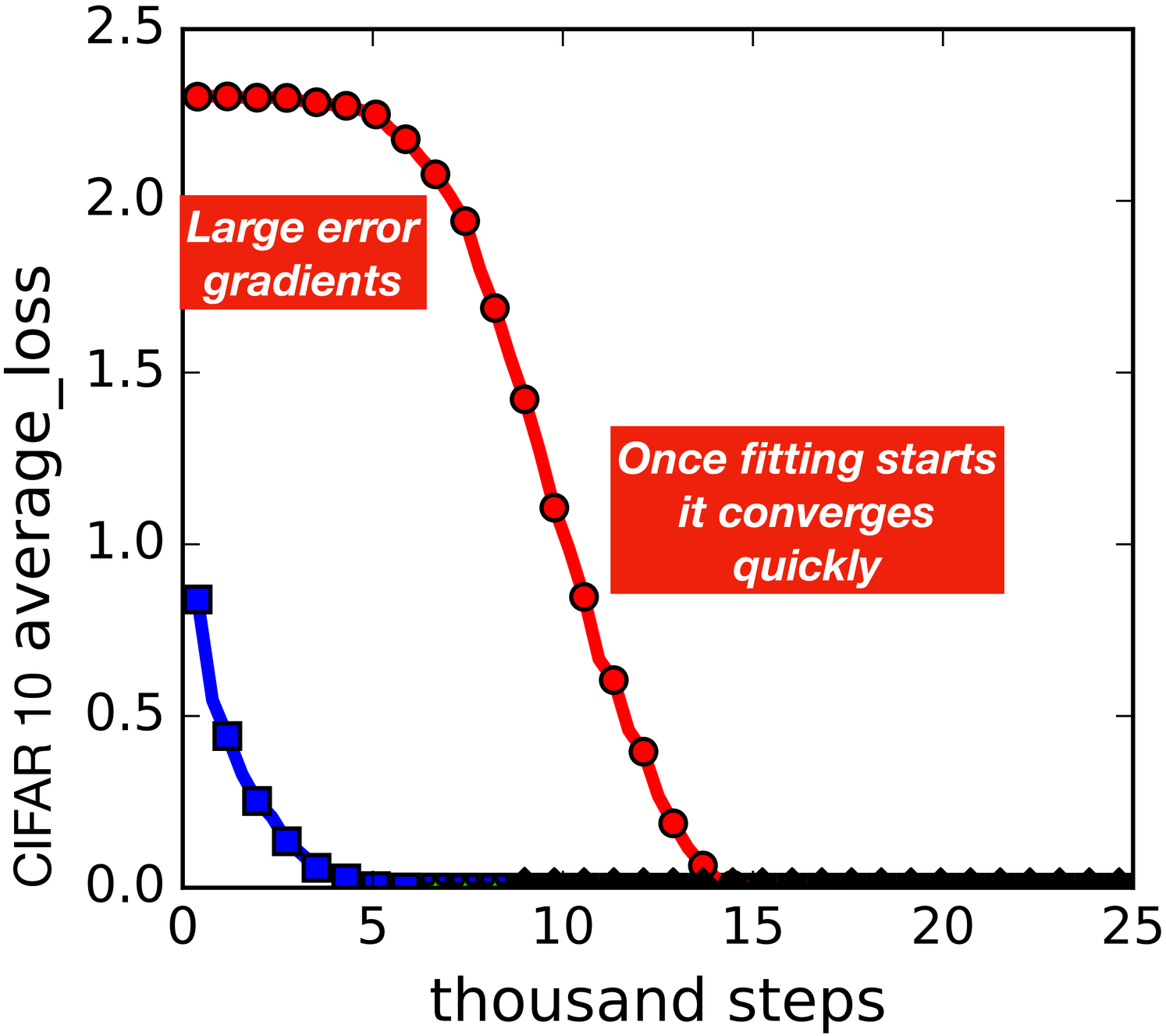
What do we expect?

- We’ve decoupled the input and the output so the network should do badly even on training?
- There is a local structure inductive bias in the networks for image data that we break with random inputs…
- Error gradients should be crazy - will we even be able to learn anything?

What do we expect?

- We've decoupled the input and the output so the network should do badly even on training?
- There is a local structure inductive bias in the networks for image data that we break with random inputs…
- Error gradients should be crazy - will we even be able to learn anything?

- Neural nets are universal approximators so it doesn't matter that the inputs and outputs are decoupled. Just fitting a very non smooth function
- We're in the overparameterized regime with regularization turned off - training error should be zero

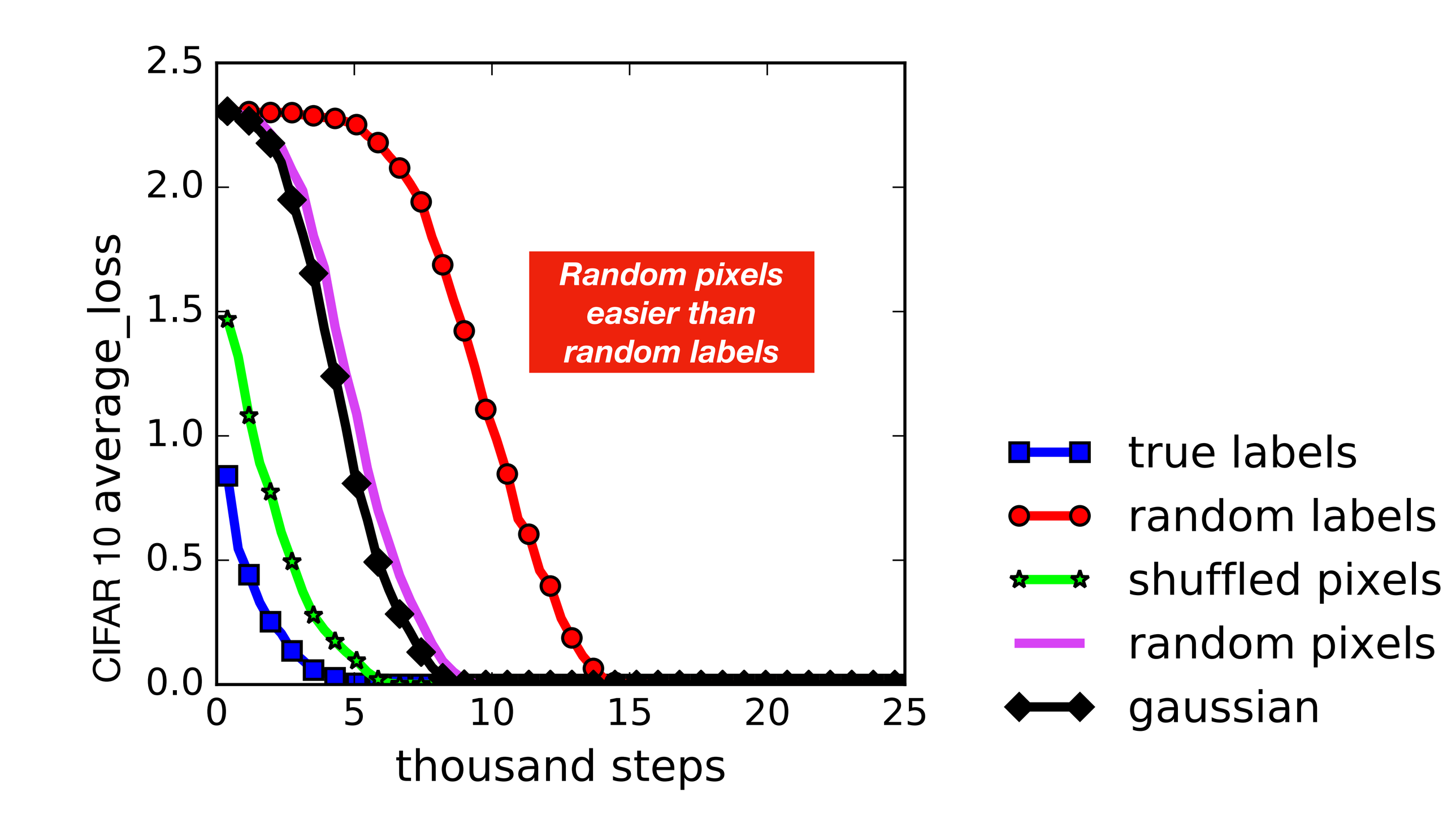


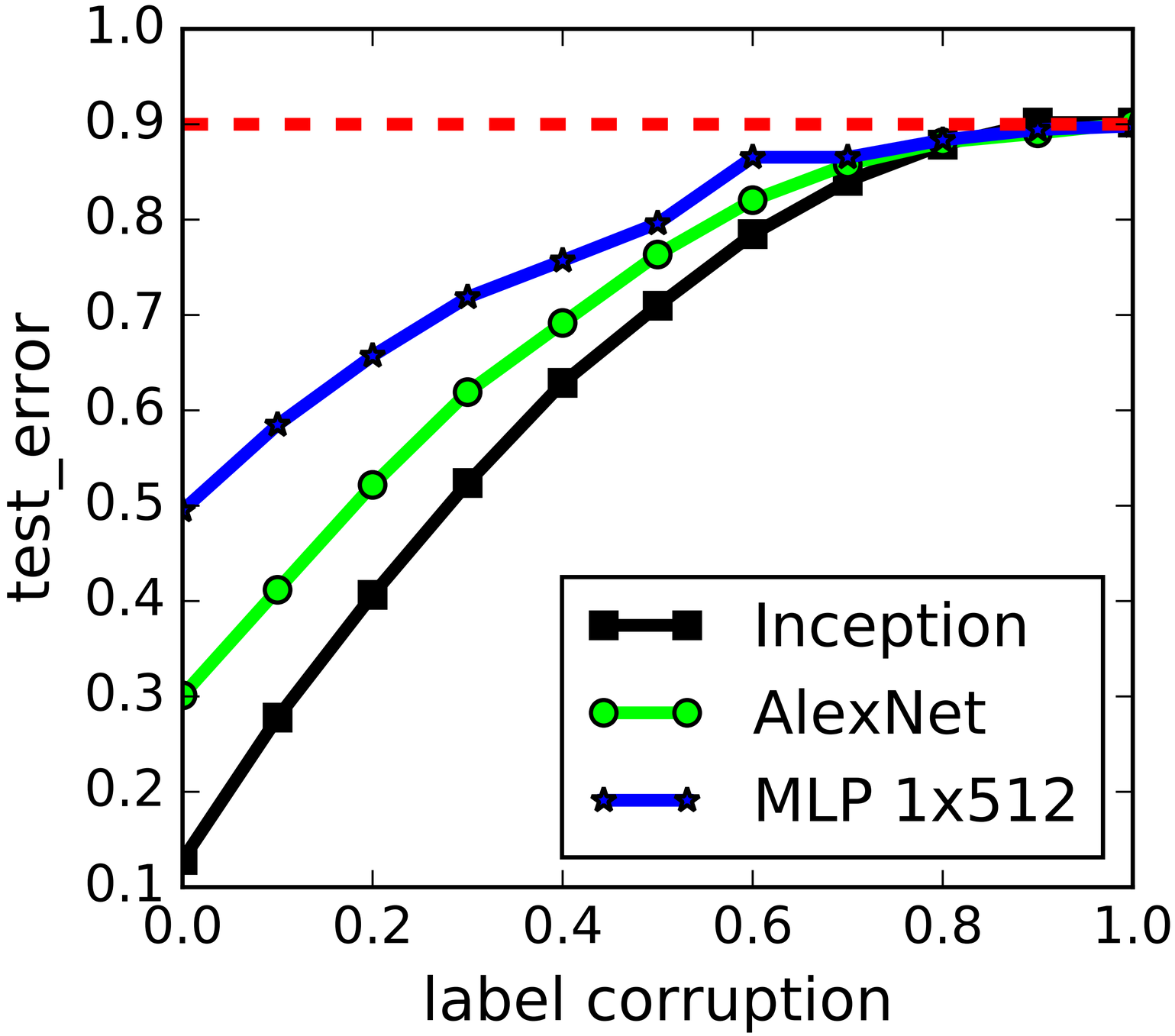
It takes 3 times as long… but you can still drive training error to 0

true labels

random labels

On ImageNet 95.2% top-1 accuracy


2.5
2.0
1.5
1.0
0.5
0.0
CIFAR 10 average_loss
0
5
10
15
20
25
thousand steps
Random pixels
easier than
random labels
true labels
random labels
shuffled pixels
random pixels
gaussian


1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
test_error
0.0
0.2
0.4
0.6
0.8
1.0
label corruption
Inception
AlexNet
MLP 1x512

Why does this matter?

Generalization bounds give probabilistic guarantees for how well we can expect our classifiers to perform on test data.

But we need a way of talking about how “wiggly” our function can get.

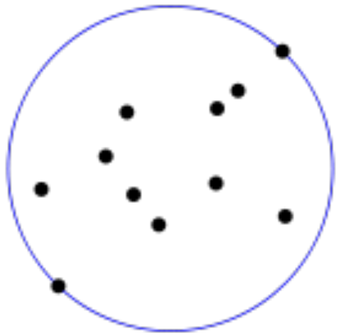
One way of formalizing “wiggliness” - **Rademacher** complexity.

Hans Rademacher

$$\hat{\mathfrak{R}}_n(\mathscr{H}) = \mathbb{E}_\sigma \left[\sup_{h \in \mathscr{H}} \frac{1}{n} \sum_{i=1}^{n} \sigma_i h\left(x_i\right) \right]$$

Randomization tests show $\hat{\mathfrak{R}}_n(\mathscr{H}) \approx 1$ for the models we use, so standard bounds don’t tell us anything

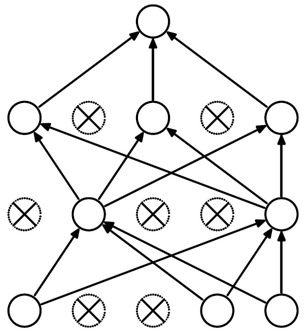
Okay… but what about regularization?



Regularization motivation: control the size of the hypothesis class.

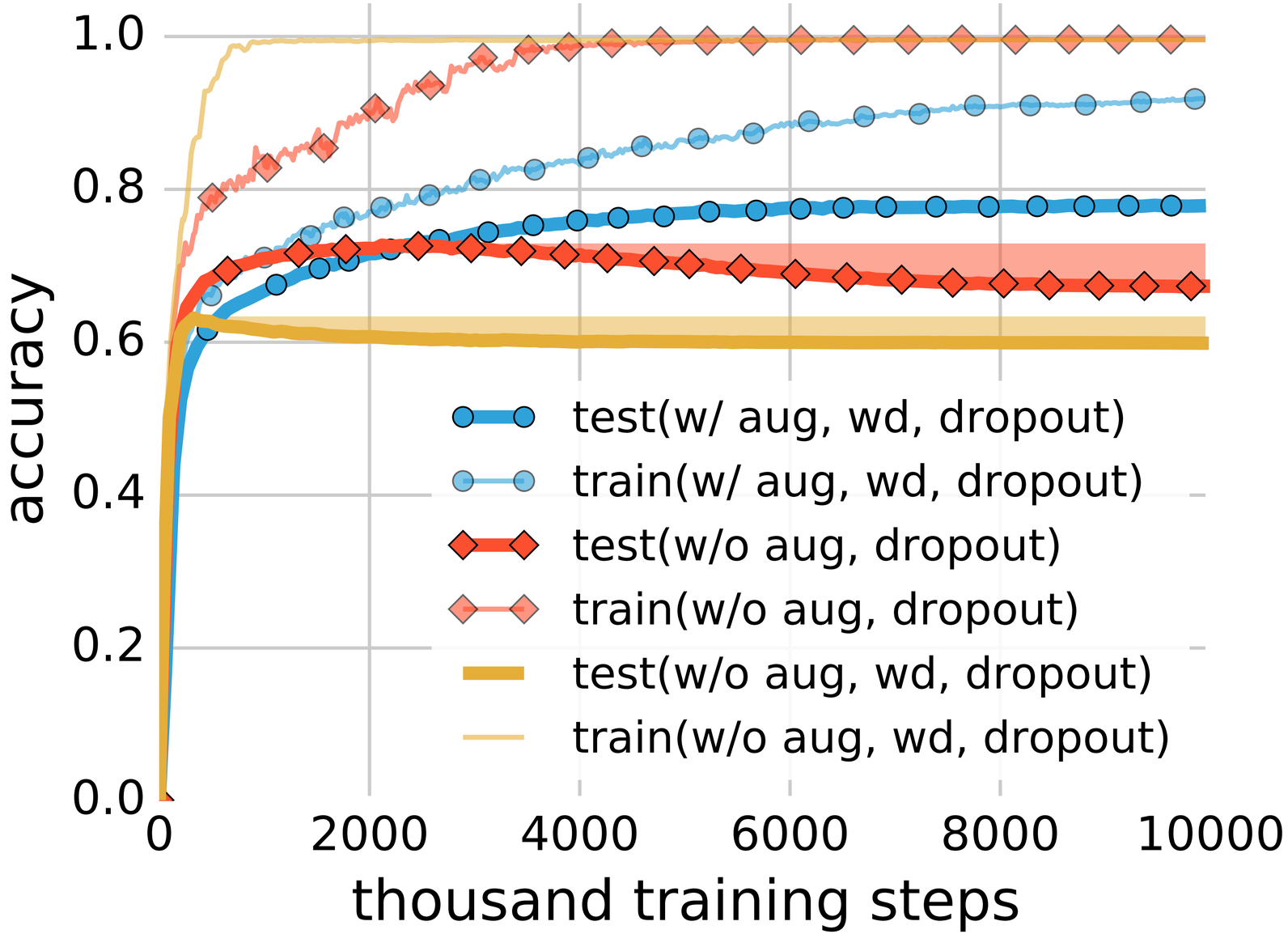
- **Weight decay** (l_2 regularization). equivalent to a hard constrain of the weights to an Euclidean ball.

- **data augmentation** use domain-specific transformations. e.g. random cropping, random brightness, saturation, etc.



- **dropout** mask out each element of a layer output randomly with a given dropout probability.

Do they prevent the massive overfitting we saw in the randomization tests?



(a) Inception on ImageNet

data aug	dropout	weight decay	top-1 train	top-5 train	top-1 test	top-5 test
ImageNet 1000 classes with the original labels						
yes	yes	yes	92.18	99.21	77.84	93.92
yes	no	no	92.33	99.17	72.95	90.43
no	no	yes	90.60	100.0	67.18 (72.57)	86.44 (91.31)
no	no	no	99.53	100.0	59.80 (63.16)	80.38 (84.49)
Alexnet (Krizhevsky et al., 2012)			-	-	-	83.6
ImageNet 1000 classes with random labels						
no	yes	yes	91.18	97.95	0.09	0.49
no	no	yes	87.81	96.15	0.12	0.50
no	no	no	95.20	99.14	0.11	0.56

Some theoretical results…

Finite sample expressivity

Theorem 1. *There exists a two-layer neural network with ReLU activations and* $2\mathrm{n} + \mathrm{d}$ *weights that can represent any function on a sample of size* n *in* d *dimensions*.

i.e. neural networks can perfectly interpolate the training set of any function in the over-parameterized regime.

Key proof idea:

Lemma 1. *For any two interleaving sequences of* n *real numbers* $b_1 < x_1 < b_2 < x_2 \cdots < b_n < x_n$, *the* $n \times n$ *matrix* $A = [\max\{x_i - b_j, 0\}]_{ij}$ *has full rank. Its smallest eigenvalue is* $\min_i x_i - b_i$.

The role of implicit regularization

- What role does SGD play in generalization performance?
- SGD update: $w_{t+1} = w_t - \eta_t e_t x_{i_t}$
- If $w_0 = 0$ then: $w = \sum_i^n \alpha_i x_i = X^T \alpha$ so w lies in the span of the data points x.
- If we enforce interpolation, we have $Xw = y$ and so $XX^T \alpha = y$ which we can solve exactly in the linear case. This is also the minimum *l2* norm solution to $Xw = y$.

data set	pre-processing	test error
MNIST	none	1.2%
MNIST	gabor filters	0.6%
CIFAR10	none	46%
CIFAR10	random conv-net	17%

Conclusions

- Overparameterized neural nets lead to vacuous generalization bounds
- Regularization helps with test set generalization performance but doesn’t significantly change empirical Rademacher complexity (and hence doesn’t affect bounds).
- Some evidence that SGD acts as a implicit regularizer in the linear case.