Compression Approaches to Generalization Questions in Deep Learning

SANJEEV ARORA

PRINCETON UNIVERSITY

http://www.cs.princeton.edu/~arora/

Group website: unsupervised.princeton.edu

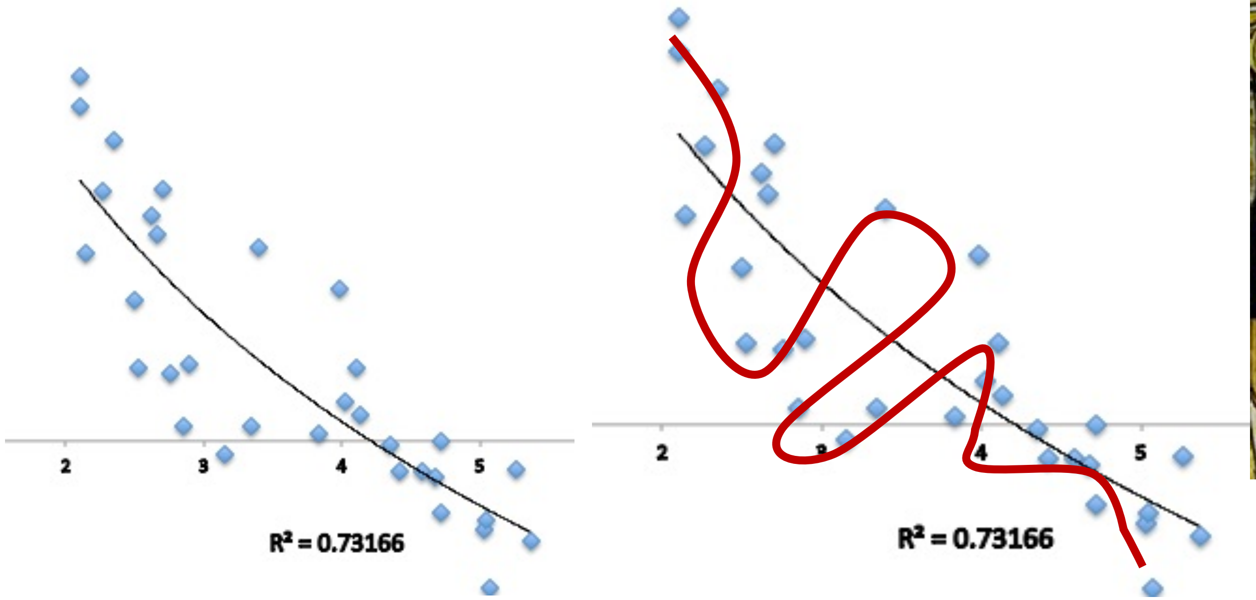
Blog: www.offconvex.org

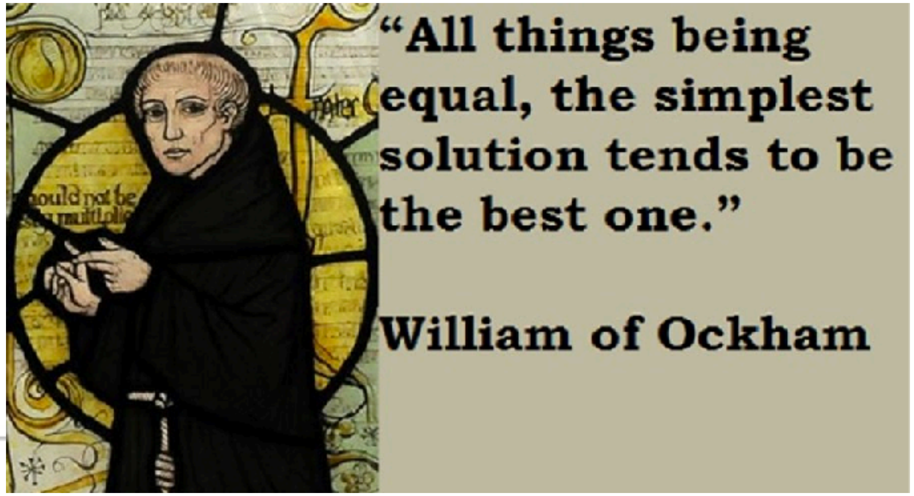
Twitter: @prfsanjeevarora

Grant support: NSF, Simons Foundation, ONR, DARPA/SRC, AWS

NO

Motivation: Overfitting mystery of deep learning

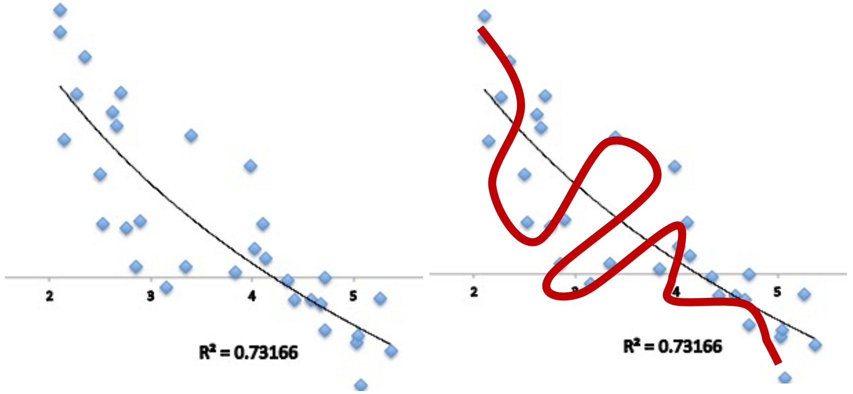




Rule of thumb: Overcomplicated models (e.g. when # parameters >> # datapoints) overfit and do not "generalize" to explaining new data.

Overparametrized nets (capable of fitting even random data [Zhang et al.17]) outperform smaller nets.

Generalization Bounds in Nutshell



Data distribution $\mathscr{D}$.
$\ell_\theta(x) =$ Loss of deep net θ on labeled datapoint x

Test loss/Population Loss $= E_{x \in \mathscr{D}}[\ell_\theta(x)]$ Training loss on sample S $= E_{x \in S}[\ell_\theta(x)]$

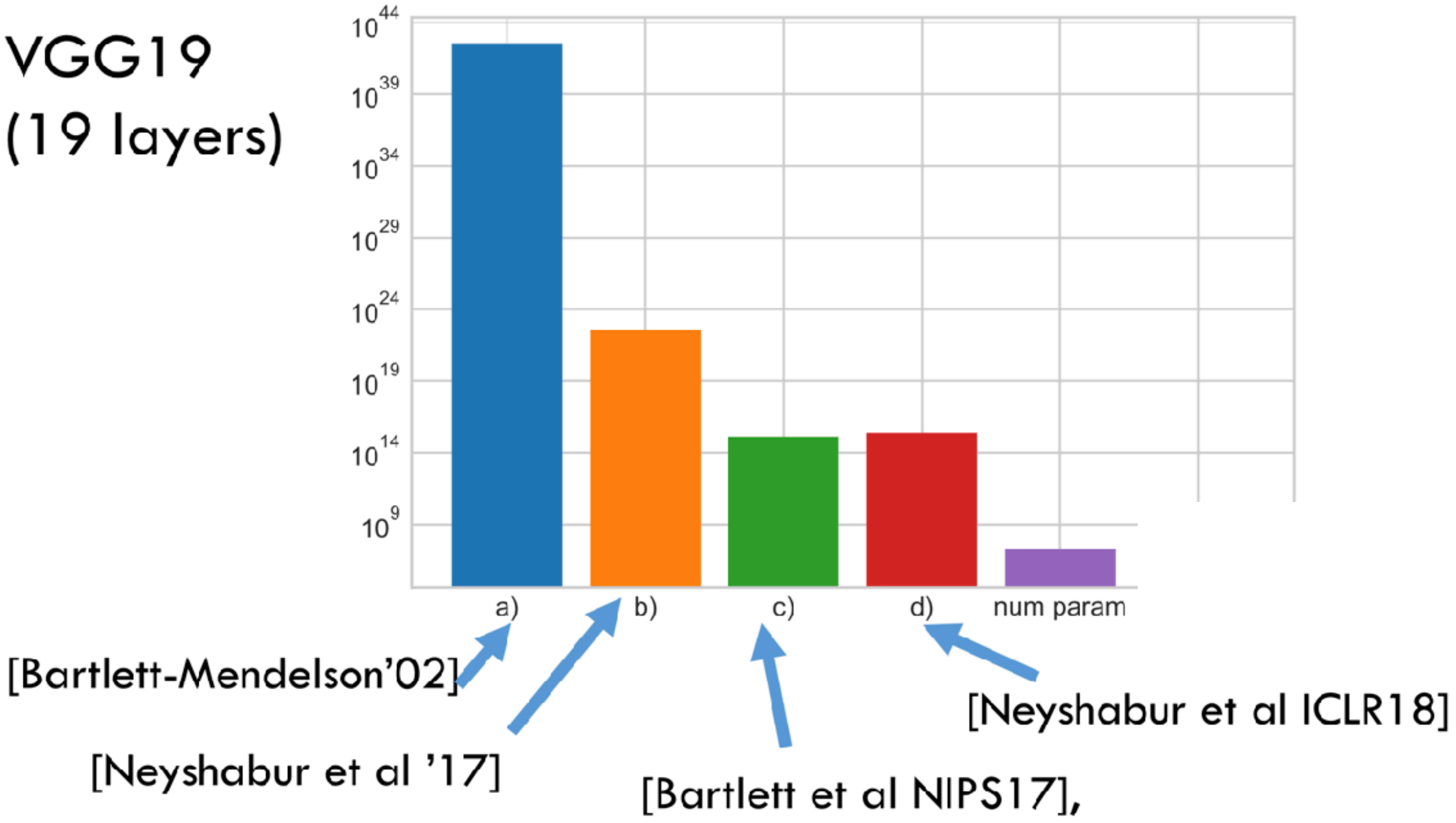
Generalization error = Test loss - Training Loss

Typical upper bound has form: $\sqrt{\frac{C(\theta)}{m}} +$ sampling error

Examples of $C(\theta)$ include
parameters, $||\theta||_2$ etc.

$C(\theta)$ = Estimate of complexity of θ ("true" # of parameters)
$m =$ Size of training set

Search for non-vacuous estimates of $C(\cdot)$



Main conceptual hurdle

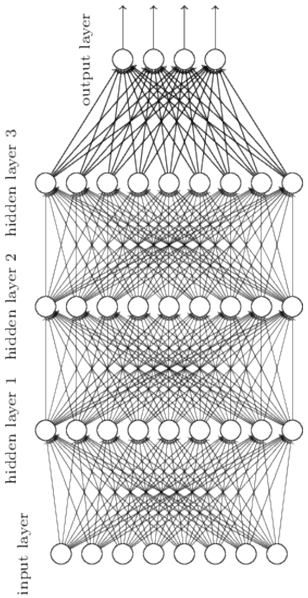
While training, deep nets appear to pick up “irrelevant” information about datapoints.
(See cute example from Nagarajan-Kolter’19, even for linear regression)

Takeway: any kind of norm of parameter vector θ seems very pessimistic estimate of its “true complexity” wrt the task

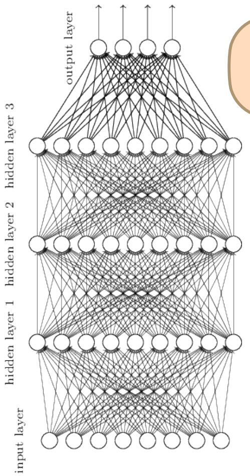
Estimating “true complexity” must involve some form of compression.
(Indeed, the powerful PAC-Bayes method [McAllester’99] is information theoretic.)

Compression-based method

[A., Ge, Neyshabur, Zhang ICML'18] "user-friendly PAC-Bayes"



#parameters $\gg$ #data points



Generalizes!!

#parameters $\ll$ #data points

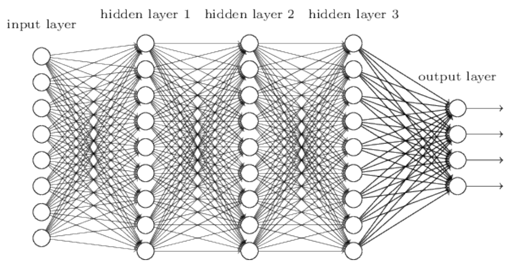
only small change in training error

Important: Compression method and its randomness are fixed before seeing the training data

(No retraining after compression.)

Noise stability experiment

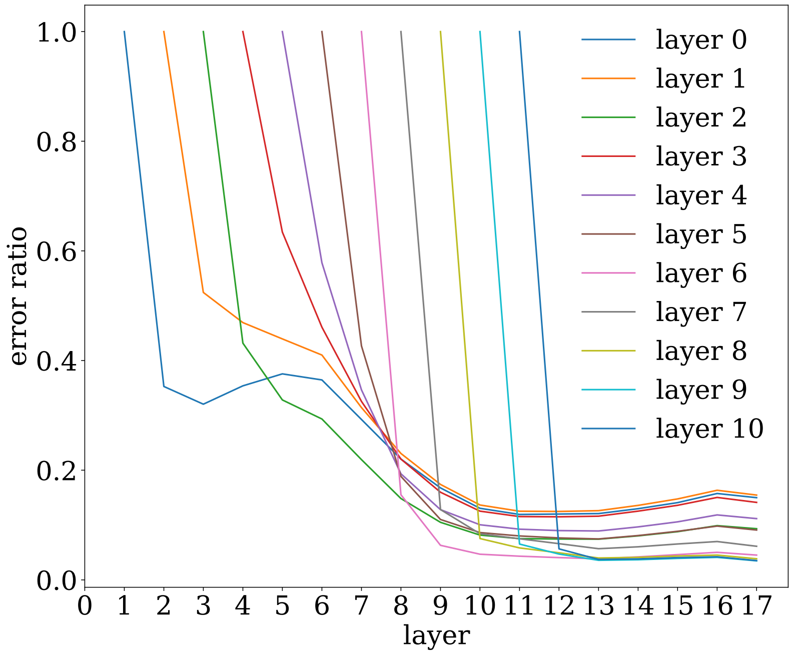
[A., Ge, Neyshabur, Zhang ICML'18]



Take trained net

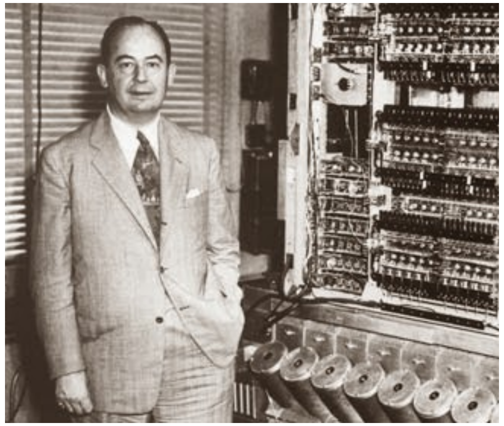
Noise injection: Add gaussian η to output x of a layer ($|\eta| = |x|$)

Measure percent change in higher layers.
(If small, then net is noise stable.)



Results for VGG19 (19 layers)

(Similar results for other architectures)

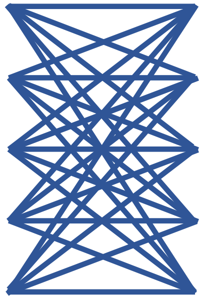


Von Neumann, J. (1956).
Probabilistic logics and the synthesis of reliable organisms from unreliable components.

Key Insight: Can improve reliability of circuits by allowing redundancy.

Noise stability experiment suggests great redundancy inside trained nets!

Noise stability: understanding one layer (no nonlinearity)



η : Gaussian noise

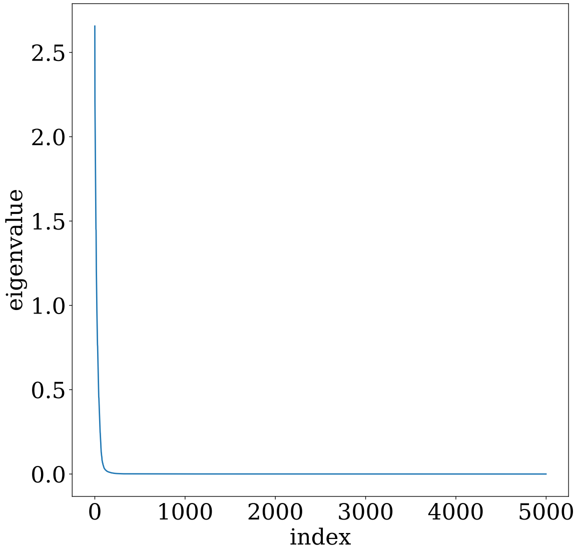
$|Mx|/|x| \gg |M\eta|/|\eta|$

$\wedge\!/ \qquad\qquad\qquad\qquad ||$

$\sigma_{max}(M) \qquad (\sum_i \sigma_i(M)^2)^{1/2}/\sqrt{n}$

"Stable Rank"

Layer Cushion = ratio
(roughly speaking..)



Distribution of singular values in a filter of layer 10 of VGG19. Such matrices are compressible…

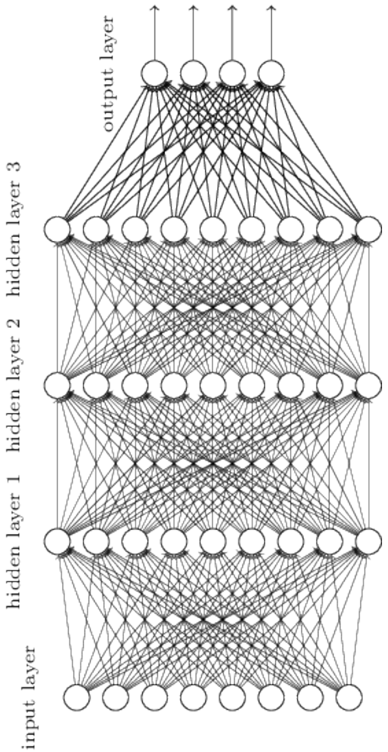
Noise stability → deep net can be made low-dimensional (minimal change to training error)

Idea 1: Compress a layer (randomized; errors introduced are "Gaussian like")

Idea 2: Errors attenuate as they go through network, due to noise stability. So output changed not much.

Compression:
(1) *Generate k random sign matrices $M_1, \ldots, M_k$ (impt: picked before seeing data)*

(2) $\hat{A} = \frac{1}{k}\sum_{t=1}^{k} \langle A, M_t \rangle M_t$

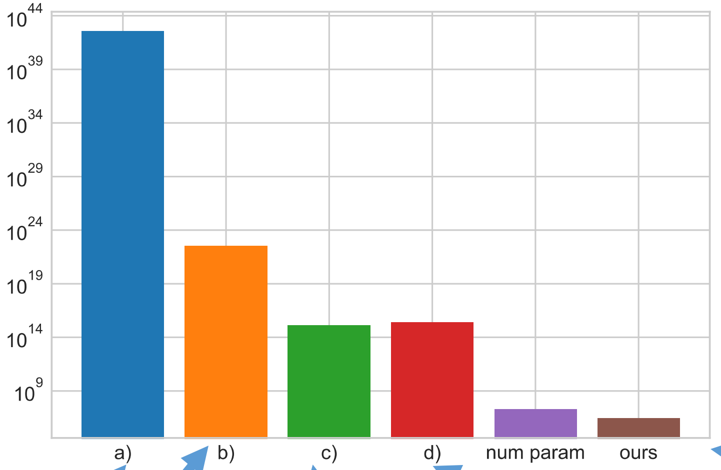


k is logarithmic in original size, so matrix becomes low-dimensional

The Quantitative Bound

$$\text{capacity} \approx \left(\frac{\text{depth} \times \textit{activation contraction}}{\text{layer cushion} \times \text{interlayer cushion}} \right)^2$$

VGG19
(19 layers)



[Bartlett-Mendelson'02]

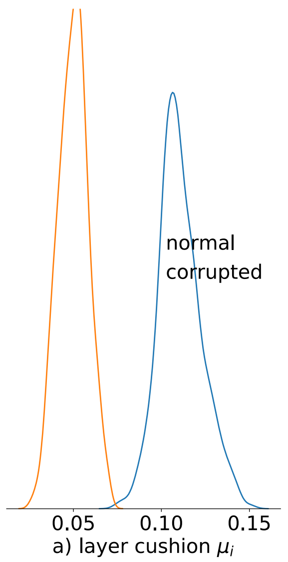
[Neyshabur et al'17]

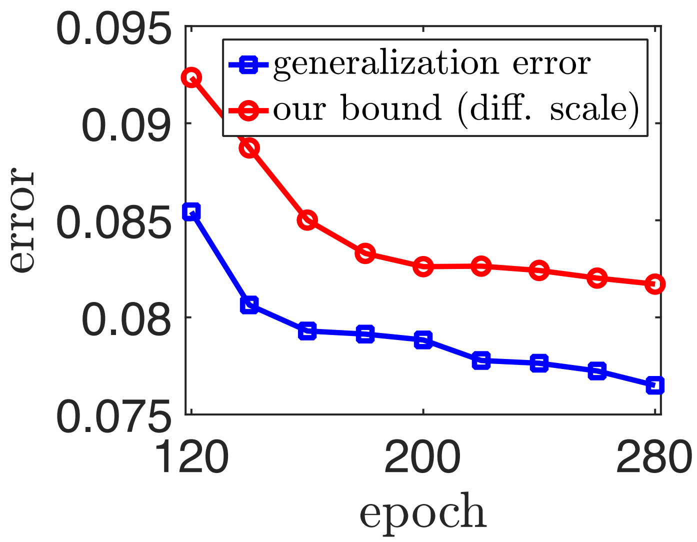
[Bartlett et al'17],

[Neyshabur et al'17]

[A., Ge, Neyshabur, Zhang'18]

Correlation with Generalization (qualitative check)





Layer cushion much higher when trained on normal data than on corrupted data

Evolution during training on normal data..

Concluding thoughts on generalization

Final story still to be written; quantitative bounds too weak to explain generalization with 20M parameters on 50k datapoints.

My Current view: Correct argument needs to include insight about data distribution and/or training algorithm.

(There's a flurry of work, including from my group, about how dynamics of training algorithm leads to nontrivial generalization behavior; see my posts on offconvex.org)

Part 2: Rip Van Winkle's Razor

(A new estimate for Adaptive Data Analysis)

Rip Van Winkle's Razor: A Simple Estimate of Overfit to the Holdout
[A. and Yi Zhang, 2021. Appearing on arxiv today.]

" *Thou shalt not train on the holdout set…* "

Old proverb in Stats Land about "Data Hygiene"

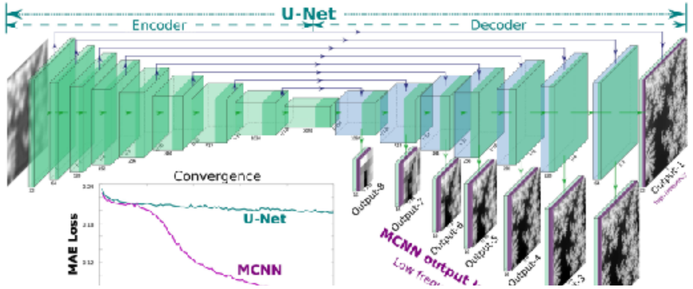
(Dwork et al'15): "Is Machine Learning effectively training on the holdout set ??"

(Also related: "reproducibility crisis in social science" "p-value hacking", "garden of forking paths"…) [Gelman-Loken'13]

Unchanged since 2013

Design-Test-Redesign Cycle

Deep Learning Maven



AlexNet VGG GoogLeNet ResNet DenseNet

Model gets published if test error is good.

Next model builds upon it. (architecture ideas, training algo, even actual code….)

Millions of models that didn't work out were discarded

Statistical guarantees breaks down.
"p-value hacking",
"garden of forking paths"…
[Gelman-Loken'13]

Meta-Overfitting Error (MOE)

$E_{x \in \mathcal{D}}[\ell_\theta(x)]$ $-$ $E_{x \in \text{holdout}}[\ell_\theta(x)]$

Population loss Holdout Loss

Note: Concentration bounds $\Longrightarrow$ MOE minuscule if data hygiene were observed. (holdout set has size > $10k$!)

Dwork et al. 15: If Holdout set has size N and model designer can use previously tested models then a sequence of just t models can yield a model with Holdout loss $O(\sqrt{t/N})$

# of models tested on ImageNet Holdout is estimated to be $10^7 - 10^8$!!

*** Recht et al ICML'19: "Do Imagenet classifiers generalize to ImageNet?"*

6+ years of "Adaptive Data Analysis" (crude summary)

- MOE can be kept low* if experimenters fastidiously follow special protocols while accessing the holdout (Differential Privacy, "Look only at the leader", "Forget older models",…)
- …..But real-life experimenters don't do this, and impossible to verify anyway.
- Empirical evidence** that MOE may not be large; based upon attempted constructions of new holdout set for ImageNet and other datasets. (But, new holdout gave very different accuracy numbers (systematic issues)!)

* [Dwork et al'15], [Blum,Hardt'15], [Bassily et al'16], [Zrnic,Hardt'19]……

** *Recht et al ICML'19: Do Imagenet classifiers generalize to ImageNet?*

Needed: Better Estimates of MOE

1) Useful guidance to scientists in other disciplines with fixed public datasets used by hordes of researchers. (eg, genomics, astronomy, finance etc.).

2) In many settings (one-time phenomena, e.g. in finance or astronomy) it is impossible to create a new Holdout set ever again (ie what Recht et al tried to do for ImageNet)

How about description length?

Theorem(informal, folklore): If a model has description length k bits and holdout set has size N, then with high probability, MOE of this model is at most $2\sqrt{k/N}$

(* immaterial how the model was created!!s)

What is the description length of ResNet-34?

Option 1: k := # of param in the model, 20M -> vacuous

Option 2: k:= size of the code that produced it

Difficulty: "size of code" must include all called libraries (cannot exclude libraries because they were written post 2013 ; implicitly "contaminated" by ImageNet).

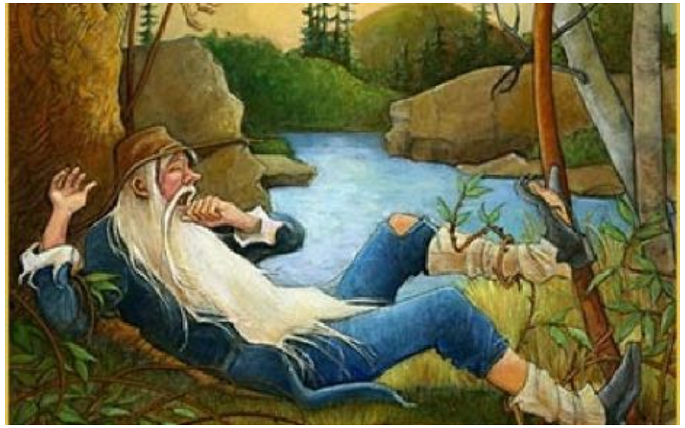
Description length via reproducibility

(Journal/conference referees must be able to reproduce the model from the description)

Description Length = # of bits in a description that allows net of same performance to be constructed by a suitable referee using ImageNet training set.

1) INFORMED:
Knows everything known (e.g. about deep learning, math, optimization, statistics etc.) right up to moment of creation of ImageNet Holdout set (2012)

1) UNBIASED:
Fell asleep at that moment and knows nothing that's happened since then.



"Rip van Winkle's Razor"

Describing modern deep nets to Rip van Winkle

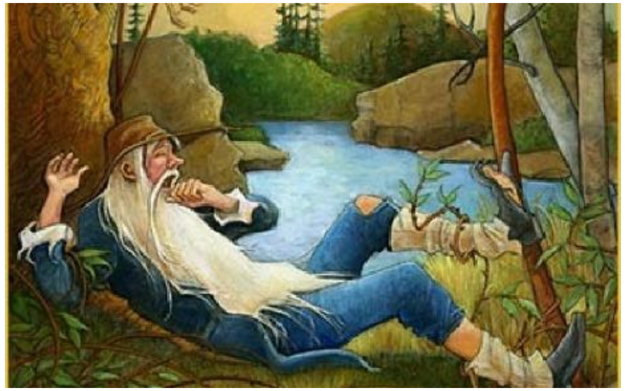
(goal: bit length via Huffman coding should be small!!)

Normal English (preferably "Simple English," with vocabulary size ~ 1k) ✓

Small vocabulary that was well-known in DL in 2012: basic math, convolution, ReLU, layer, gradient, layer, stride, SGD, learningrate, epoch, weight, pixel, downsampling, etc.… ✓

Concepts developed since 2012 (e.g., Batch Norm, Residual Layers,..) need to be defined before first use.

Hardware-specific details can be left out (assuming they only affect total training time, not final accuracy)



ResNet34 Description

Batch-Norm(BN):
bn:= at each node apply the function(x)
$x := b + g * (x - mean)/\sqrt{var + 0.01}$
mean := batch mean of node, variance := batch variance,
b, g are trainable, init b := 0, g := 1
Architecture:
Layer := convolution then BN then relu
block(k) on input a := a+ two layers 3*3 convolution k channels (a)
block-downsample(k):= downsample(a) by 2 + two layers 3*3 convolution k channels (a), first layer has stride 2
Forwardpass:
7x7 convolution 64 channel stride 2 pool(2) block(64) repeats 3
block-downsample(128) block(128) repeats 3
block-downsample(256) block(256) repeats 5
block-downsample(512) block(512) repeats 5
avg pool fully-connected 1000 softmax
Initialization: Xavier
DataAugmentation:
rescale: mean = 0, variance = 1
SVD 3x3 covariance matrix of RBG pixels is $\lambda_i,\ v_i,$
Add to each pixel noise $\sum_{i=1}^{3} \alpha_i \lambda_i v_i,\ \alpha_i \sim N(0, 0.1)$ drawn once for each image.
Training:
SGD batchsize 256 weight-decay 0.0001 momentum 0.9 iteration 60e4
learningrate init 0.1; every 30 epochs learningrate -> learningrate/10
Testing:
convolve net with image to at each scale of 224 256 384 480 640 (with horizontal flip)
to get in total 50 logits
final logits := average over all the logits

190 words ($\approx$ 2.5k bits)

MOE estimate < 7%

- **Elementary**
- **No prior technique in adaptive data analysis yielded any nonvacuous estimates**
- **Seems applicable to other areas of science with reasonable sized datasets**

Conclusions/Takeaways

- Compression arguments seem to be at the heart of understanding generalization phenomena in deep learning. (PAC-Bayes bound of McAllester is a concrete phrasing)
- Can yield simple and striking insights that seem hard to attain by other means.

THANK YOU!