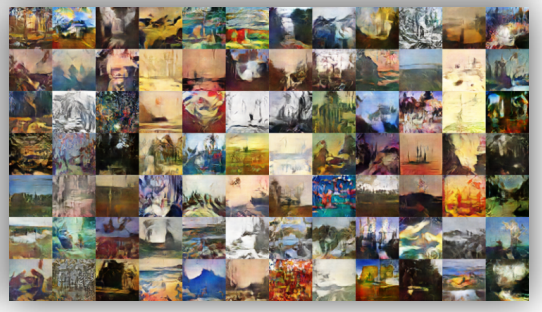
Generalization in Deep Learning

Behnam Neyshabur

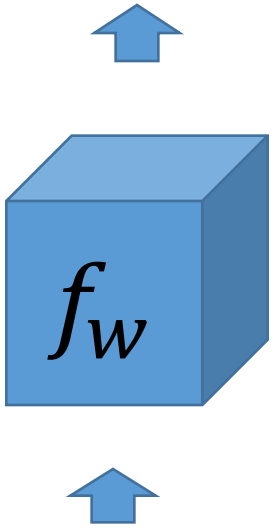
Institute for Advanced Study & NYU

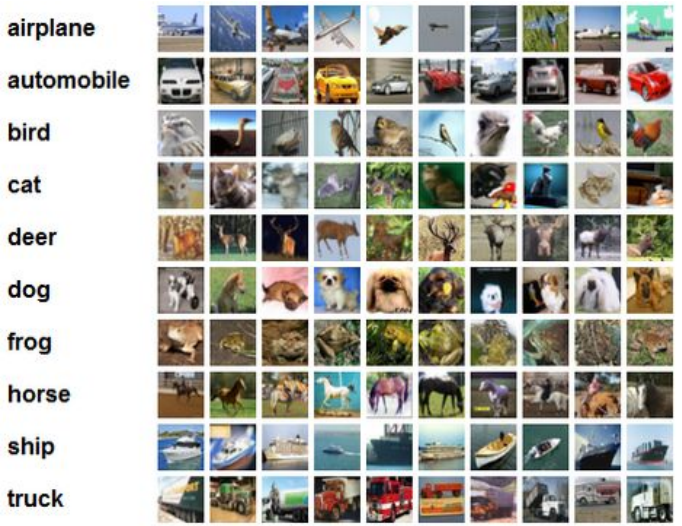
AI≈Deep Learning≈Neural Networks

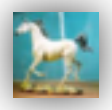


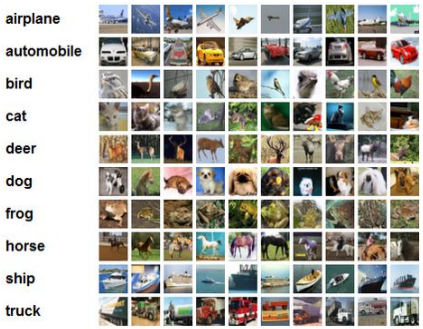
Classification

	airplane	car	bird	cat	deer	dog	frog	horse	ship	truck
prediction: $f_w(x) =$	0.01	0.01	0.01	0.03	0.2	0.1	0.02	0.6	0.01	0.01



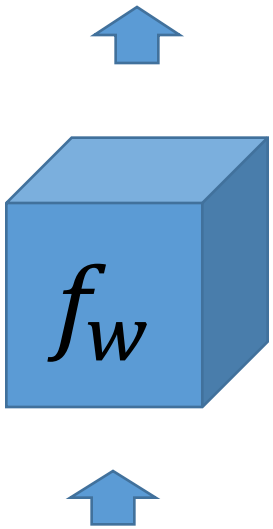


Input: $x =$ 

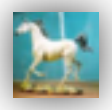


Classification

	airplane	car	bird	cat	deer	dog	frog	horse	ship	truck
prediction: $f_w(x) =$	0.01	0.01	0.01	0.03	0.2	0.1	0.02	0.6	0.01	0.01



- Example: $f_W(x) = \mathrm{W}\mathrm{x}, \quad x \in \mathbb{R}^D, W \in \mathbb{R}^{10 \times D}$
- Loss function: $\mathrm{L}(f_w(x), y = "horse")$

Input: $x =$ 

Learning Framework

- x: input, y:label
- Unknown distribution $(x, y) \sim D$
- Training set $S = \{(x_1, y_1), \ldots, (x_m, y_m)\}$

Goal: Find a predictor $f: \mathcal{X} \to \mathcal{Y}$ with minimum test err(f) expected error for $(x, y) \sim D$

generalization error(f)

$$test\ err(f) = training\ err(f) + test\ err(f) - training\ err(f)$$

Calculated on the distribution

Calculated on the sample

Hoeffding's Inequality

- Let $X_1, \ldots, X_m$ be independent random variables.
- Let g be any function such that $0 \leq g(X_i) \leq 1$.
- Define empirical mean $\bar{g} = \frac{1}{m}(g(X_1) + \cdots + g(X_m))$.

$$P(E[\bar{g}] - \bar{g} > \epsilon) \leq \exp(-m\epsilon^2) = \delta$$

w.p. $\underbrace{1-\delta}_{0.99}$ over $\bar{g}$: $\quad E[\bar{g}] \leq \bar{g} + \underbrace{\sqrt{\frac{\ln(1/\delta)}{m}}}_{\leq 0.1}$

$m \geq 500$

See "CS229 Supplemental Lecture notes Hoeffding's inequality" by John Duchi for the proof.

Generalization of a Classifier

- For any f, w.p. $1-\delta$ over the choice of the training set of size m,

$$L(f) \leq \hat{L}(f) + \sqrt{\frac{\ln(1/\delta)}{m}}$$

($L(f)$: test err; $\hat{L}(f)$: training err)

- The above statement is true for any f. Aren't we done?
- The order of quantifiers matters! f is must be chosen before seeing the training set.
- How to fix this issue?

Generalization of a Function Class

- Fix class F. We know that: $\forall_f, \mathrm{P}(\mathrm{L(f)} - \hat{L}(f) > \epsilon) \leq \exp(-m\epsilon^2)$

$$\mathrm{P}(A \cup B) \leq P(A) + P(B)$$

$$\mathrm{P}(\exists_{f \in F} \mathrm{L(f)} - \hat{L}(f) > \epsilon) \leq \sum_{f \in F} \mathrm{P}(\mathrm{L(f)} - \hat{L}(f) > \epsilon) \leq |F| \exp(-m\epsilon^2)$$

For any F, w.p. $1 - \delta$ over the choice of the training set of size m,

$$\forall_{f \in F} \quad \underbrace{L(f)}_{\textit{test err}} \leq \underbrace{\hat{L}(f)}_{\textit{training err}} + \sqrt{\frac{\ln|F| + \ln(1/\delta)}{m}}$$

What if the size of F is infinite?

Generalization of a Function Class

- Fix class F. We know that: $\forall_f, \mathrm{P}(\mathrm{L(f)} - \hat{L}(f) > \epsilon) \leq \exp(-m\epsilon^2)$

$$\mathrm{P}(A \cup B) \leq P(A) + P(B)$$

$$\mathrm{P}(\exists_{f \in F} \mathrm{L(f)} - \hat{L}(f) > \epsilon) \leq \sum_{f \in F} \mathrm{P}(\mathrm{L(f)} - \hat{L}(f) > \epsilon) \leq |F| \exp(-m\epsilon^2)$$

Finite Precision

For any F, w.p. $1 - \delta$ over the choice of the training set of size m,

$$\forall_{f \in F} \quad \underbrace{L(f)}_{test\ err} \leq \underbrace{\hat{L}(f)}_{training\ err} + \sqrt{\frac{O(\#\mathrm{param}) + \ln(1/\delta)}{m}}$$

What if the size of F is infinite?

Growth Function

- Idea: We count number of ways m points can be classified!
- Growth function ($\Gamma_m(F)$): number of ways m points can be classified by functions in F

For any F, w.p. $1-\delta$ over the choice of the training set of size m,

$$\forall_{f \in F} \quad \underbrace{L(f)}_{test\ err} \leq \underbrace{\hat{L}(f)}_{training\ err} + O\left(\sqrt{\frac{\log \Gamma_m(F) + \ln(1/\delta)}{m}}\right)$$

How can we calculate $\Gamma_m(F)$?

VC-dimension

- VC-dim(F): cardinality of the largest set of points that can be shattered by F.

shatter: generate all possible labels.

- Example: Let F be the set of linear separators of dimension d:

$$F = \{f(x) = \text{sign}(\langle w, x\rangle + b) \,|\, w \in R^d, b \in R\}$$

$$\text{VC-dim}(F) = \text{d} + 1$$

For any F, w.p. $1-\delta$ over the choice of the training set of size m,

$$\forall_{f \in F} \quad \underbrace{L(f)}_{\textit{test err}} \leq \underbrace{\hat{L}(f)}_{\textit{training err}} + \tilde{O}\left(\sqrt{\frac{\text{VC}-\dim(F) + \ln(1/\delta)}{m}}\right)$$

Learning Framework

- x: input, y:label
- Unknown distribution $(x, y) \sim D$
- Training set $S = \{(x_1, y_1), \ldots, (x_m, y_m)\}$

Goal: Find a predictor $f: \mathcal{X} \to \mathcal{Y}$ with minimum expected error for $(x, y) \sim D$ (expected error: test err(f))

$$test\ err(f) = training\ err(f) + test\ err(f) - training\ err(f)$$

($test\ err(f) - training\ err(f)$: generalization error(f))

- We choose predictor f from the model class F.

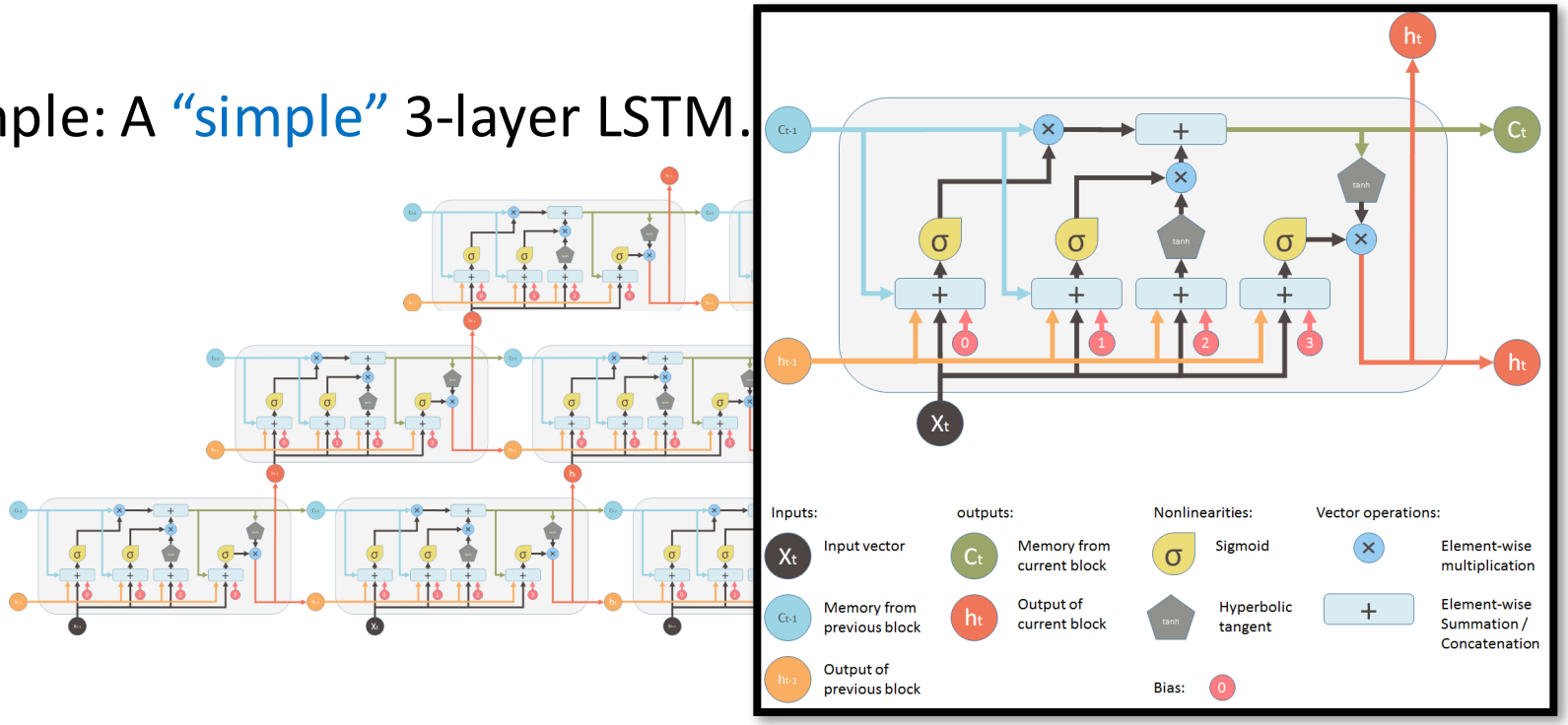
$$test\ err(f) \leq training\ err(f) + c\sqrt{\frac{\text{capacity}}{m}}$$

size of F ⇧ $training\ err(f)$ ⇩ generalization error(f) ⇧

Deep Learning

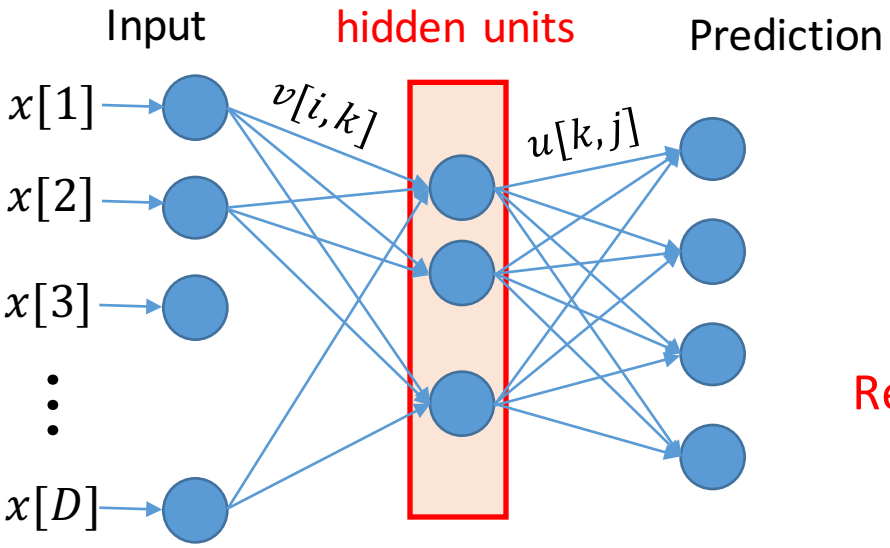
- Deep Learning Model f_w:
  - Composition of differentiable parameterized functions
  - Over-parameterized: tens of millions of parameters

- Example: A "simple" 3-layer LSTM.



source: https://medium.com/@shiyan/understanding-lstm-and-its-diagrams-37e2f46f1714

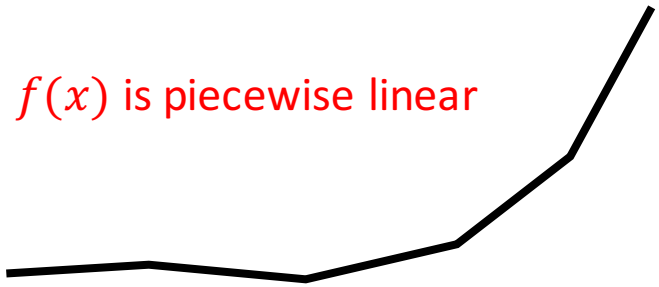
Fully Connected Networks



$$f(x) = U[Vx]_+$$

ReLU activation: $[z]_+ = \max\{0, z\}$

$f(x)$ is piecewise linear



Learning Fully Connected Networks

Fix architecture (depth and #hidden units)

$$F = \{ f_w(x) = output\ of\ net\ with\ weights\ w \}$$

$$test\ err(f) = training\ err(f) + generalization\ error(f)$$

- Expressive Power / Approximation
  - Any continues function with a large enough network

- Computation / Optimization
  - Even if function exactly representable with single hidden layer with $\Theta(\log D)$ units: no poly-time algorithm always works
    [Kearns Valiant 94; Klivans Sherstov 06; Daniely Linial Shalev-Shwartz '14]

- Capacity / Generalization ability / Sample Complexity
  - VC-dim: O(depth × number of weights) [Bartlett et al.'17]

* Bartlett, Peter L., et al. "Nearly-tight vcdimension and pseudodimension bounds for piecewise linear neural networks." *arXiv* (2017).

Simple Experiment

Task: Handwritten digit classification

Input: image of a digit $x \in \mathbb{R}^{1024}$

Labels: 0-9

Training Set: 60000 images

Optimization: Stochastic Gradient Descent

- Initialize:

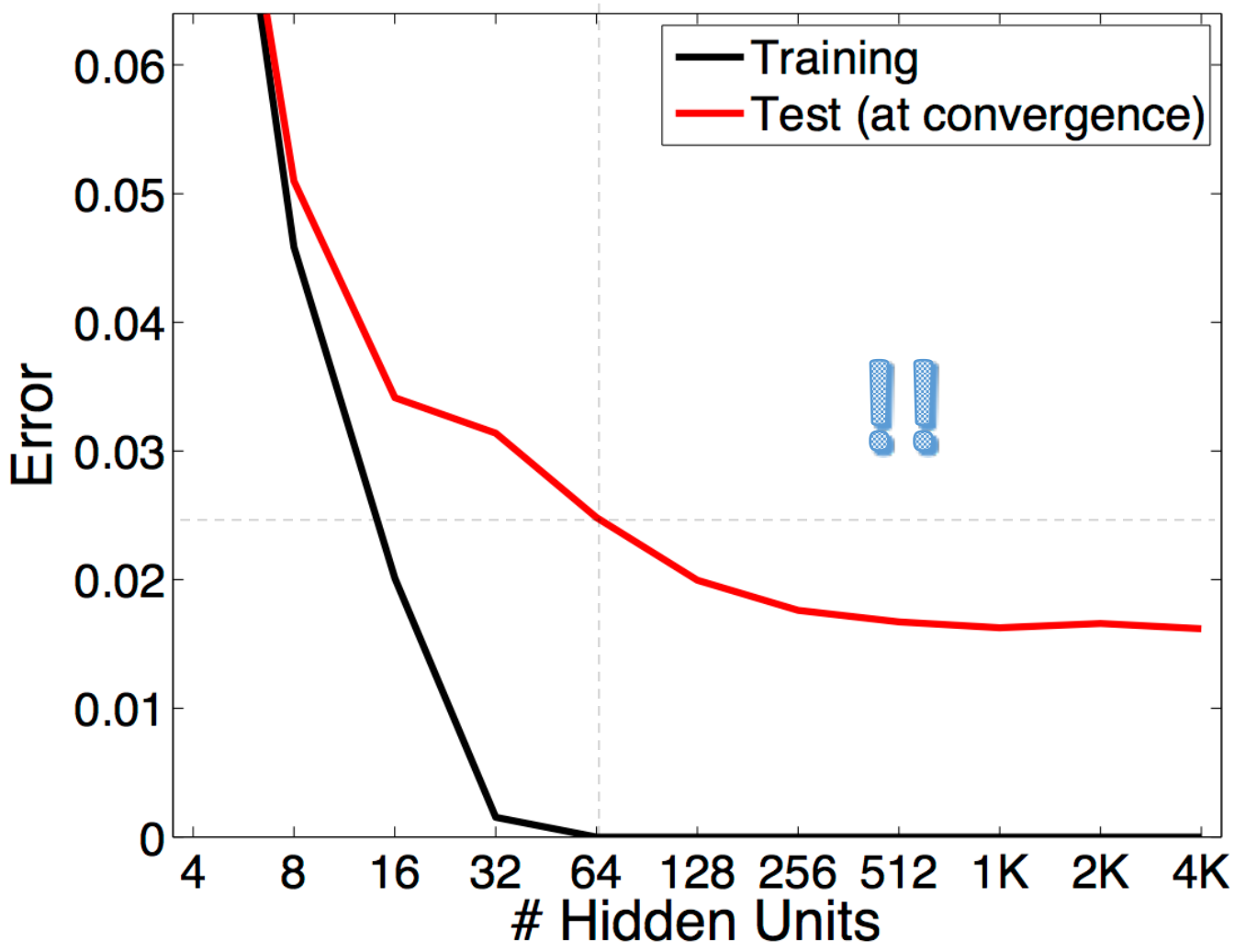
$$w^0 \sim \mathcal{N}(0, I)$$

- Iteration t:

$$w^t = w^{t-1} - 0.01 \nabla_w L(f_w(x), y)$$

We train networks with different number of hidden units H.

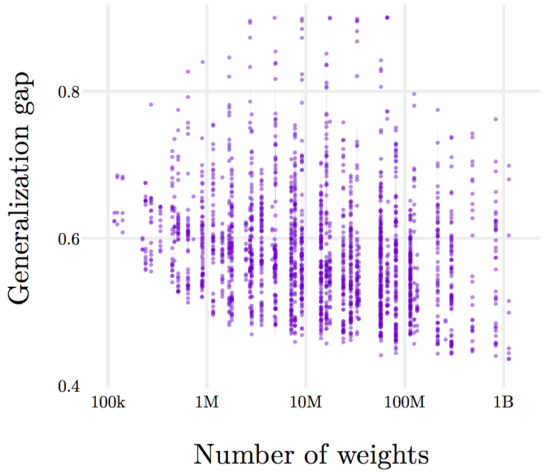
Increasing the Network Size (Number of Hidden Units)



[Neyshabur, Tomioka, Srebro. ICLR'15]

Local Search + Real Data

Similar findings in a comprehensive set of experiments on 2160 networks [Novak et al. ICLR '18]



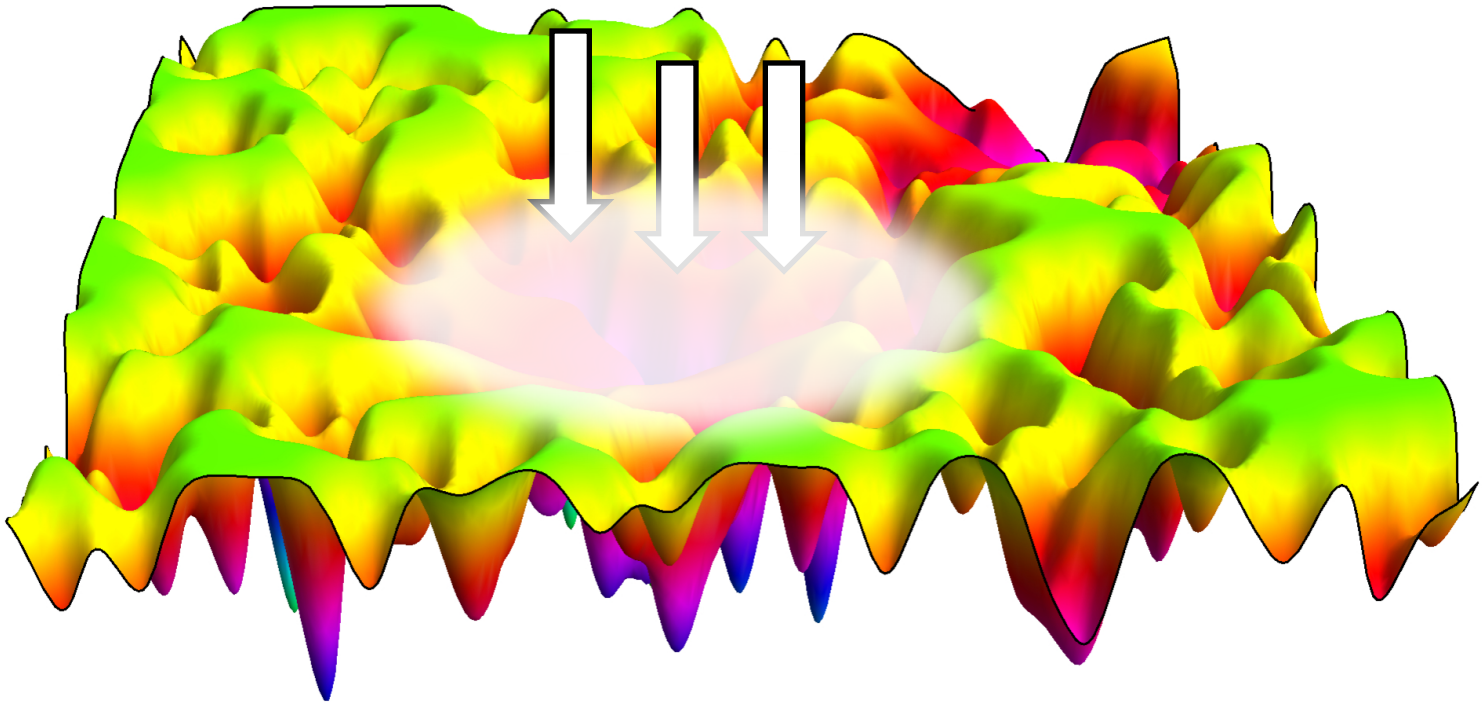
Main empirical observations:

- Large networks can be *optimized* better than small networks
- Large networks *generalize* better than small networks

What is the magic property of the real data that makes local search work?

Novak, Roman, et al. "Sensitivity and Generalization in Neural Networks: an Empirical Study." *ICLR* (2018).

Local Search + Real Data



Energy landscape taken from http://www-wales.ch.cam.ac.uk/

Learning Fully Connected Networks

- Fix architecture (depth and #hidden units)

$$F = \{\, f_w(x) = output\ of\ net\ with\ weights\ w \,\}$$

$$F^* = \{f \in F \mid f \text{ is learned by local search on real data}\}$$

$$test\ err(f) = training\ err(f) + generalization\ error(f)$$

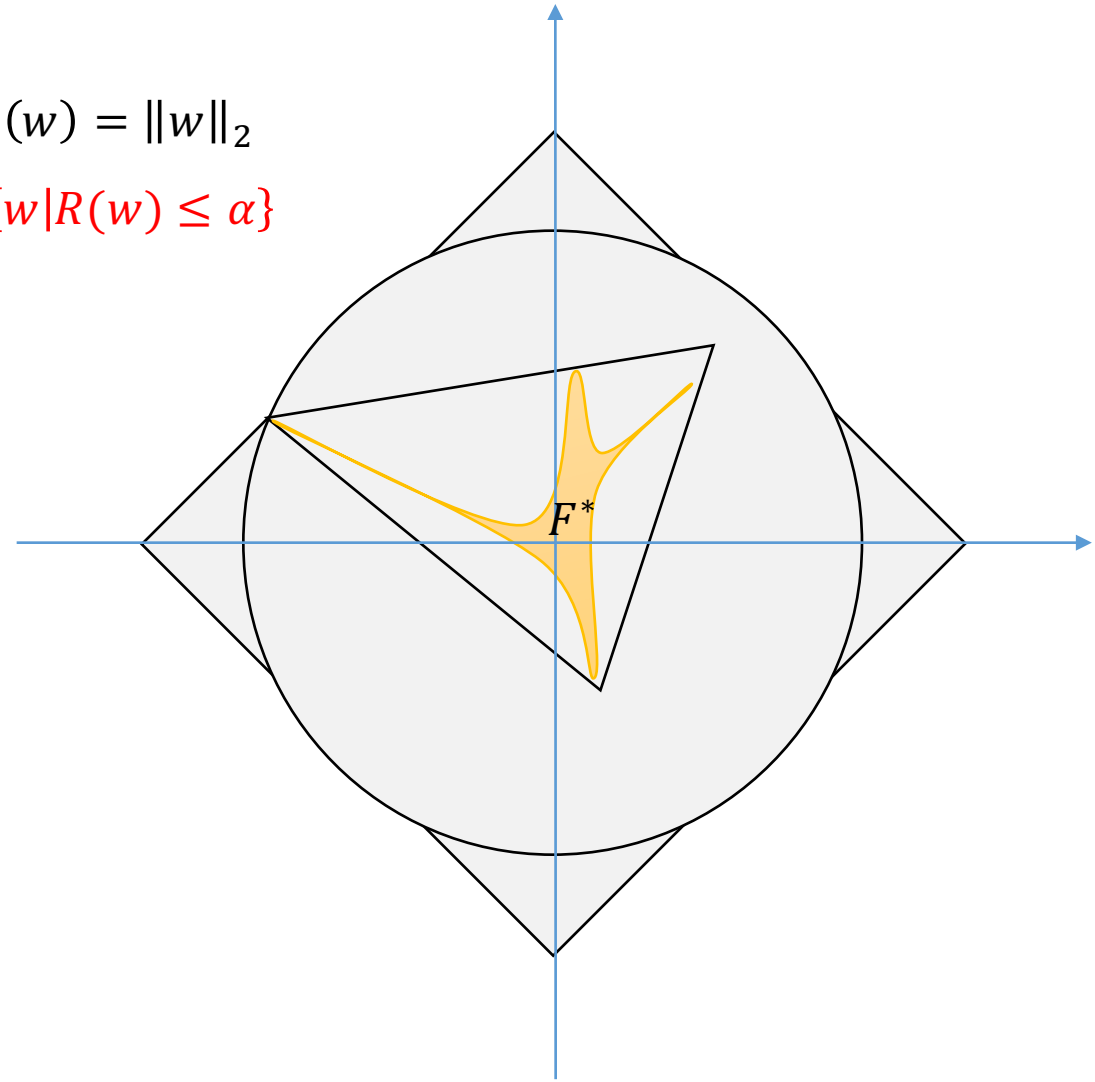
- Expressive Power / Approximation
  - Any continues function with a large enough network
- Computation / Optimization
  - Even if function exactly representable with single hidden layer with $\Theta(\log D)$ units: no poly-time algorithm always works
    [Kearns Valiant 94; Klivans Sherstov 06; Daniely Linial Shalev-Shwartz '14]
- Capacity / Generalization ability / Sample Complexity
  - depth $\times$ number of weights

True Capacity

w: the parameter vector.

$R(w)$: complexity measure, ex. $R(w) = \|w\|_2$

Choose smallest α s.t. $F^* \subset \hat{F} = \{w|R(w) \leq \alpha\}$



Goal

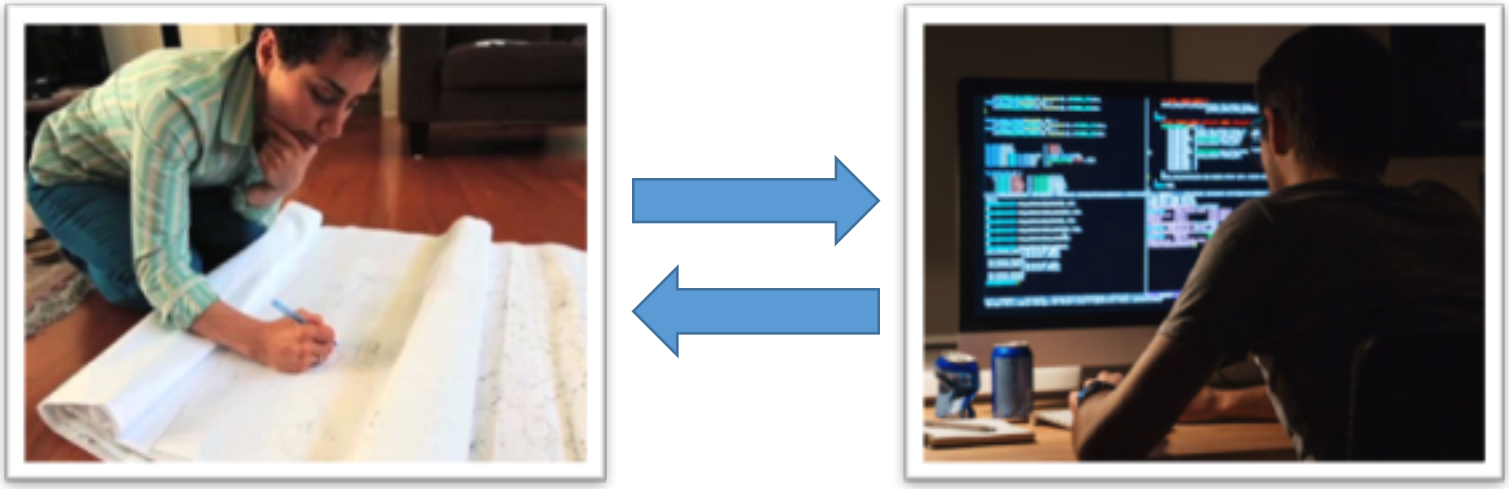
- What is the goal of studying generalization?

- How can studying generalization help us?
  - Designing optimization algorithms
  - Designing architectures

Goal: Find a measure $R(w)$ with the following properties:

- Reasonable generalization bound for the class $H_\alpha = \{f_w | R(w) \leq \alpha\}$
- $R(w)$ correlates with generalization: w_1 has better generalization than w_2 iff $\mu(w_1) \leq \mu(w_2)$

Research on Generalization



- Develop the theory.
- Improve your intuition about what kind of $R(w)$ controls complexity

- Evaluate the generalization bound
- Improve your intuition about what kind of $R(w)$ correlates with test error

Getting to the core issue

$$F^* = \{f \in F \mid f \text{ is learned by local search on real data}\}$$

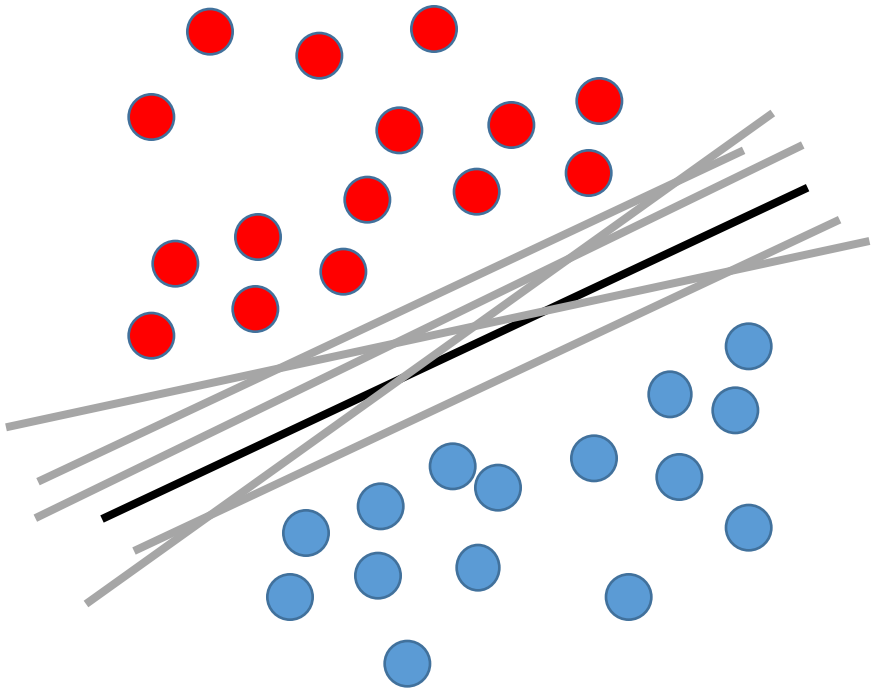
- Let's forget about optimization and focus on real data for now.
- Main observation in neural nets:
  #parameters $\gg$ #data points but they generalize!

How to explain this? Simpler models with similar behavior?
Yes, linear classifiers!

Spam Filtering

- Task: classify emails as spam or non-spam
  - Vocabulary of size 100K words
  - 10^{10} bigrams (seq. of two words).
- Input x: 10^{10} dim. vector that is freq. of bigrams
- Label: 1 for spam, -1 for non-spam
- Model: linear separator $f_w(x) = \langle w, x \rangle$
- In practice, we can generalize with even 10K samples!

Margin-based Generalization



$\|x_i\|_2^2 \leq 1$ $\quad$ $y_i\langle w, x_i\rangle \geq \gamma$ $\quad$ margin: $\frac{\gamma}{\|w\|}$

Rademacher Complexity

- Rademacher complexity of a function class F on the training set $S = \{x_1, \ldots, x_m\}$:

$$\mathfrak{R}_S(F) = \mathbb{E}_{\xi \in \{\pm 1\}^m} \left[\sup_{f \in F} \frac{1}{m} \sum_{i=1}^{m} \xi_i f(x_i) \right]$$

$\mathfrak{R}_S(F)$: how well F can fit random labels on average

For any F, w.p. $1 - \delta$ over the choice of the training set of size m,

$$\forall_{f \in F} \quad \underbrace{L(f)}_{\textit{test err}} \leq \underbrace{\hat{L}_\gamma(f)}_{\textit{training err}} + O\left(\frac{\mathfrak{R}_S(F)}{\gamma} + \sqrt{\frac{\ln(1/\delta)}{m}} \right)$$

Example: Linear Separator

- Let $F = \{\langle w, x\rangle \mid \|w\|_2 \leq \alpha\}$ and let $\{\|x\|_2 \leq 1\}$ be the input domain. Then

$$\Re_S(F) \leq \sqrt{\frac{\alpha^2}{m}}$$

$$L(f) \leq \hat{L}_\gamma(f) + O\left(\sqrt{\frac{\alpha^2}{\gamma^2 m}} + \sqrt{\frac{\ln(1/\delta)}{m}}\right)$$

- Improves over VC-dimension when $\frac{\alpha^2}{\gamma^2} \leq d$

Implicit Regularization of Gradient Descent

- If we regularize the norm of the weights properly, the solution will have maximum margin ☺

- What if we just do gradient descent without regularization?

Gradient descent implicitly regularizes the weights and biases the solution toward the one with maximum margin [Soudry et al. 2017]! ☺

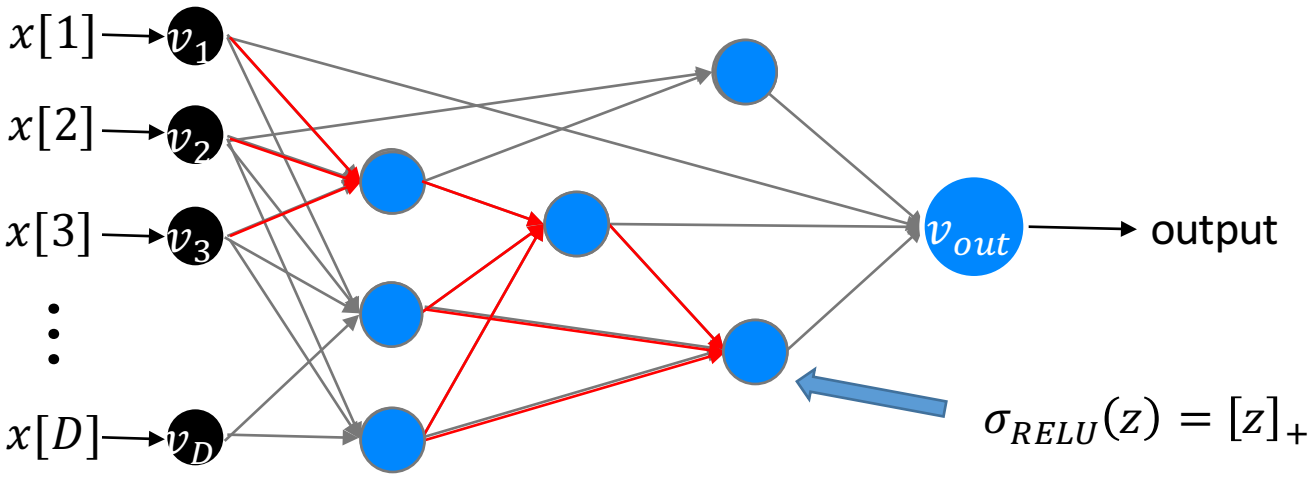
Remember the big picture:

$$F^* = \{f \in F | \ f \text{ is learned by local search on real data}\}$$

Soudry, Daniel, Elad Hoffer, and Nathan Srebro. "The Implicit Bias of Gradient Descent on Separable Data." *arXiv preprint arXiv:1710.10345* (2017).

Group Norm Regularization

$$\mu_{p,q}(w) = \left(\sum_{v \in V} \left(\sum_{(u \to v) \in E} |w(u \to v)|^p \right)^{q/p} \right)^{1/q}$$



[Neyshabur, Tomioka, Srebro. COLT'15]

Group Norm Regularization

$$\mu_{p,q}(w) = \left(\sum_{v \in V} \left(\sum_{(u \to v) \in E} |w(u \to v)|^p \right)^{q/p} \right)^{1/q}$$

$\mathbf{p} = q$
overall ℓ_p regularization

$\mathbf{p} = q = \mathbf{2}$: weight decay
$\mathbf{p} = q = \mathbf{1}$: sum of absolute weights

$$\mu_{p,p}(w) = \left(\sum_{e \in E} |w(e)|^p \right)^{\frac{1}{p}}$$

$q = \infty$
per-unit ℓ_p regularization

$p = \mathbf{2}$: max-norm regularization
$p = \mathbf{1}$: per-unit ℓ_1 regularization

$$\mu_{p,\infty}(w) = \max_{v \in V} \left(\sum_{(u \to v) \in E} |w(u \to v)|^p \right)^{1/p}$$

Sample Complexity

$$\text{capacity} \propto \frac{\mu^{2d}\left(2H^{[1-(1/p+1/q)]_+}/\sqrt{d}\right)^{2d}}{\gamma^2}$$

depth: d (max length directed path)

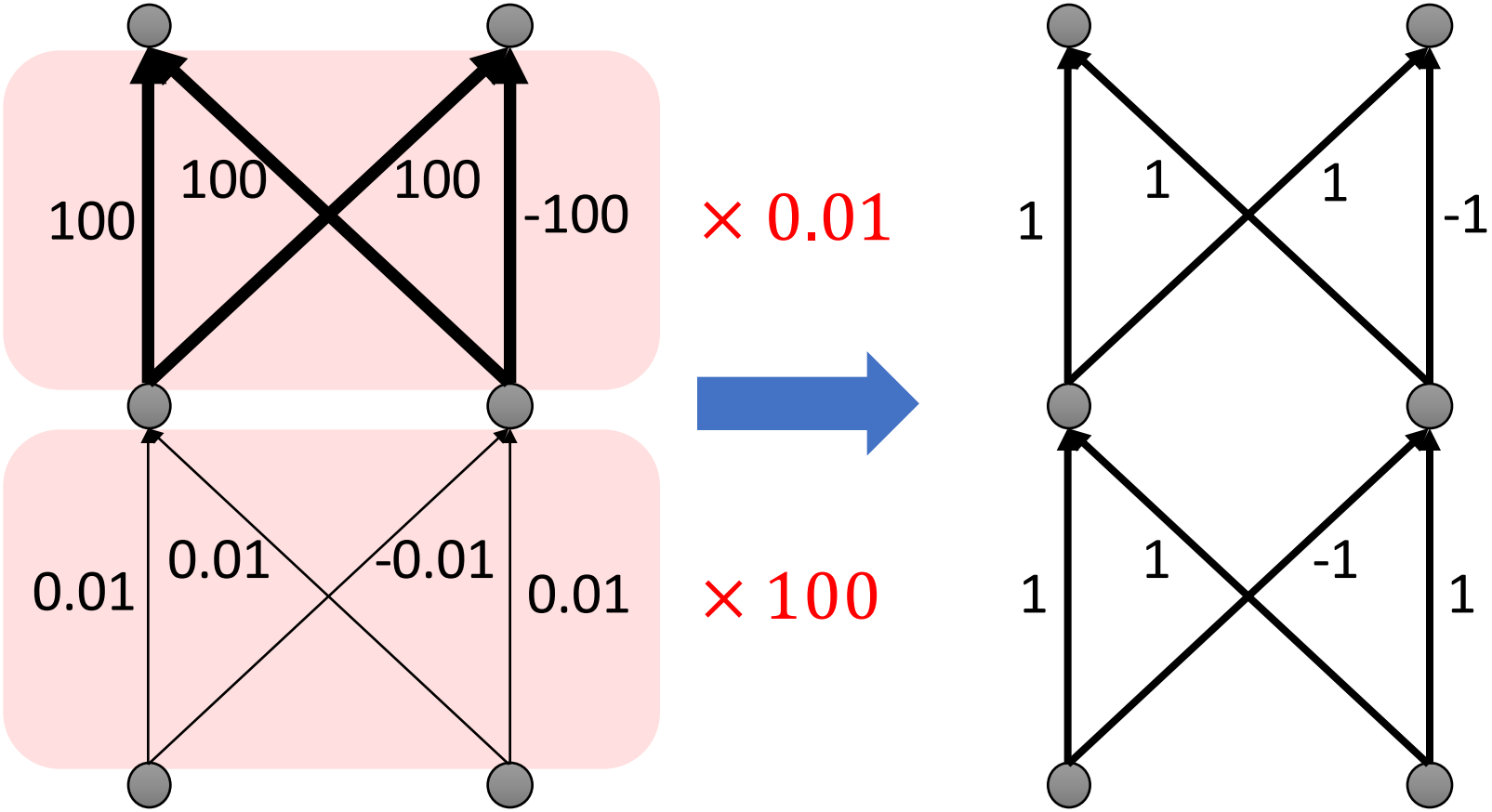
width: H (max in-degree of a vertex)

Generalization guarantees:

- Independent of #params if $\frac{1}{p}+\frac{1}{q} \geq 1$
- Exponential dependence on depth d
- Both H and d dependencies are ***tight***!

[Neyshabur, Tomioka, Srebro. COLT'15]

μ can be *fooled*



$\mu_{2,2} \approx 200$ **How to fix this?** $\mu_{2,2} = 2\sqrt{2}$

Independent of width H?

Whenever $\frac{1}{p} + \frac{1}{q} = 1$ (ℓ_p and ℓ_q are dual of each other):

$$\text{capacity} \propto \frac{\prod_{i=1}^{d} \|W_i\|_{p,q}^2}{\gamma^2}$$

- $p = 2$: Frobenious norm (weight decay)
- $p = 1$: Per-unit ℓ_1 norms [Bartlett, 2002].

Covering Number

- Given a training set S, how many behaviors $V \subset \mathbb{R}^m$ can capture all behaviors $F(S)$ up to margin γ?

For any F, w.p. $1-\delta$ over the choice of the training set of size m,

$$\forall_{f \in F} \quad \underbrace{L(f)}_{\textit{test err}} \leq \underbrace{\hat{L}_\gamma(f)}_{\textit{training err}} + O\left(\sqrt{\frac{\log N_{\gamma,S}(F)}{m}} + \sqrt{\frac{\ln(1/\delta)}{m}}\right)$$

$$\text{capacity} \propto \frac{\prod_{i=1}^{d}\|W_i\|_2^2\left(\sum_{i=1}^{d}\|W_i\|_{1,2}^2/\|W_i\|_2^2\right)}{\gamma^2}.$$ [Bartlett et al, NIPS 2017]

Bartlett, Peter L., Dylan J. Foster, and Matus J. Telgarsky. "Spectrally-normalized margin bounds for neural networks." *NIPS*. 2017.

Experiments on True and Random Labels

Experiments Settings:

- A VGG network trained on a subset of CIFAR10.
- Training on true vs. random labels.
- Optimization by SGD with fixed step-size 0.01 and momentum 0.9

Expect to see two phenomena:

- Norm of a model trained on true labels should be lower.
- Gap between true and random labels should increase significantly.

[Neyshabur, Bhojanapalli, Srebro, NIPS'17]

Norms and Margin

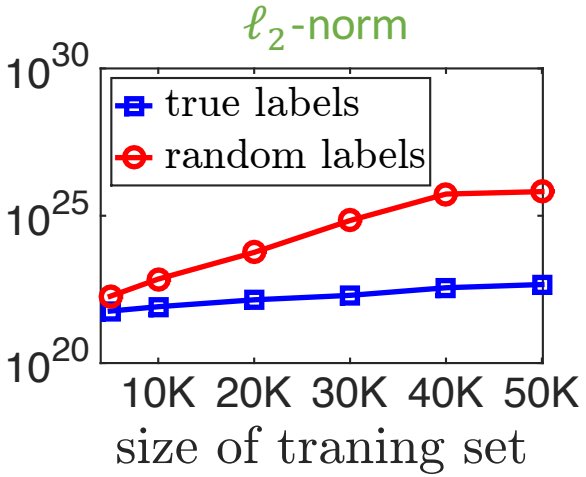
margin = score of the correct label – maximum score of other labels

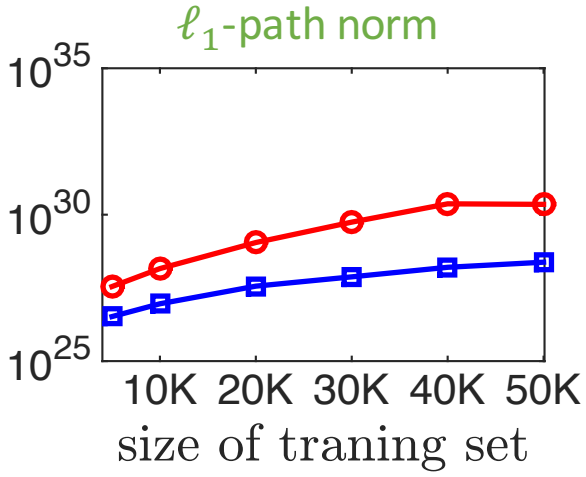
γ= largest value such that only a small portion of the data points violate the margin

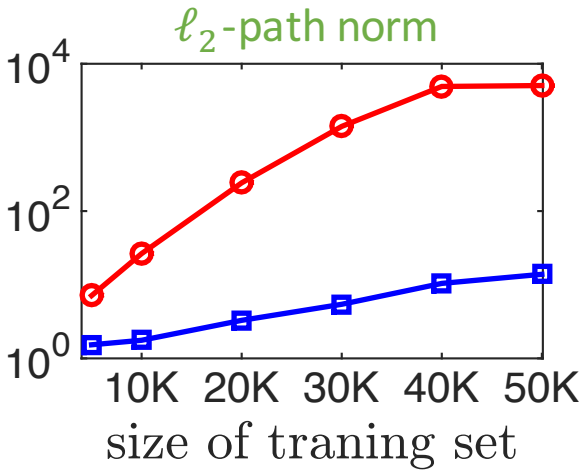
Norm-based measures:

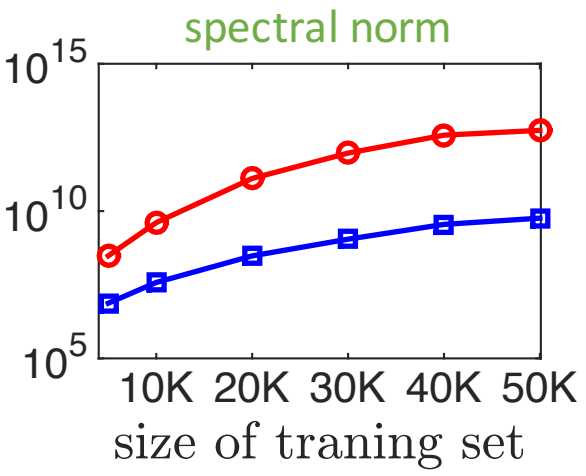
- ℓ_2-norm with capacity $\propto \frac{\prod_{i=1}^{d} \|W_i\|_F^2}{\gamma^2}$
- ℓ_1-path norm with capacity $\propto \frac{\prod_{i=1}^{d} \|W_i\|_{1,\infty}^2}{\gamma^2}$
- ℓ_2-path norm with capacity $\propto h^d \frac{\prod_{i=1}^{d} \|W_i\|_{2,\infty}^2}{\gamma^2}$
- spectral norm with capacity $\propto \frac{\prod_{i=1}^{d} \|W_i\|_2^2 \left(\sum_{i=1}^{d} \|W_i\|_{1,2}^2 / \|W_i\|_2^2\right)}{\gamma^2}$.

Experiments on True and Random Labels





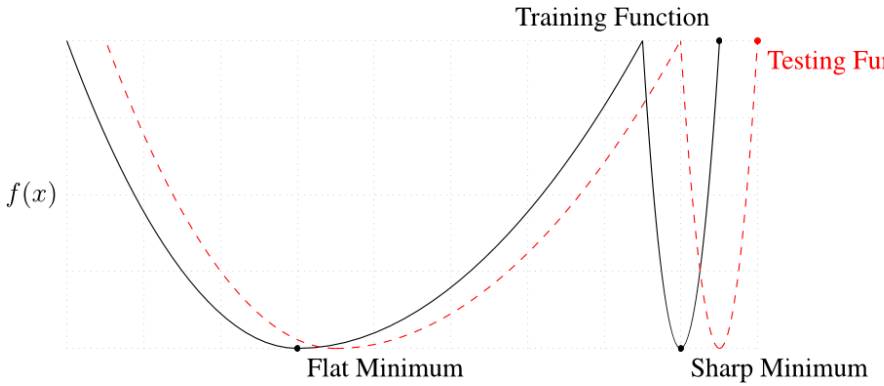


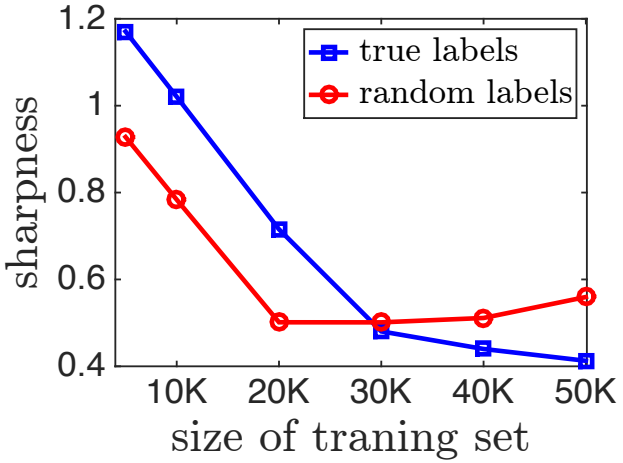


[Neyshabur, Bhojanapalli, Srebro, NIPS'17]

Sharpness

$$sharpness(\alpha) = \max_{\|v\| \leq \alpha} L(w+v) - L(w)$$ [Keskar et al.17]





PAC-Bayesian

- Given a prior P that is independent of data, provides generalization guarantee for randomized predictor drawn from a distribution Q [McAllester 98].

- With probability $1-\delta$ over training data:

$$\mathbb{E}_\nu[L(w+\nu)] \le \mathbb{E}_\nu\left[\hat{L}(w+\nu)\right] + 4\sqrt{\frac{1}{m}\left(KL(w+\nu||P) + \ln\frac{2m}{\delta}\right)}$$

$$\mathbb{E}_\nu[L(w+\nu)] \le \hat{L}(w) + \underbrace{\mathbb{E}_\nu\left[\hat{L}(w+\nu)\right] - \hat{L}(w)}_{\text{expected sharpness}} + 4\sqrt{\frac{1}{m}\underbrace{\left(KL(w+\nu||P) + \ln\frac{2m}{\delta}\right)}}$$

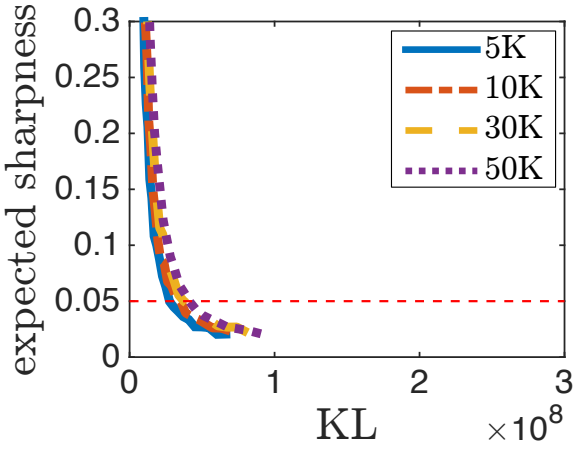
$$\frac{\|w\|_2^2}{2\sigma^2} \text{ if } \begin{cases} P = N(0,\sigma^2) \\ \nu \sim N(0,\sigma^2) \end{cases}$$

PAC-Bayes bounds are numerically low [Dziugaite and Roy, UAI 2017].

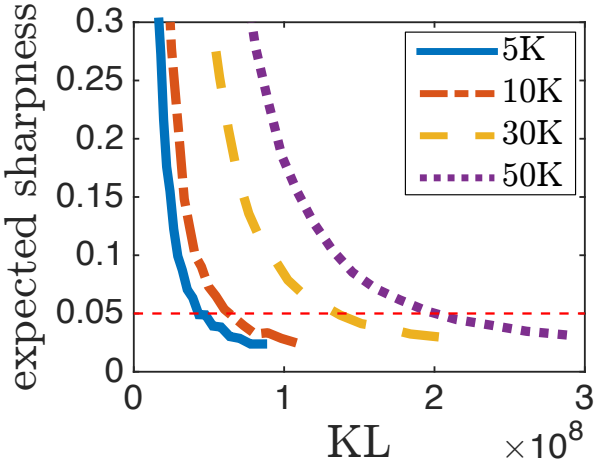
Experiments on True and Random Labels

$$\mathbb{E}_v[L(w+v)] \leq \hat{L}(w) + \underbrace{\mathbb{E}_v\left[\hat{L}(w+v)\right] - \hat{L}(w)}_{\text{expected sharpness}} + 4\sqrt{\frac{1}{m}\underbrace{\left(KL(w+v||P) + \ln\frac{2m}{\delta}\right)}_{\text{KL-term}}}$$

true labels

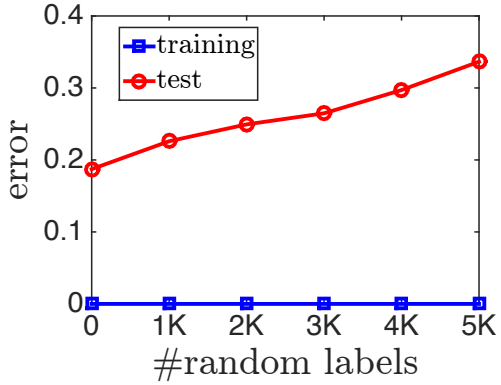


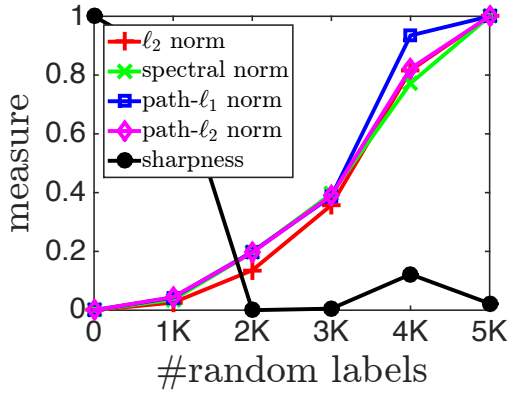
random labels

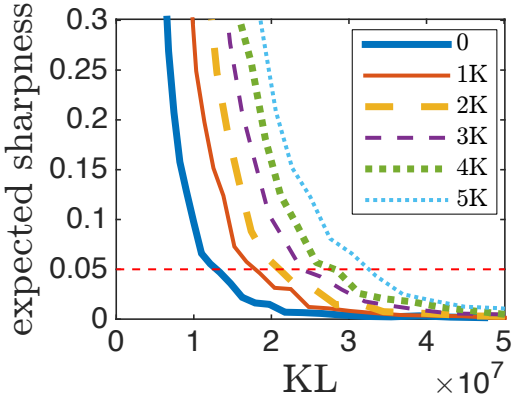


Experiments on different global minima

- Can these measures indicate which global minima generalizes better?
- Optimization on a loss that includes training set and confusion set

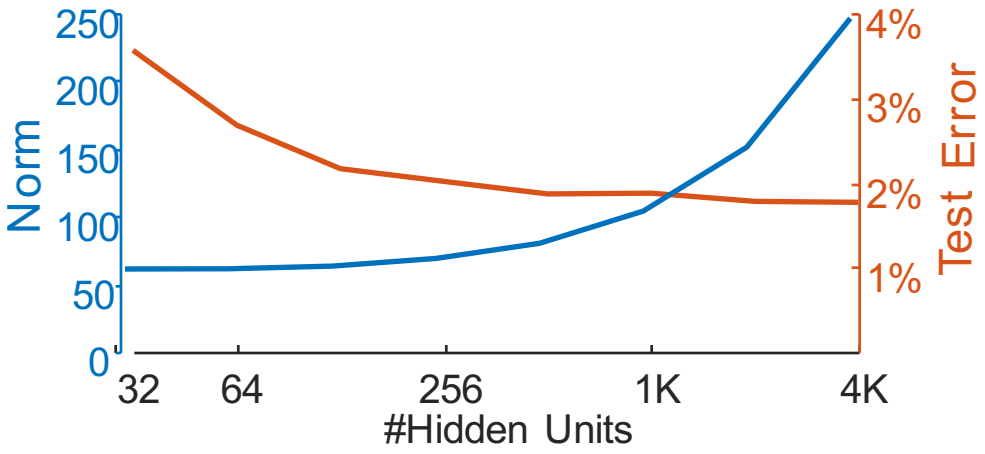


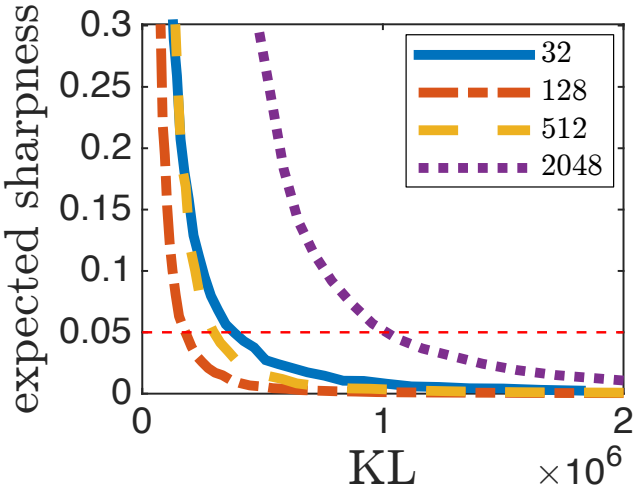




Experiments with varying number of hidden units

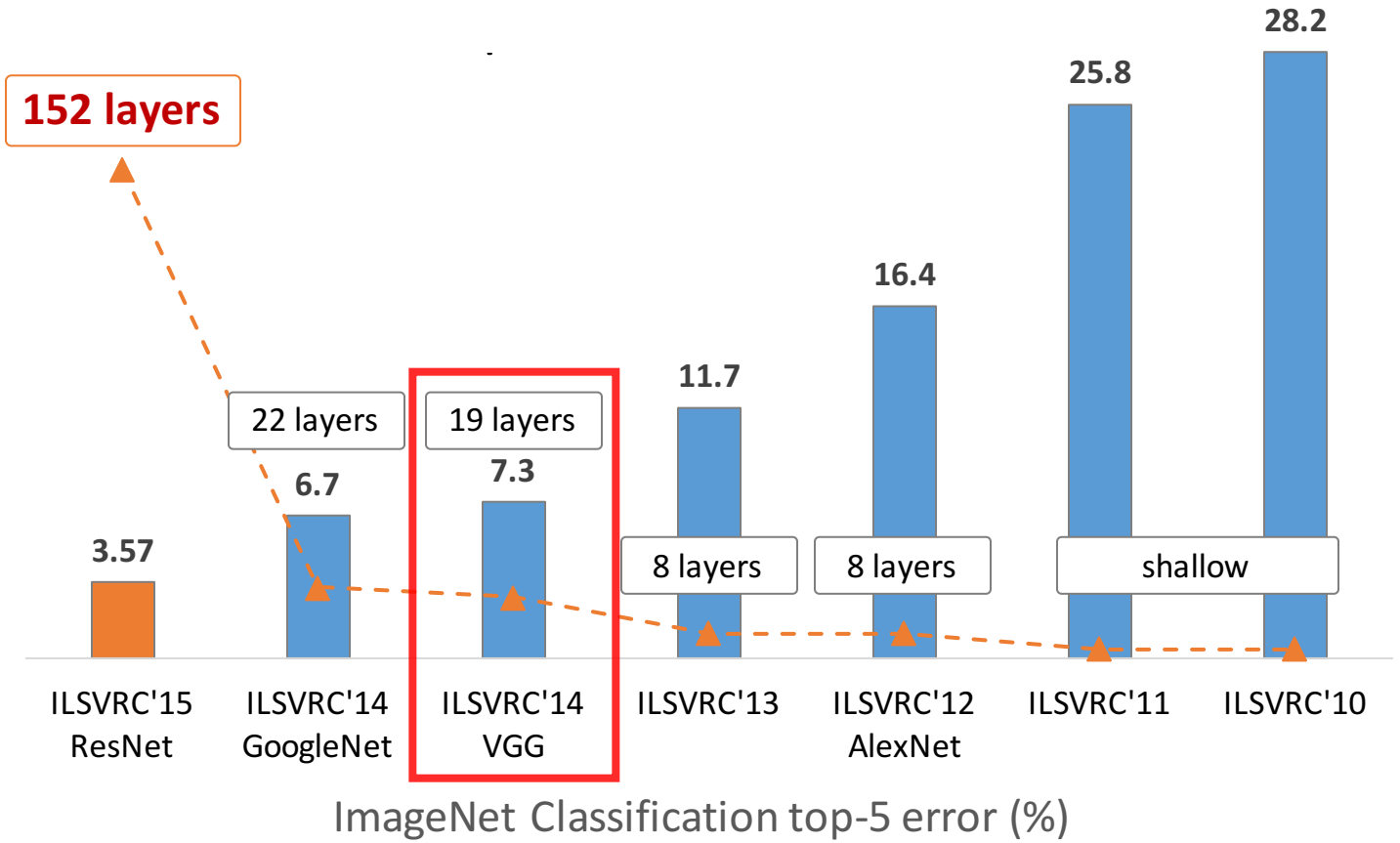
Going back to the first experiment…





How about depth?

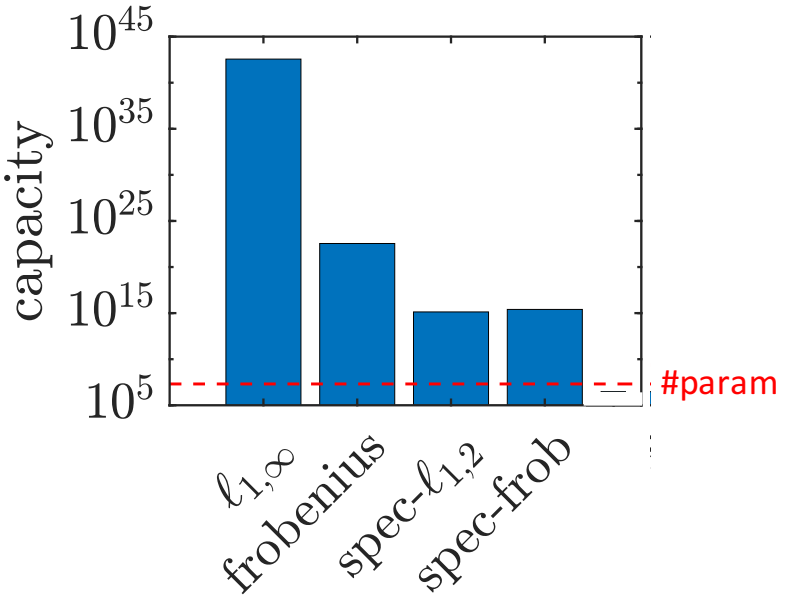
Revolution of Depth



Slide taken from an ICML 2016 tutorial on "Deep Residual Networks" by Kaiming He.

Norm-based Generalization Bounds

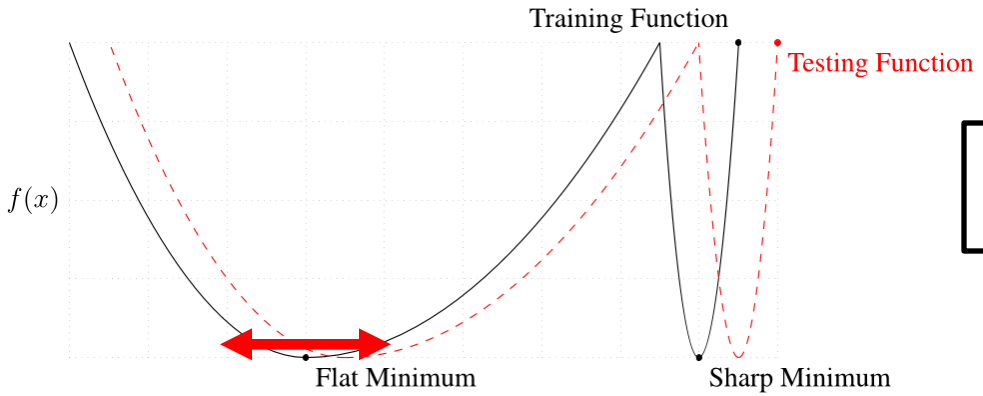
- $\ell_{1,\infty}$: Bartlett et al. (2002)
- Frobenius: Neyshabur et al. (2015)
- spec-$\ell_{1,2}$: Bartlett et al. (2017)
- spec-frob: Neyshabur et al. (2017)

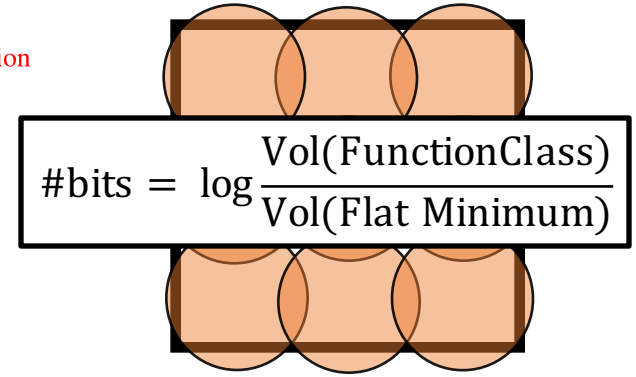


Can we use properties of networks trained on real data to get a bound that is better than VC-dim and correlates with generalization?

Qualitative Explanations

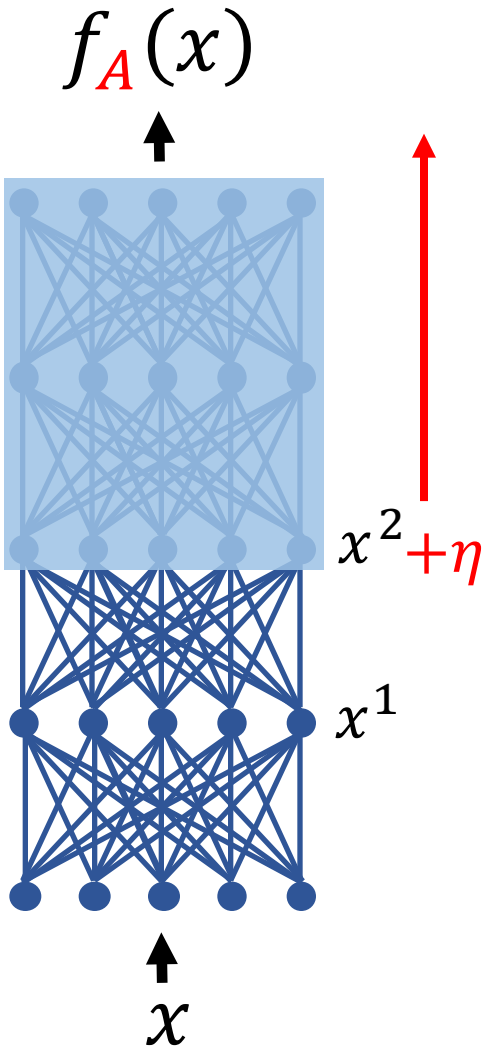
- Flat Minima:
  - Hochreiter and Schmidhuber (1997), Hinton and Van Camp (1993)
  - Recently Keskar et al. (2016)

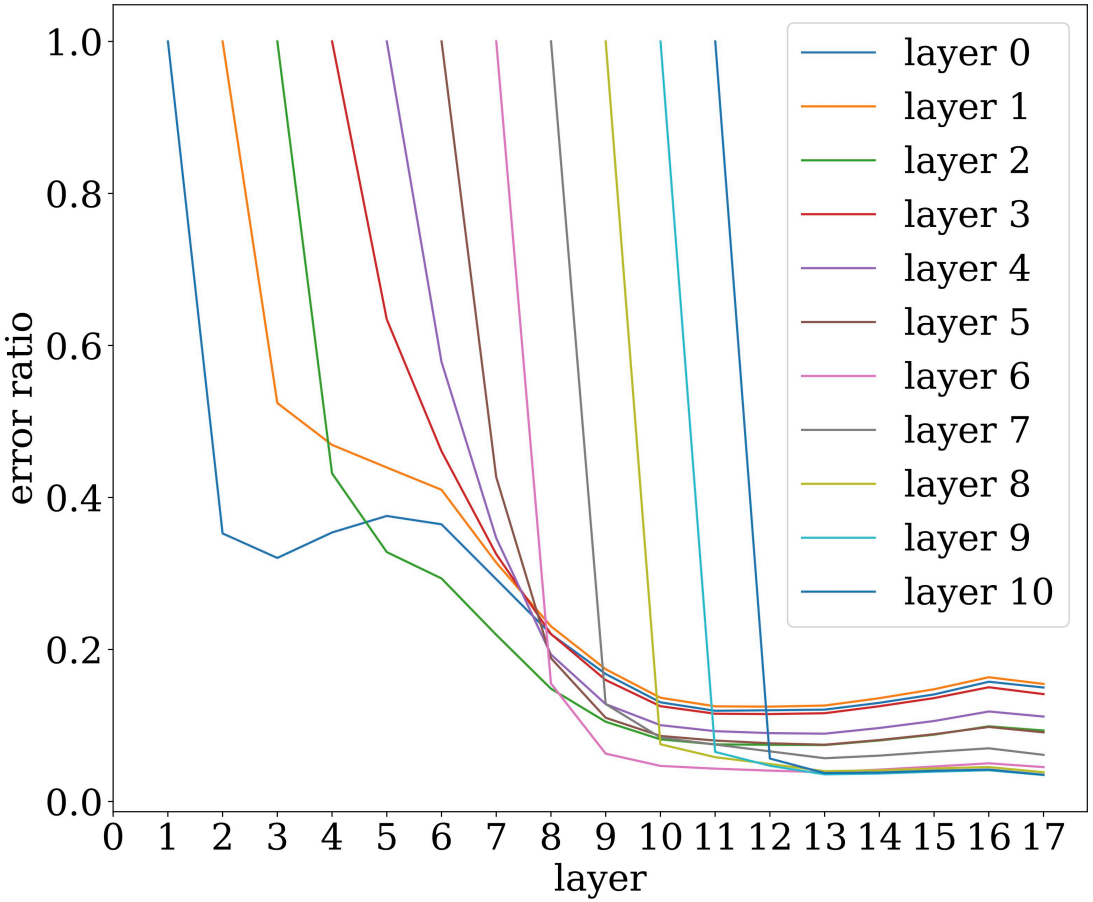




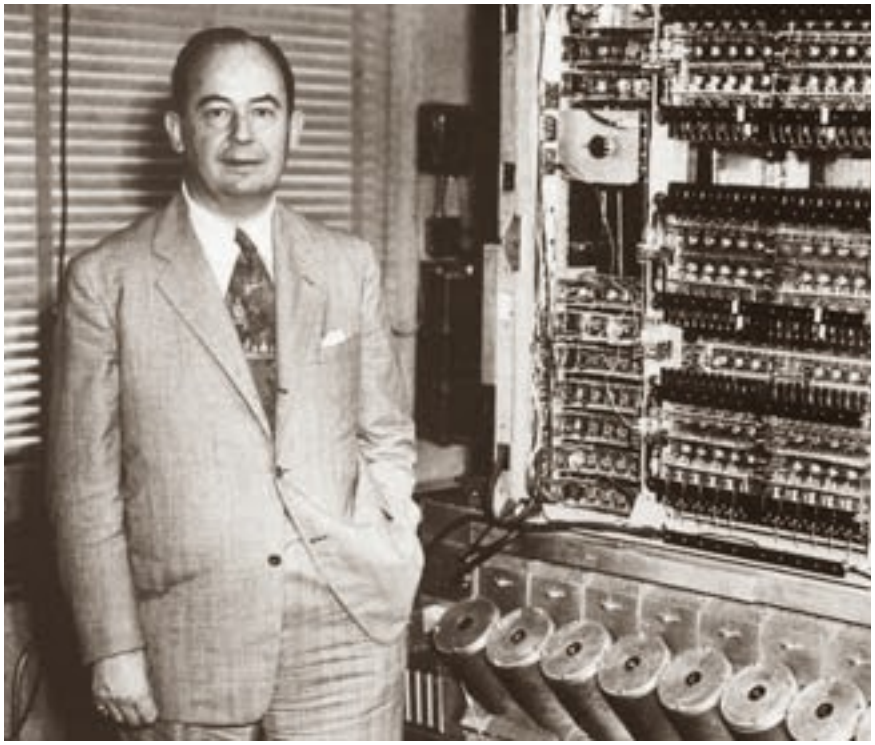
- Noise stability (noise added to parameters):
  - Langford and Caruana (2001), McAllester (1998)
  - Recently Dziugaite and Roy (2017), Neyshabur et al. (2017)
- Noise stability (noise added to layer inputs):
  - Morcos et al. (2018)
  - Connection to dropout and and batch normalization

Noise stability





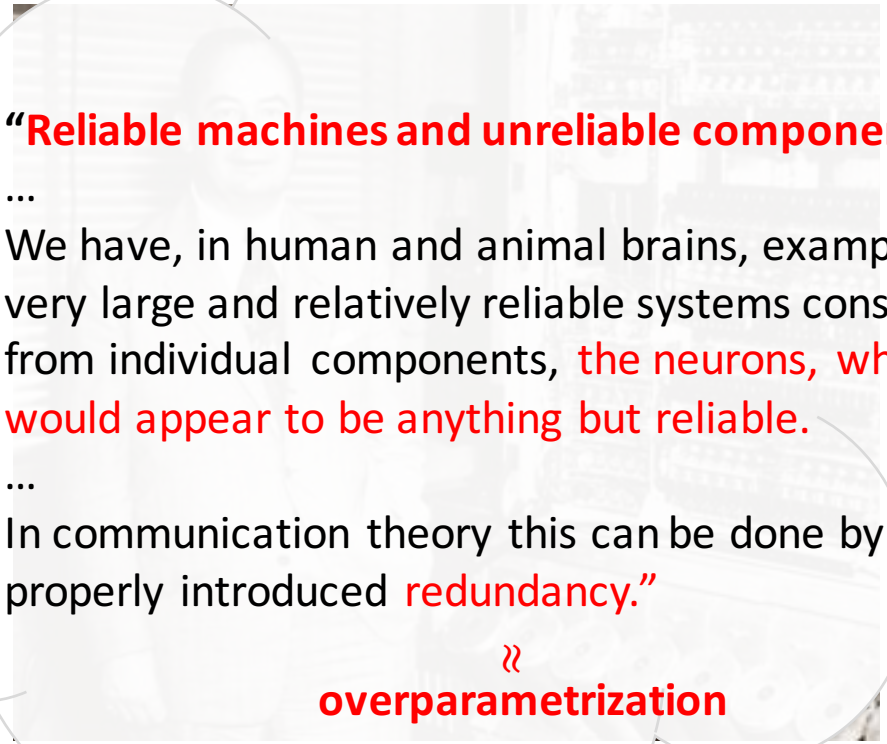
Reliable machines built from unreliable components



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

Shannon, C. E. (1958).
Von Neumann's contributions to automata theory.

Reliable machines built from unreliable components



"Reliable machines and unreliable components

…

We have, in human and animal brains, examples of very large and relatively reliable systems constructed from individual components, the neurons, which would appear to be anything but reliable.

…

In communication theory this can be done by properly introduced redundancy."

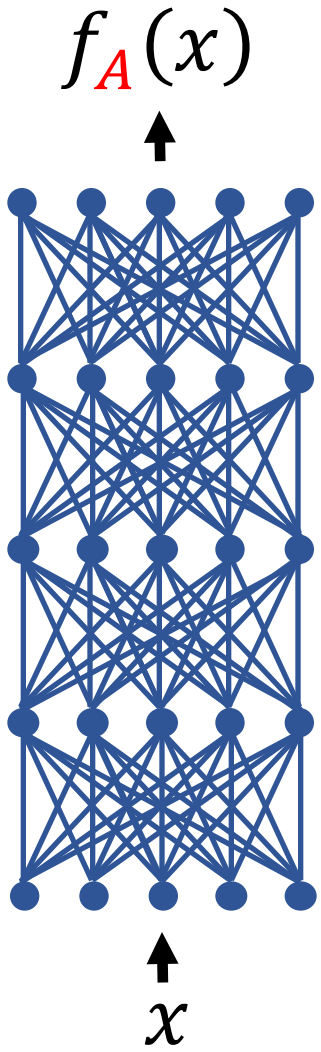
≈

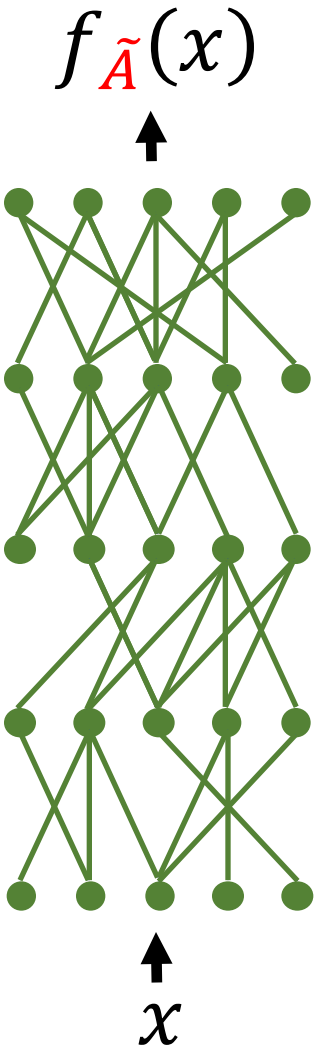
overparametrization

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

Shannon, C. E. (1958).
Von Neumann's contributions to automata theory.

Main idea: compression

compress 

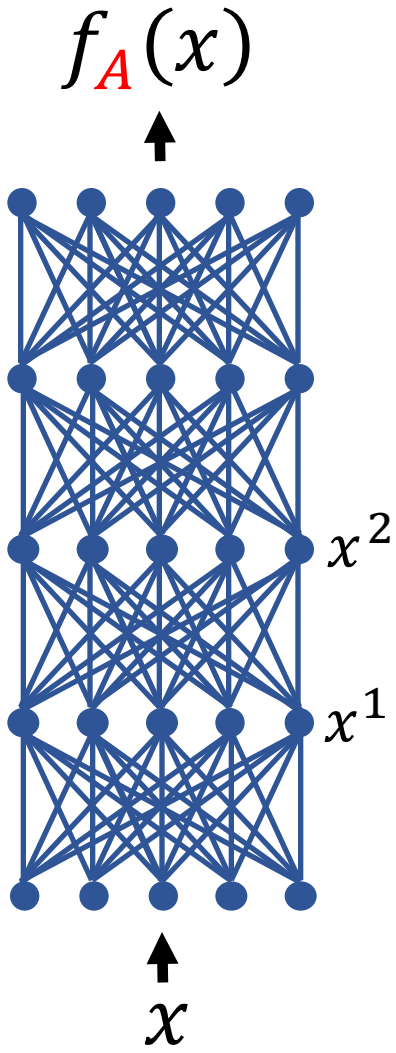
to 

s.t. $|f_A(x) - f_{\tilde{A}}(x)| \leq \epsilon$

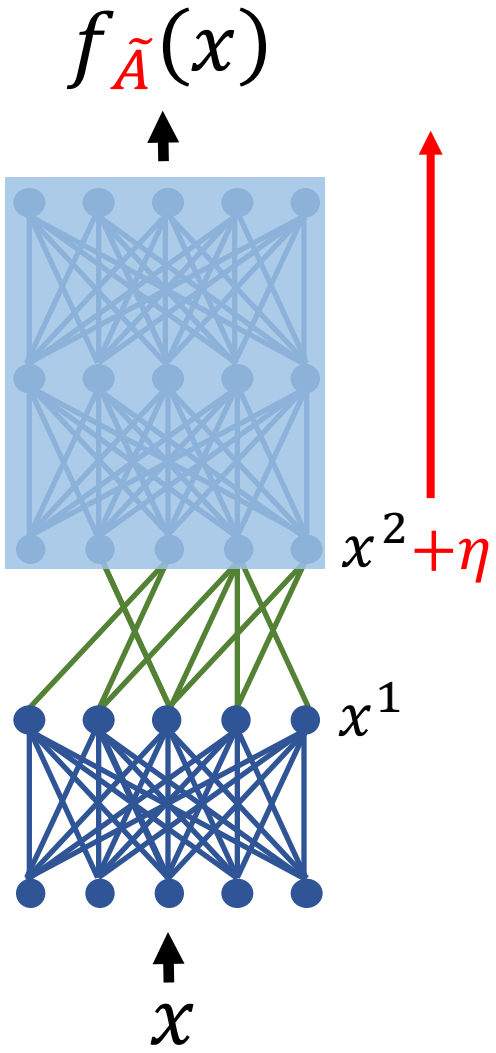
Now, we can simply count the parameters!

Main idea: compression

compress



to



How does the noise propagate?

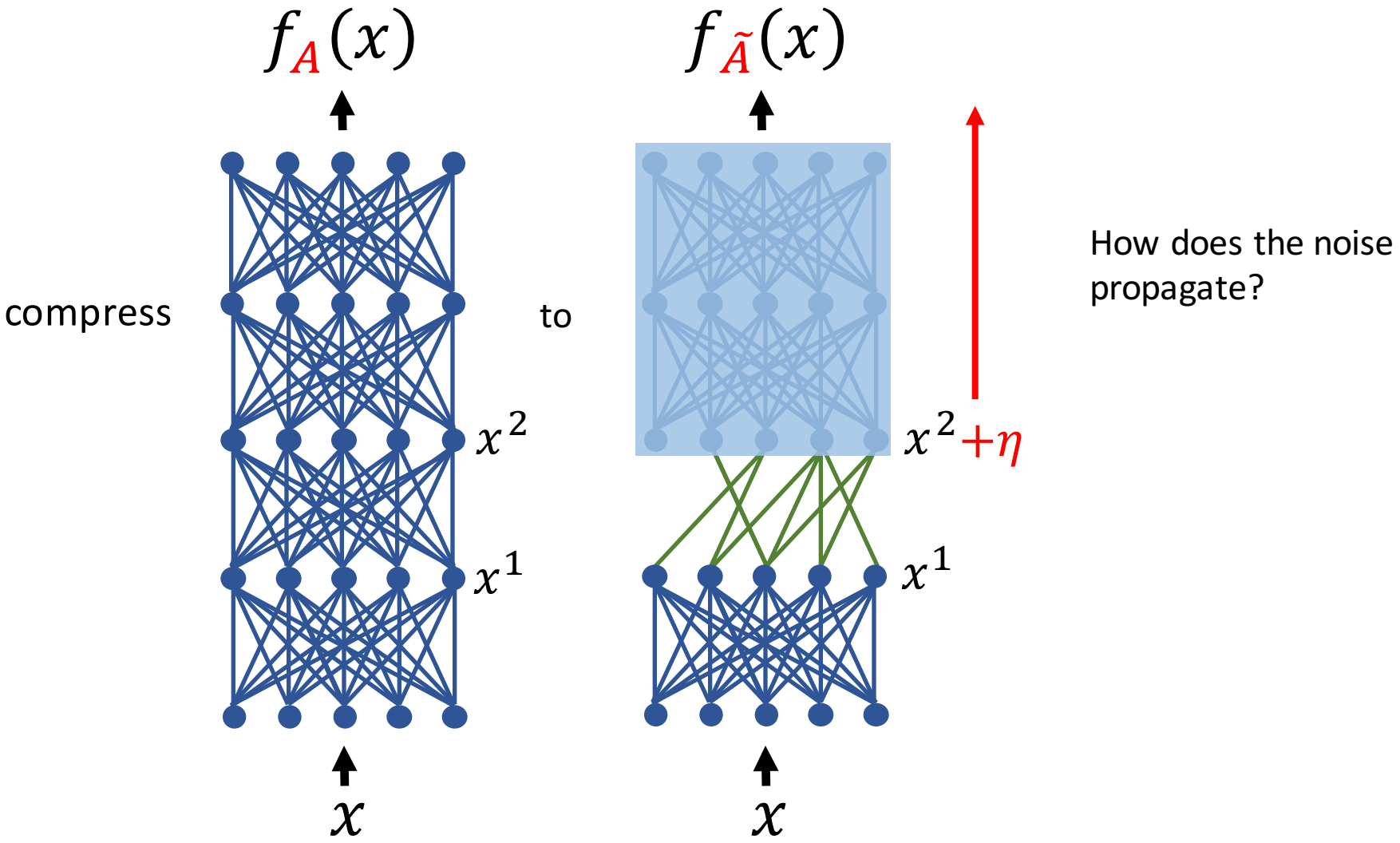
Compression Algorithm

- *Generate* k *random sign matrices* $M_1, \ldots, M_k$

- Return $\hat{A} = \frac{1}{k} \sum_{t=1}^{k} \langle A, M_t \rangle M_t$

parameters independent of data

Main idea: compression



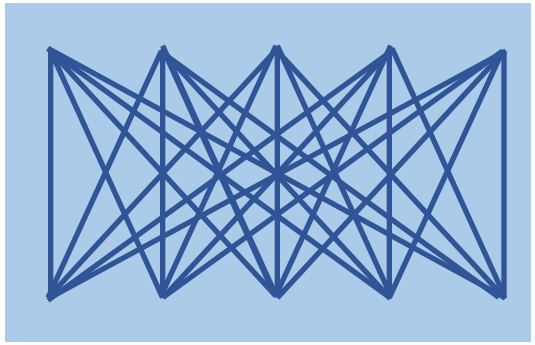
How does the noise propagate?

Identifying properties that allow noise stability

- Layer cushion
- Interlayer cushion
- Activation Contraction
- Interlayer smoothness

Layer Cushion

$f(x)+?$



$x + \eta$

$$f(x) = Ax$$

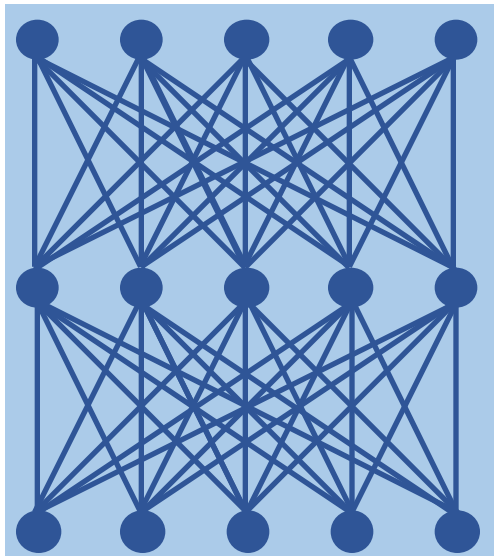
Layer cushion: the relative change in the output of a layer for Gaussian noise added to the input

$$\frac{\|Ax\|}{\|A\|_F \|x\|}$$

Interlayer Cushion

$f(x)+?$

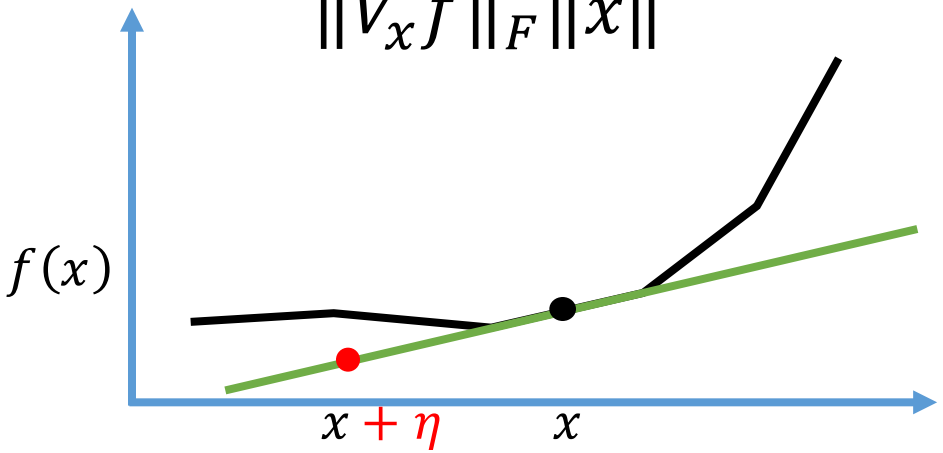
$$f(x) = \nabla_x f . x$$



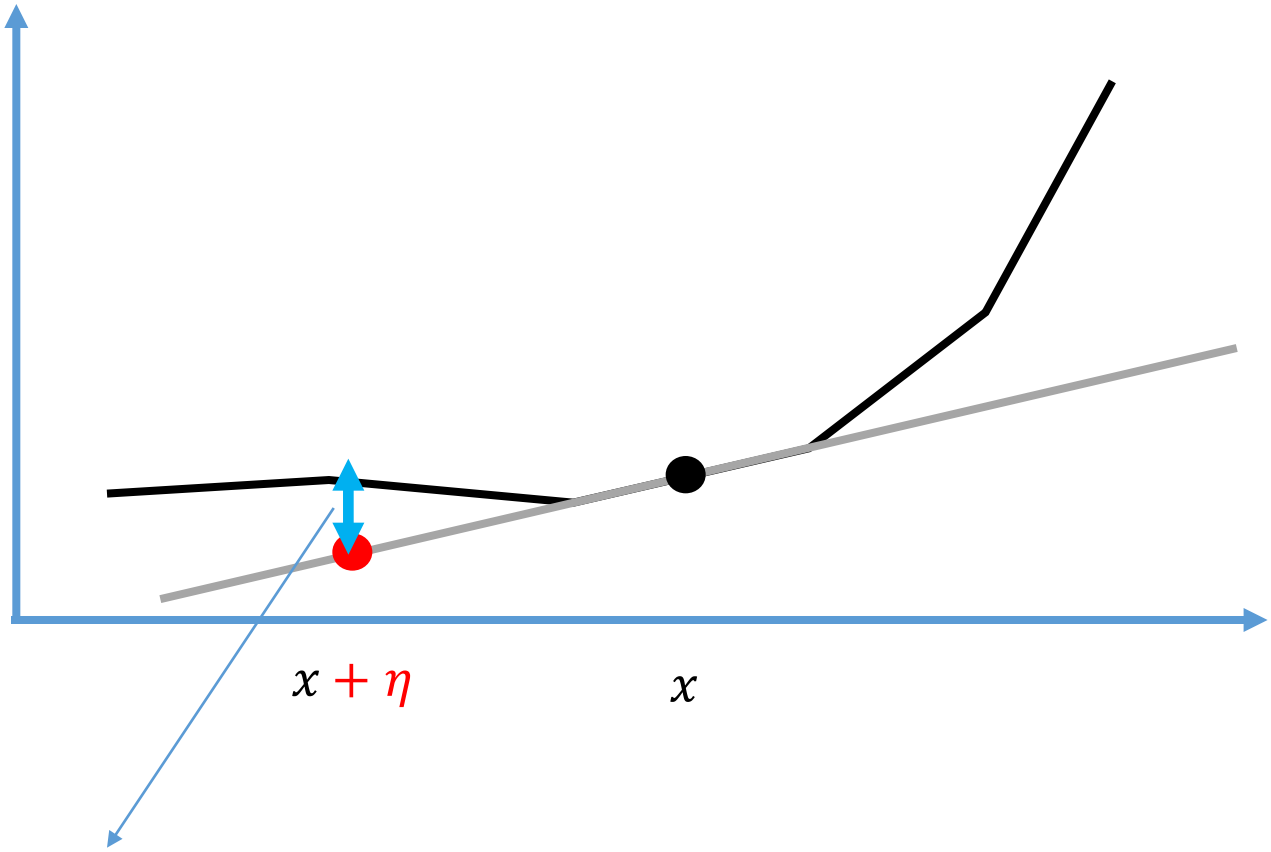
$x + \eta$

Interlayer cushion: the relative change in the output of linearized multiple layers for Gaussian noise added to the input

$$\frac{\|\nabla_x f . x\|}{\|\nabla_x f\|_F \|x\|}$$

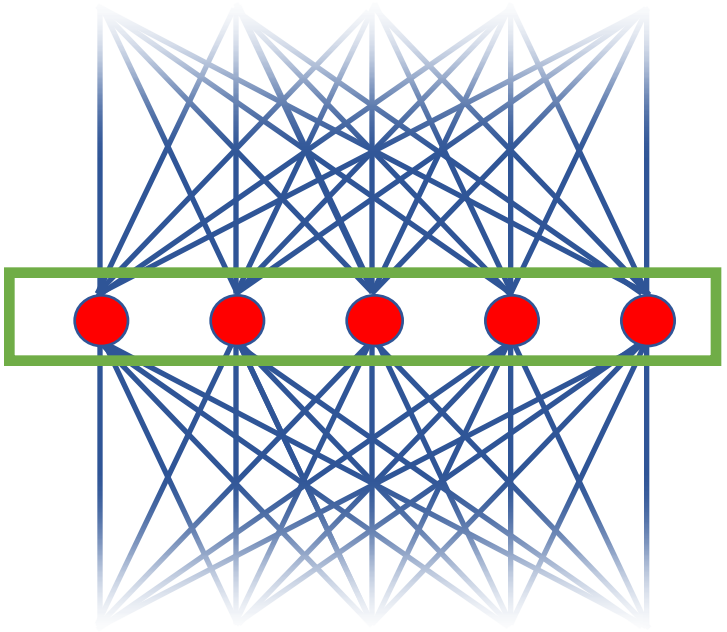


Interlayer Smoothness



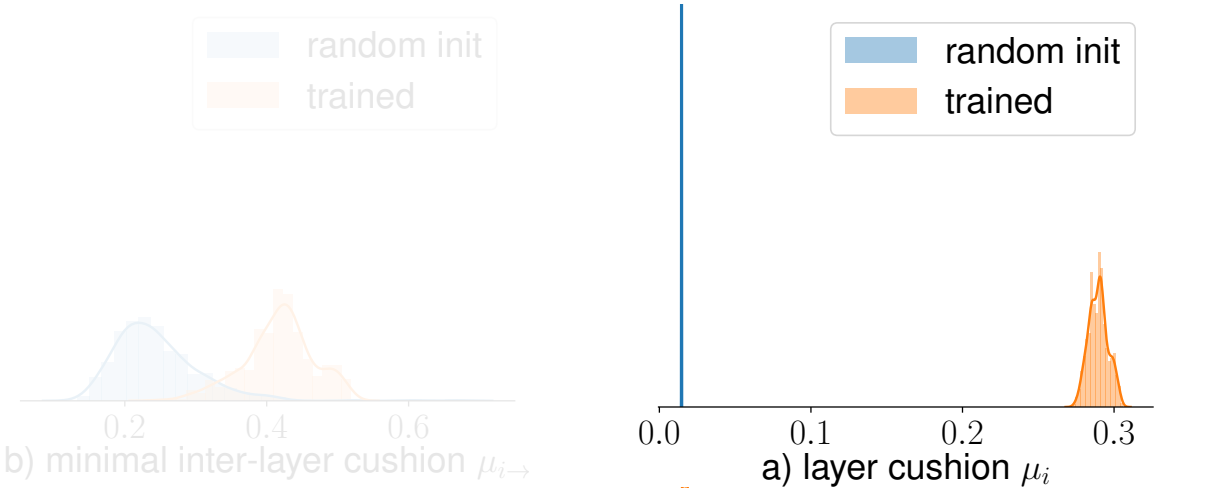
This distance is small and depends on the magnitude of noise η

Activation Contraction



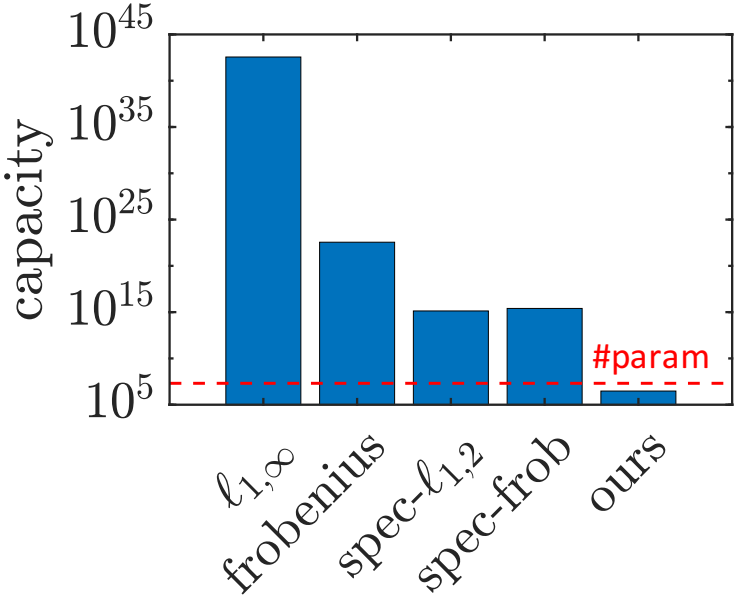
$$\frac{\|z\|_2}{\|[z]_+\|_2} \leq c$$

Empirical investigation of properties

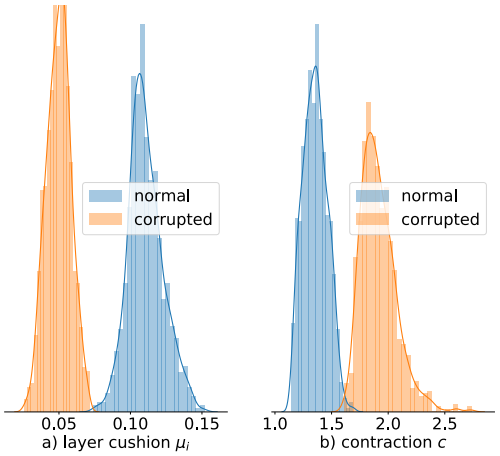


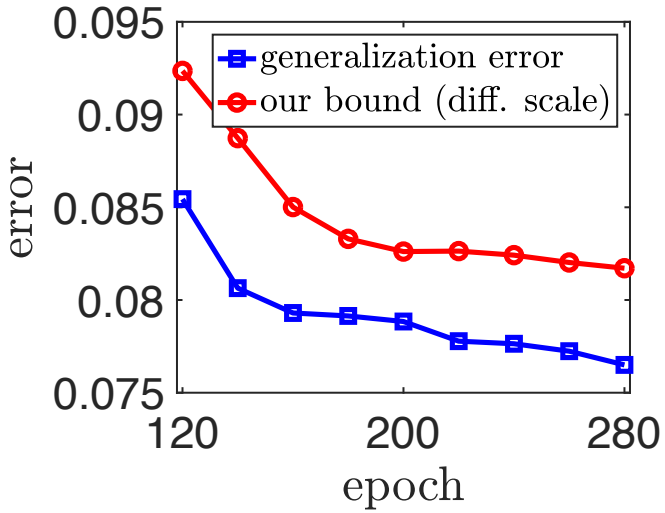
The Generalization Bound

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



Correlation to Generalization





Conclusion

- The role of real data and local search
- Challenges of generalization theory
- Generalization via compression
- Identified properties of networks that allow noise stability
- Stronger Generalization bound that is empirically correlated with the generalization error.

Group Norm Regularization

$$\mu_{p,q}(w) = \left(\sum_{v \in V} \left(\sum_{(u \to v) \in E} |w(u \to v)|^p \right)^{q/p} \right)^{1/q}$$

$\mathbf{p} = q$
overall ℓ_p regularization

$\mathbf{p} = q = \mathbf{2}$: weight decay
$\mathbf{p} = q = \mathbf{1}$: sum of absolute weights

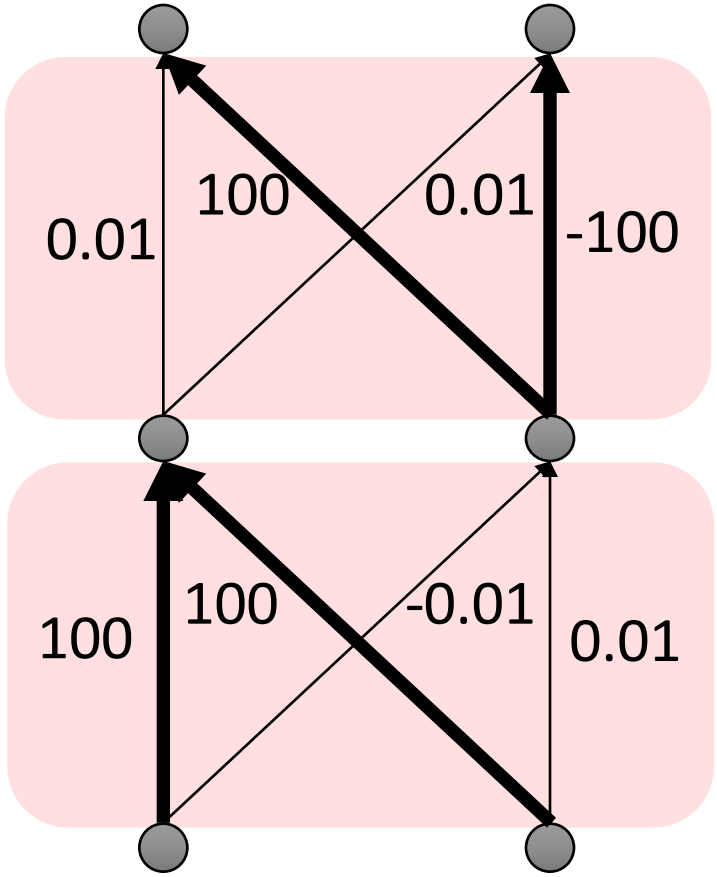
$$\mu_{p,p}(w) = \left(\sum_{e \in E} |w(e)|^p \right)^{\frac{1}{p}}$$

$q = \infty$
per-unit ℓ_p regularization

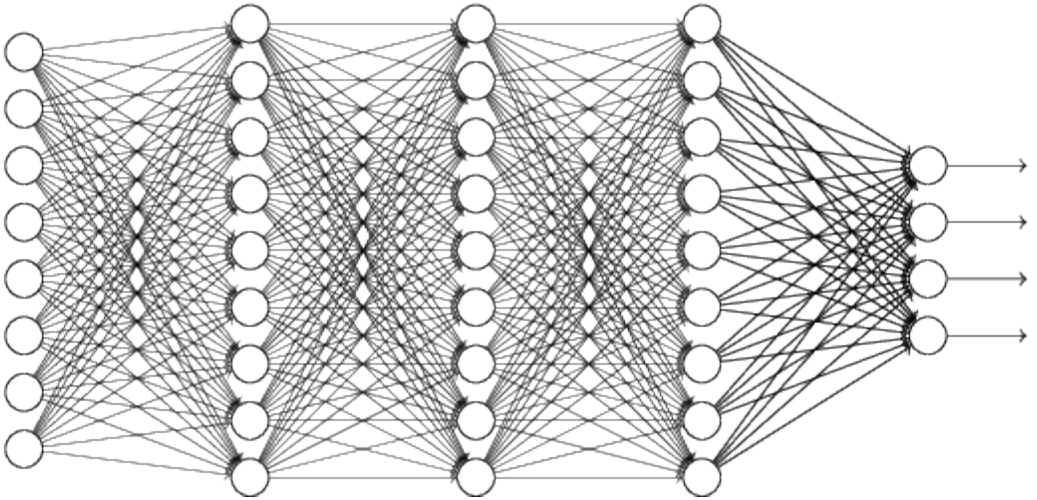
$p = \mathbf{2}$: max-norm regularization
$p = \mathbf{1}$: per-unit ℓ_1 regularization

$$\mu_{p,\infty}(w) = \max_{v \in V} \left(\sum_{(u \to v) \in E} |w(u \to v)|^p \right)^{1/p}$$

μ can be *fooled*



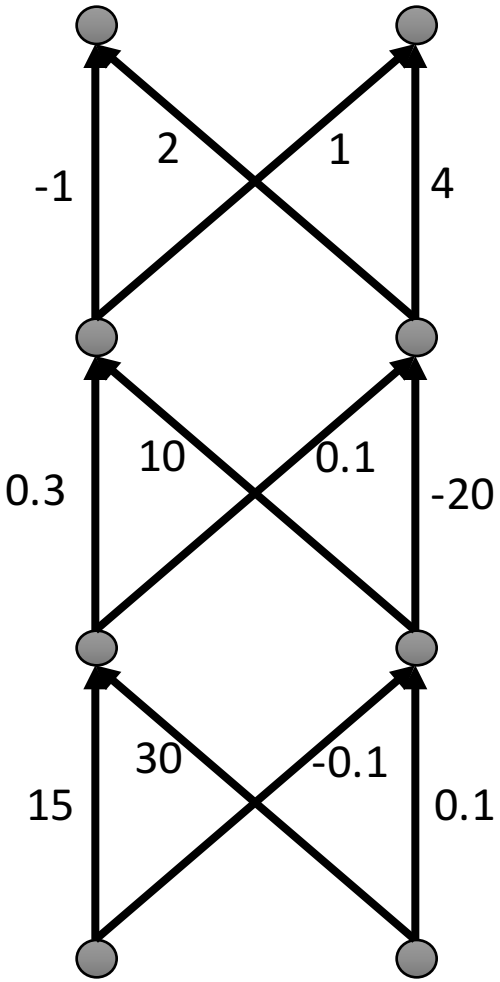
In a deep network, imbalances could be very complicated!



Better Geometry

- Can we devise a better geometry?
- More directly dependent on the **functions computed by the network**, not the vector of weights
- **Invariant to rescaling / reparamatrization**
- Captures a **natural notion of complexity** for deep networks

The Path-Norm



$$\boldsymbol{\pi(w)} = (\ -8\ ,\ 8\ ,\ 4\ ,\ -4\ ,\ -1\ ,\ 1\ ,\ldots)$$

$$\phi_p(w) = \|\pi(w)\|_p = \left(\sum_{\text{path}} \prod_{e\in\text{path}} |w(e)|^p\right)^{1/p}$$

- ✓ $\phi_p(W)$ is invariant to "weight balancing" (scaling down incoming weights and scaling up outgoing weights)
- ✓ Path-Regularization is the minimum per-unit regularization!

$$\phi_p(w) = \min_{f_W = f} \mu^d_{p,\infty}(w) = \min_{f_W = f} \sup_v \|weigts\ into\ v\|^d_p$$

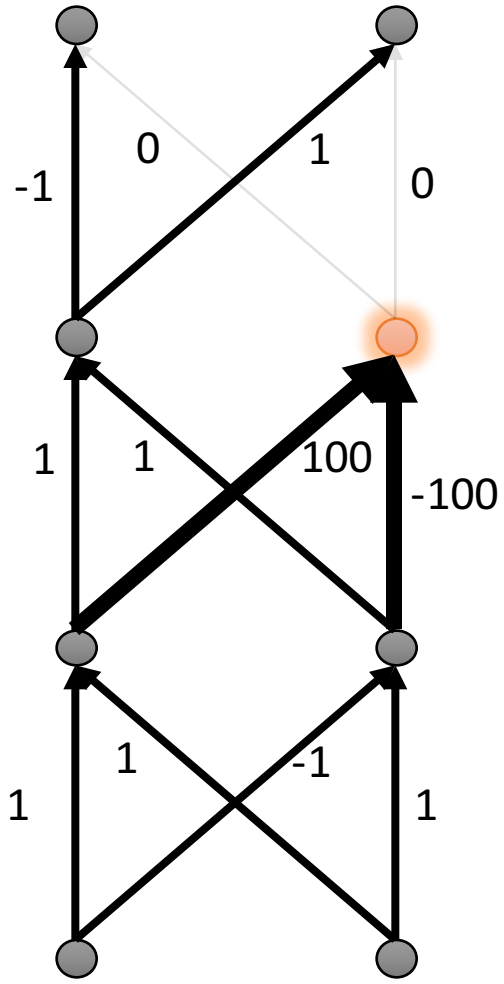
Efficient Computation

- As fast as a forward step on a single data point!

- Do a forward step on the same network with
  - Weights $\widetilde{w_e} = |w_e|^p$
  - Input $x = (1,1, \dots, 1)$

$$\phi_p(w) = \sqrt[p]{\text{sum of outputs}}$$

Path-Regularization



- ✓Path-Norm looks at the aggregated influence of all the weight.
- ✓Path-Norm is invariant to unit rescaling.
- ✓Can be computed efficiently at the cost of a single forward step

Optimization is Tied to Choice of Geometry

Steepest descent w.r.t. a geometry:

$$w^{(t+1)} = \arg\min_{w} \eta \left\langle \nabla \mathrm{L}\left(\mathrm{w}^{(t)}\right), w \right\rangle + \delta\left(w^{(t+1)}, w\right)$$

✓improve the objective as much as possible

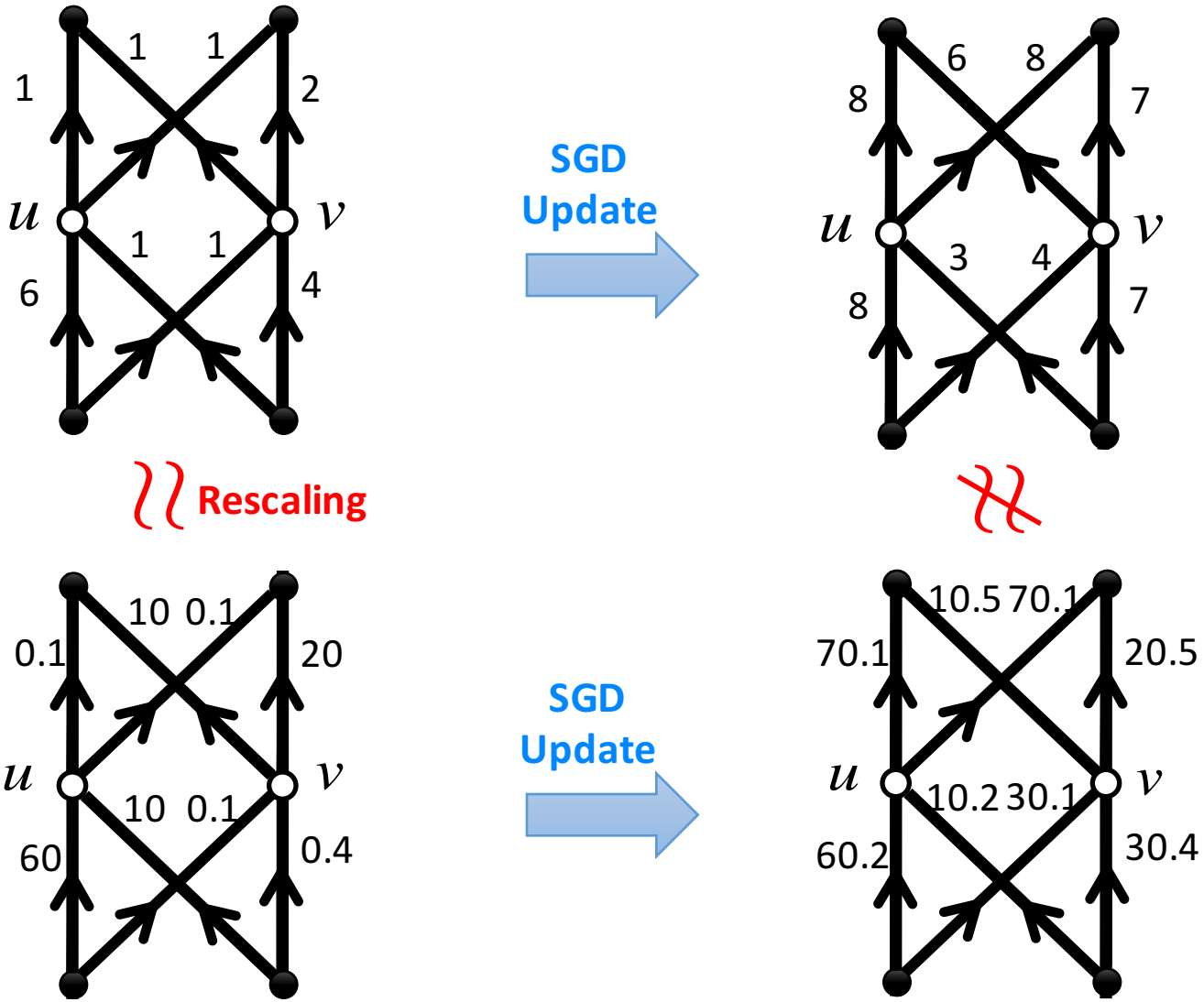
✓only a small change in the model.

Examples:

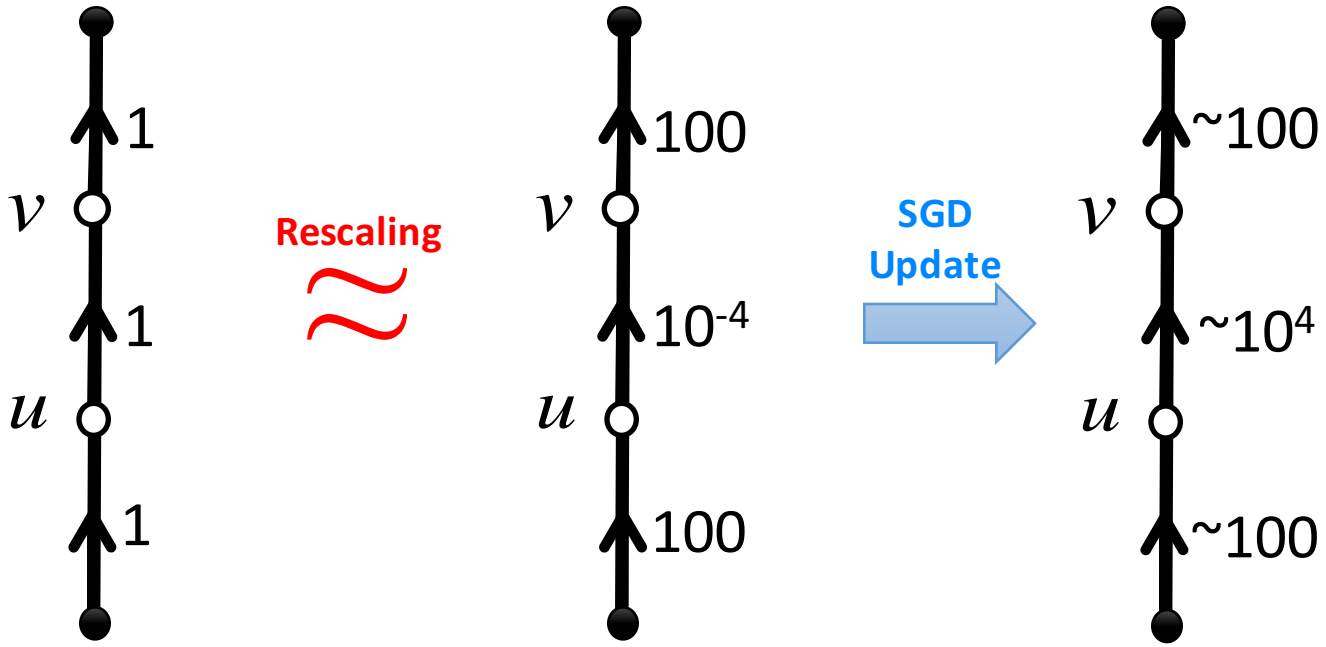
- Gradient Descent: Steepest descent w.r.t ℓ_2
- Coordinate Descent: Steepest descent w.r.t. ℓ_1

What's the geometry appropriate for deep networks?

SGD (Steepest Descent w.r.t. $\mu_{2,2}$) Not Invariant to Rescaling

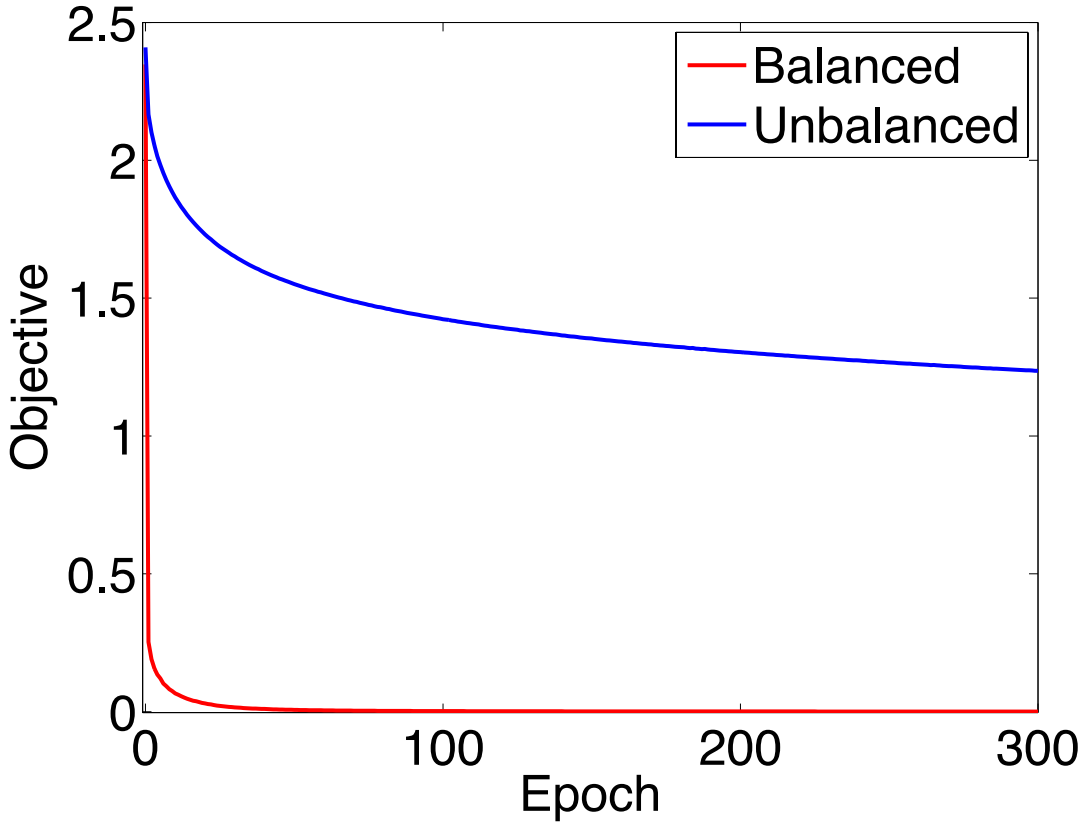


SGD on Unbalanced Networks



SGD updates for a simple case: the input is x = 1, thresholds are set to zero (constant), the stepsize is 1, and the gradient with respect to output is δ = −1.

SGD with Unbalanced



Feed-forward network with two hidden layers, each containing 4000 hidden units.

Path-SGD

- Steepest Descent on $\pi(w)$:

$$w^{(t+1)} = \arg\min_{w} \langle \nabla L(w^{(t)}), w \rangle + \frac{1}{2} \left\| \pi(w) - \pi(w^{(t)}) \right\|_p^2$$

- Update Step (after approximation):

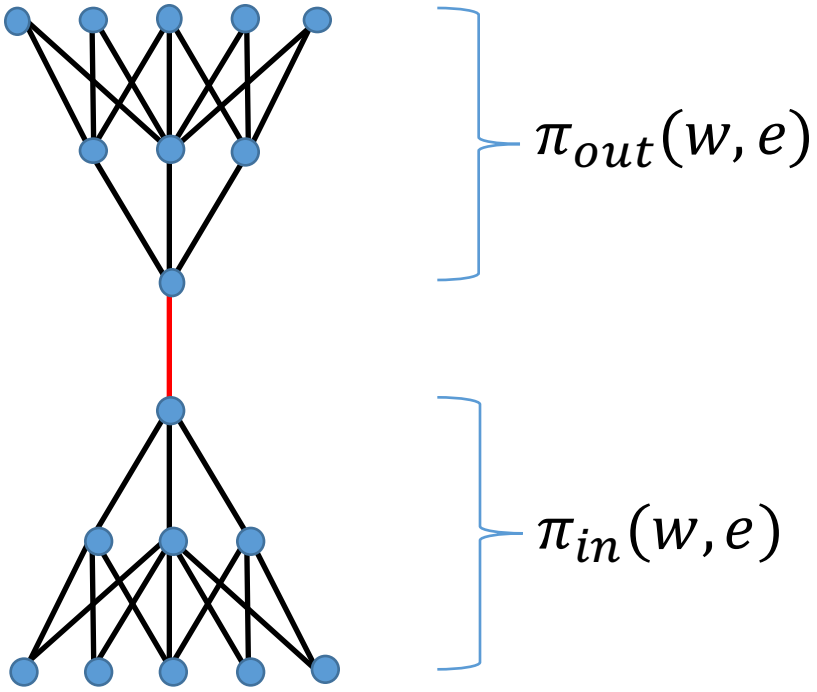
$$w_e^{(t+1)} = \mathrm{w}_e^{(t)} - \frac{\eta}{\frac{\partial L}{\partial w_e}(w^{(t)})}$$

- Invariant to rebalancing

[Neyshabur, Salakhudtinov, Srebo. NIPS'15]

Path-SGD

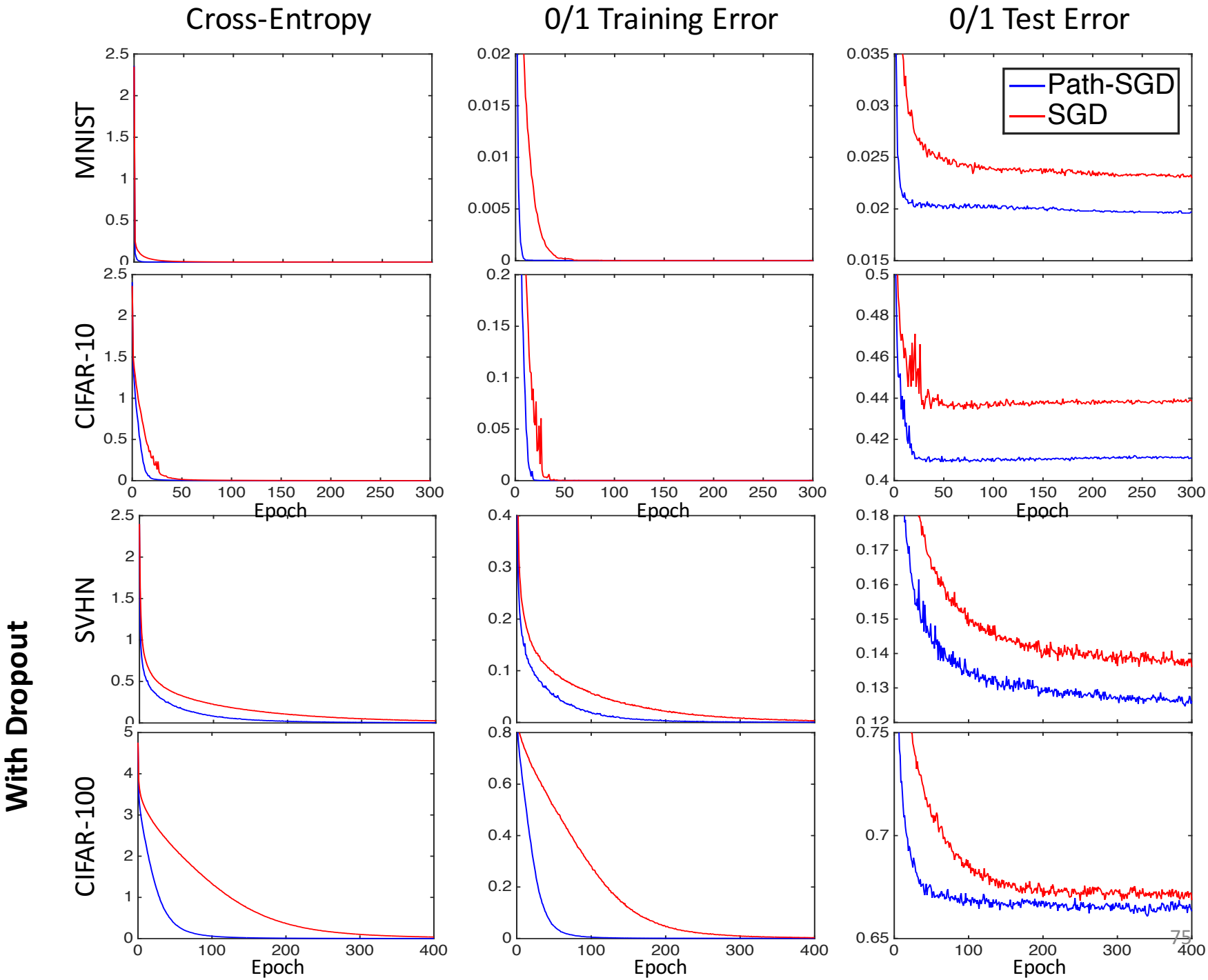
$$w_e^{(t+1)} = w_e^{(t)} - \frac{\eta}{[\kappa_e(w^{(t)})]^2} \frac{\partial L}{\partial w}(w^{(t)})$$



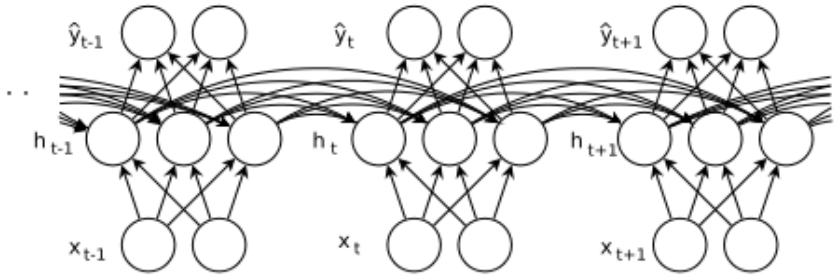
$$\kappa_e(w) = \|\pi_{in}(w,e)\|_p \|\pi_{out}(w,e)\|_p$$

✓ **As fast as a forward-backward step on a single data point** ☺

[Neyshabur, Salakhudtinov, Srebo. NIPS'15]

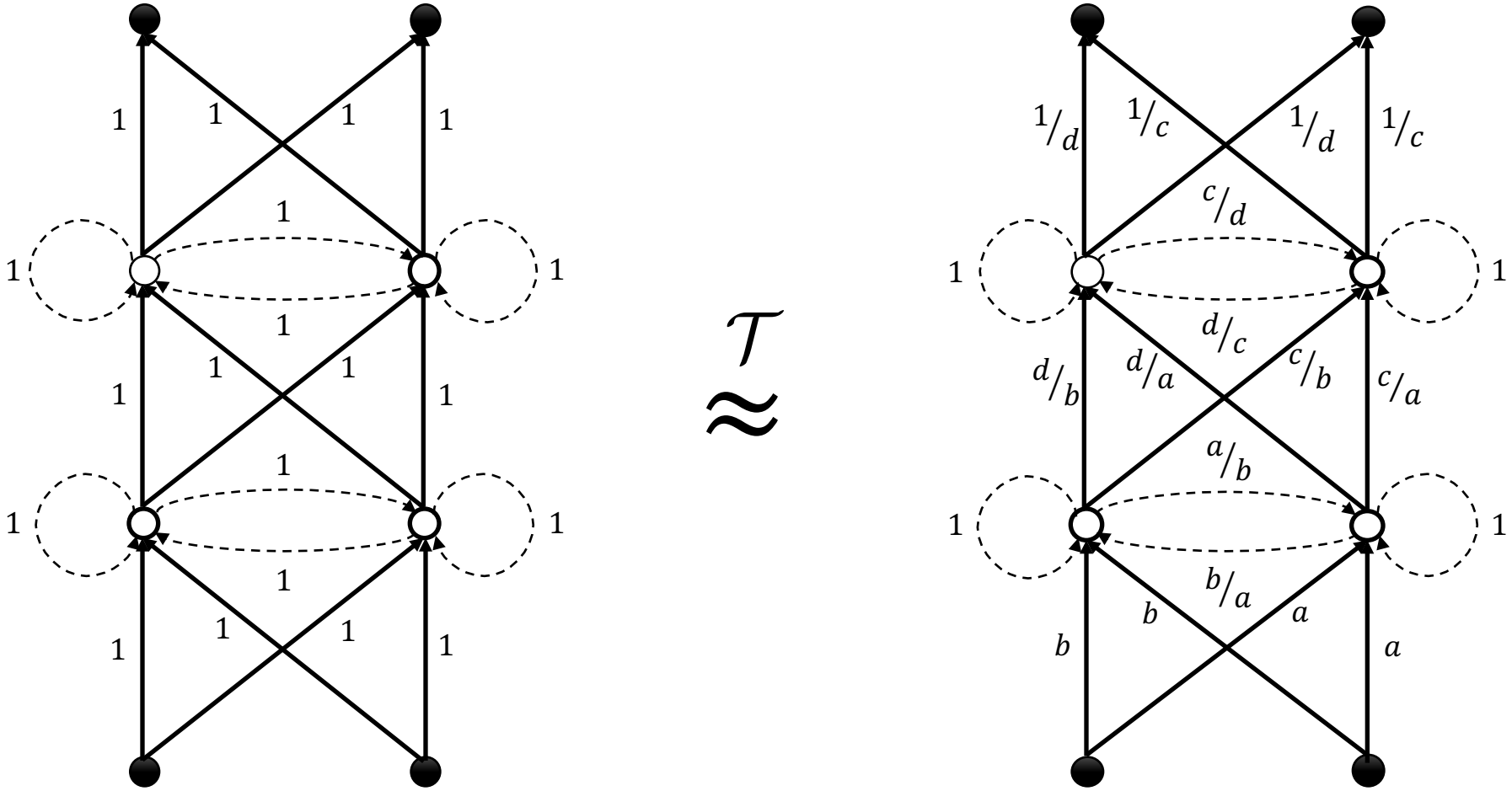

Cross-Entropy
0/1 Training Error
0/1 Test Error
With Dropout
MNIST
CIFAR-10
SVHN
CIFAR-100
Path-SGD
SGD
Epoch

Recurrent Neural Networks



- Sequence to sequence learning architecture
  - Translations, parsing, speech
- Notoriously hard to train: large effective depth and repeated weights
  - With saturating activations (sigmoid, tanh, ramp) : gradient decay
  - With unbounded activation (ReLU): weight explosion
- Most empirical success is with LSTMs, GRUs
  - Different, more complicated architecture
  - Saturating activations, but "short circuits" gradient paths
  - Is this more complicated modeling really necessary?

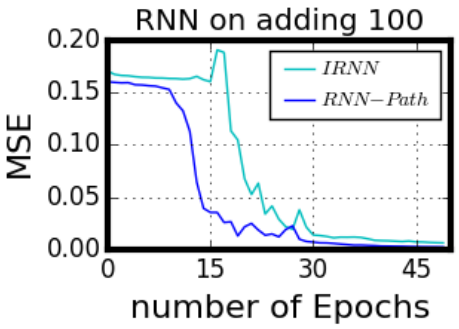
Invariances in RNNs

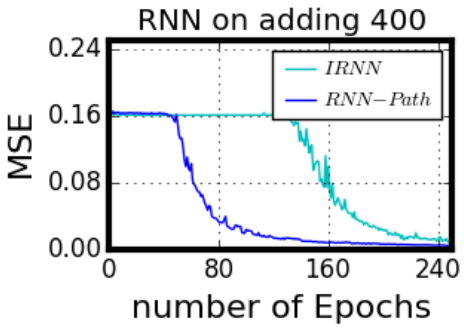


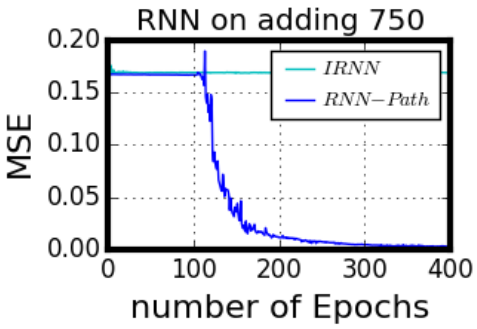
✓ Instead: Path SGD to plain vanilla ReLU RNNs

[Neyshabur, Wu, Salakhudtinov, Srebo. NIPS'16]

Path-SGD on RNNs







	Penn Tree-Bank	Text8
Plain ReLU SGD	1.55	1.65
Plain ReLU Path-SGD	1.47	1.58
LSTM	1.41	1.52
%gap to LSTM bridged	57%	53%

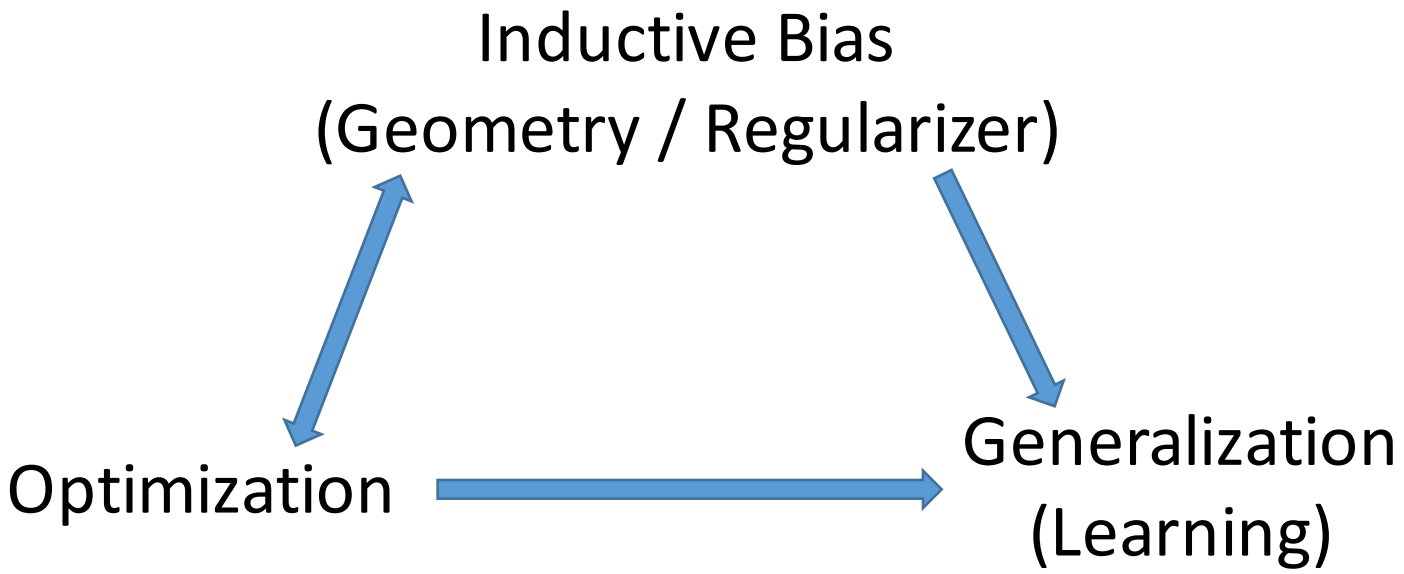
[Neyshabur, Wu, Salakhudtinov, Srebo. NIPS'16]

A Unified Framework

$$\gamma_v^2 = w_{\to v}^\top R_v w_{\to v}$$

R_v	Optimize $\widetilde{w} = \gamma^2 w$	Diagonal steepest descent	Steepest descent
$diag(\gamma_{in(v)})$	Weight Normalization	**Path-SGD**	
$Cov[z_{in(v)}]$	**Batch Normalization**	Data-Dependent-Path-SGD (DDP-SGD)	
$E[z_{in(v)} z_{in(v)}^\top]$		Diagonal Natural Gradient	**Natural Gradient**
$\alpha Cov + (1-a)\gamma$		DDP-SGD	
Invariant	No	Yes	Yes

[Neyshabur, Tomioka, Salakhudtinov, Srebo. ICLR'16]


Inductive Bias
(Geometry / Regularizer)
Optimization
Generalization
(Learning)

Thanks!