Aston University

Space of Functions Computed by Deep-layered Machines

Bo Li, Alexander Mozeika and David Saad

Joint KIT/TUe Workshop on Neuromorphic High-Speed Communications (NeuCos)
9 December 2021

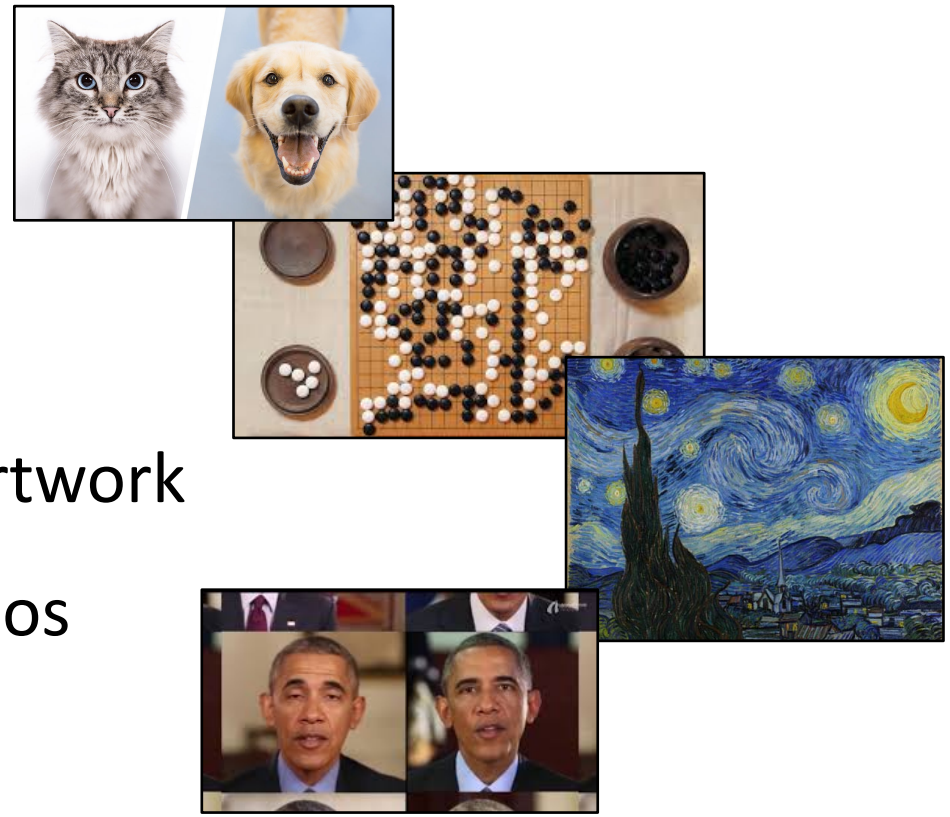
Outline

- Deep learning machines – successes and open questions
- Statistical mechanics of learning from examples and why entropy in function-space matters
- Related set-ups: Continuous/discrete weights, dense/sparse networks, correlated weights, convolutional neural networks, sensitivity to input perturbations, binarization/sparsification and finite-size effects
- Typical functions computed by DLM and recurrent networks
- Summary and future work

Bo Li and David Saad, Physical Review Letters **120**, 248301 (2018)
Bo Li and David Saad, Jour. Phys. A, **53**, 104002 (2020)
A. Mozeika, B. Li, and D. Saad, Physical Review Letters **125**, 168301, (2020)

Deep Learning Engineering Successes

- Computer vision
- Speech recognition
- Go, ATARI
- Composing music, artwork
- Generating fake videos



It is unclear:

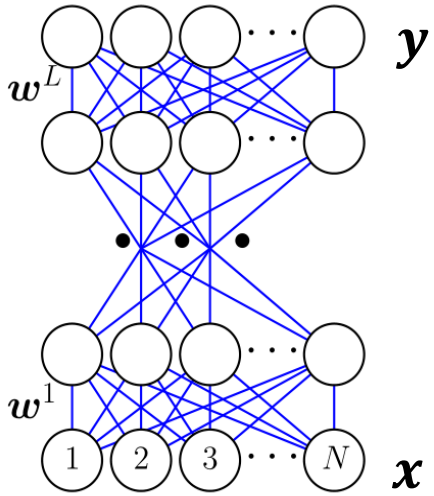
Which functions do they represent? How to optimize configurations? How to improve training? What do internal representations represent?

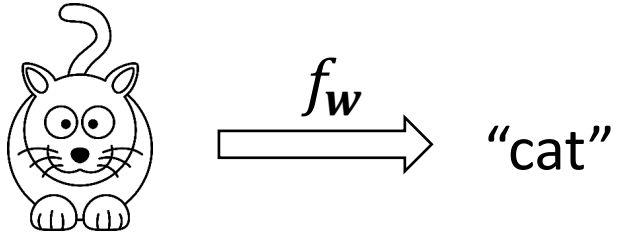
Deep Learning Machines

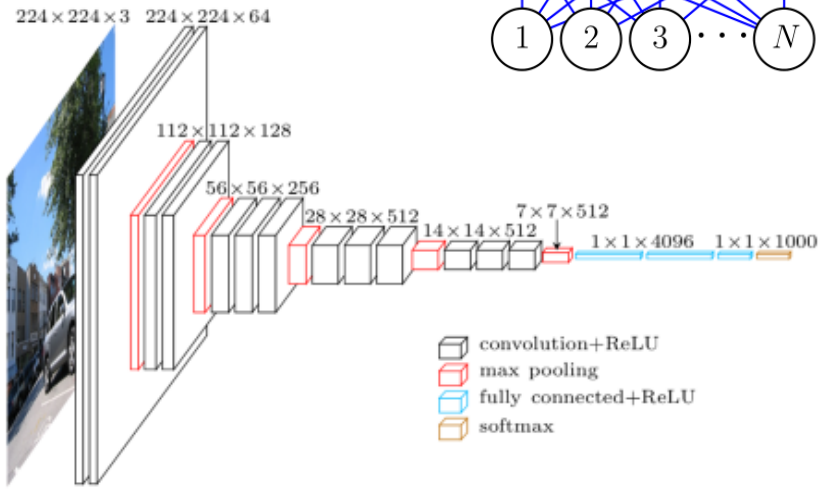
Implement an input-output mapping

$$\boldsymbol{y} = f_{\boldsymbol{w}}(\boldsymbol{x}),$$

where the parameters $\boldsymbol{w}$ are to be estimated based on the training data $\{(\xi^{\mu}, \sigma^{\mu})\}_{\mu=1,2,\ldots P}$ to perform a desired mapping.







We want to understand:

(i) Their **generalization ability** even with numerous parameters

(ii) Nature of the **internal representations**

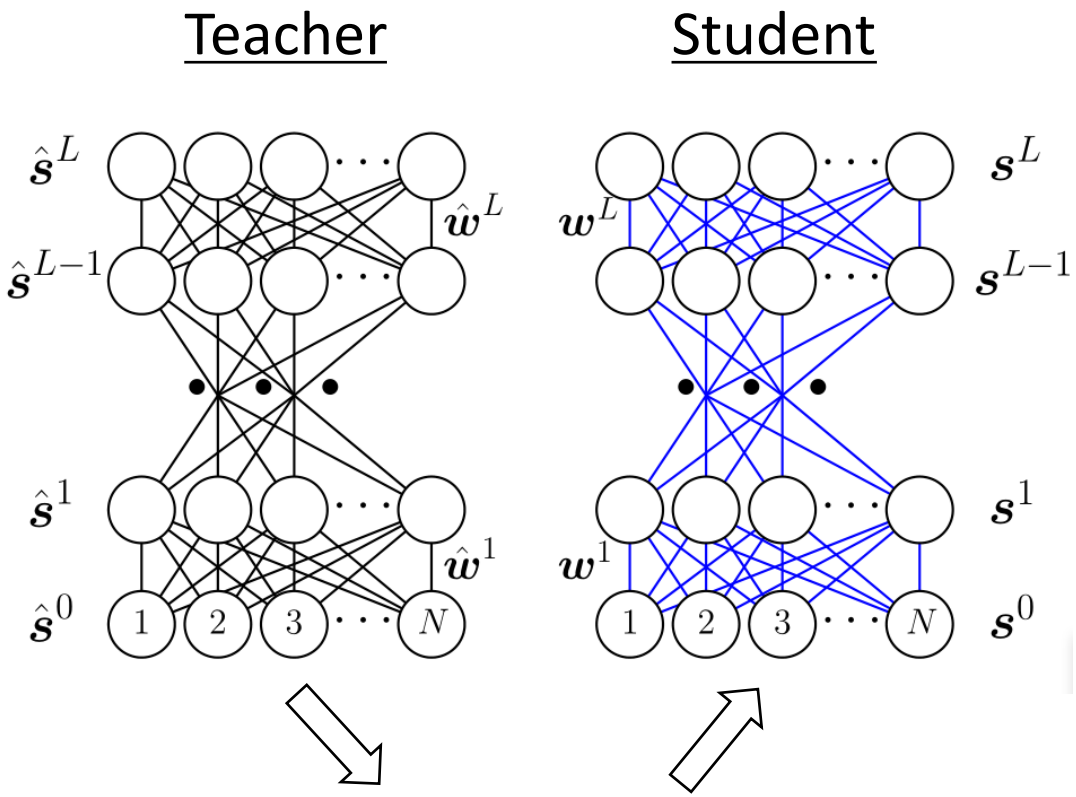
(iii) Which problems are easier/more difficult to solve?

Macroscopic Analysis –Typical Behavior

- Mapped to **disordered** systems of **infinite dimension**
- **Typical** behavior (in contrast to worse-case) of storage capacity and generalization curves (mostly single-layer)
- Technically quite involved (single or two-layer systems)
- Two-layer analysis – in the online setting only
- Input data structure and internal representations are rarely addressed

A. Engle and C. Van den Broeck, 2001,
D. Saad and S.A. Solla, Phys. Rev. Lett., 74, 4337, (1995); D. Saad and M. Rattray, Phys. Rev. Lett., 79, 2578 (1997)
M. Rattray, D. Saad and S. Amari, Phys. Rev. Lett., 81, 5461 (1998), S. Goldt, M. Mézard, F. Krzakala, and L. Zdeborová, Phys. Rev. X, 10, 041044 (2020)

Teacher-student Scenario for DLM?



$\{\xi^1, \sigma^1\}, \{\xi^2, \sigma^2\} \ldots, \{\xi^\mu, \sigma^\mu\}, \ldots, \{\xi^p, \sigma^p\}$

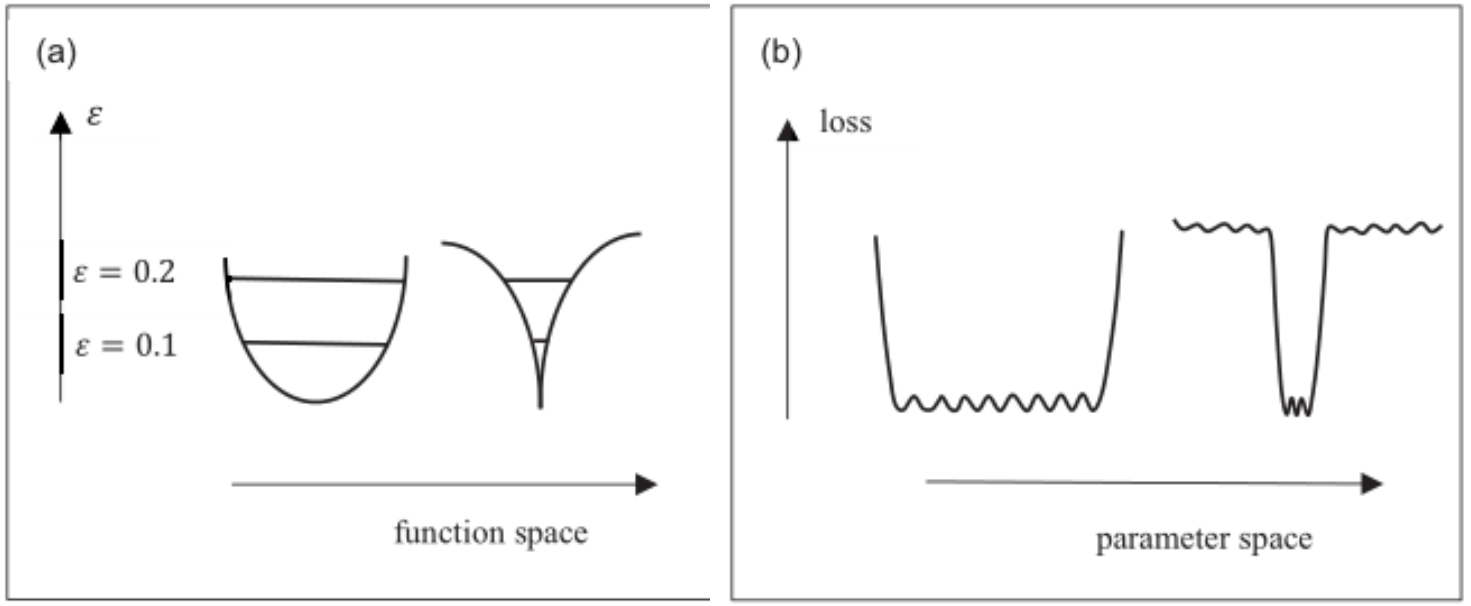
Difficulties:

- Constraints imposed by the examples (input-output pairs) on the hidden units are complex –**recursive nonlinear mapping.**
- Permutation, reflection and other **symmetries/invariances of hidden units**, no simple relation between teacher-student overlap and generalization error.

Function Space, Error and Entropy

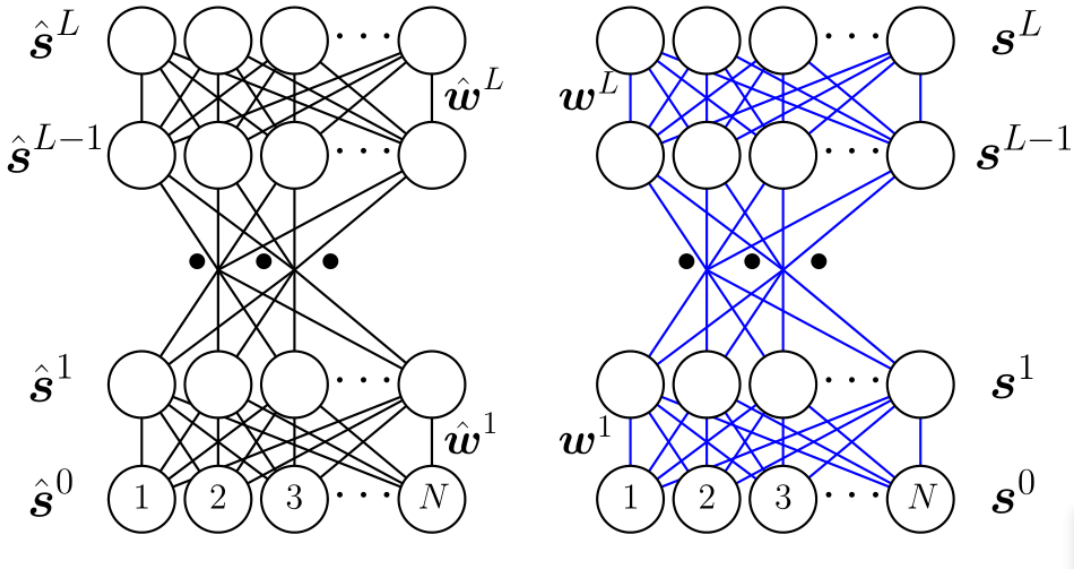
- We would like to approximate a reference/target function $f_{\hat{w}}$, as closely as possible from data.
- Given noisy data, sub-optimal training methods - more relevant to find *good approximations*. **How many such functions exist?**
- Conjecture - The entropy (log-volume) of functions at distance-ε away from $f_{\hat{w}}$ indicates how easy it is to obtain them.

Baldassi et al PRL 2015, PNAS 2016, JSM 2020



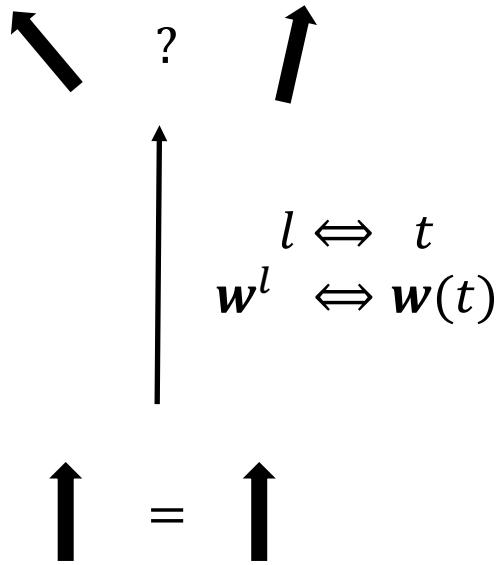
Exploring Function Space in DLM

Reference function $f_{\hat{w}}$ Perturbed function f_{w}



We map the DLM to **disordered spin systems** with discrete dynamics, $\hat{s}_i^l, s_i^l \in \{1, -1\}$, activation function is **sign function** $sgn(x)$.

The framework can be generalized to **real variables** and **other activation functions.**



Investigate the function sensitivity under small perturbations

$$w^l = \text{Perturb}(\hat{w}^l)$$

Poole et al., NIPS 2016 - Mean field theory to study input sensitivity and expressivity

DLM as a Stochastic Dynamical System

- The layer evolution of two coupled DLMs:

$$P\left(\hat{\boldsymbol{s}}^l \middle| \hat{\boldsymbol{w}}^l, \hat{\boldsymbol{s}}^{l-1}, \beta\right) = \prod_i \frac{\exp \beta \hat{s}_i^l \hat{h}_i^l(\hat{\boldsymbol{w}}^l, \hat{\boldsymbol{s}}^{l-1})}{2 \cosh \beta \hat{s}_i^l \hat{h}_i^l(\hat{\boldsymbol{w}}^l, \hat{\boldsymbol{s}}^{l-1})}, P\left(\boldsymbol{s}^l \middle| \boldsymbol{w}^l, \boldsymbol{s}^{l-1}, \beta\right) = \cdots,$$

$\hat{h}_i^l\left(\hat{\boldsymbol{w}}^l, \hat{\boldsymbol{s}}^{l-1}\right) = \sum_j \hat{w}_{ij}^l \, \hat{s}_j^{l-1}/\sqrt{N}$, β is the inverse-temperature quantifying the noise level; deterministic rule in the zero-noise limit $\beta \to \infty$.

- Any observable is given by

$$\langle O \rangle := \sum_{\{\hat{\boldsymbol{s}}^l, \boldsymbol{s}^l\}} O \cdot P(\hat{\boldsymbol{s}}^0) \delta_{\hat{\boldsymbol{s}}^0, \boldsymbol{s}^0} \prod_l P\left(\hat{\boldsymbol{s}}^l \middle| \hat{\boldsymbol{w}}^l, \hat{\boldsymbol{s}}^{l-1}, \beta\right) \cdot P(\boldsymbol{s}^l | \boldsymbol{w}^l, \boldsymbol{s}^{l-1}, \beta),$$

summed over all the trajectories subject to the path measure.

- For discrete spins, the **overlap** between activities of the two systems is of interest

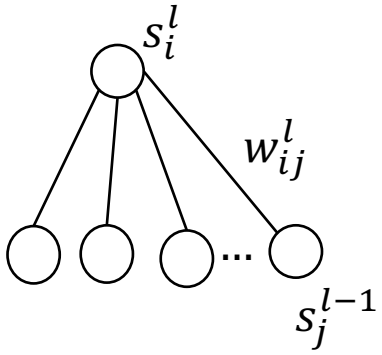
$$q^l(\hat{\boldsymbol{w}}, \boldsymbol{w}, \beta) = \frac{1}{N} \sum_i \langle \hat{s}_i^l s_i^l \rangle$$

Generating Functional Analysis

- Generating functional (characteristic function)

$$\Gamma[\hat{\boldsymbol{\psi}}, \boldsymbol{\psi}] := \left\langle \exp\left\{ -i \sum_{l,i} \left(\hat{\psi}_i^l \hat{s}_i^l + \psi_i^l s_i^l \right) \right\} \right\rangle,$$

moments such as magnetization $\langle \hat{s}_i^l \rangle$ and overlap $\langle \hat{s}_i^l s_i^l \rangle$ can be obtained by differentiating $\Gamma[\hat{\boldsymbol{\psi}}, \boldsymbol{\psi}]$; angled brackets – average over all paths.

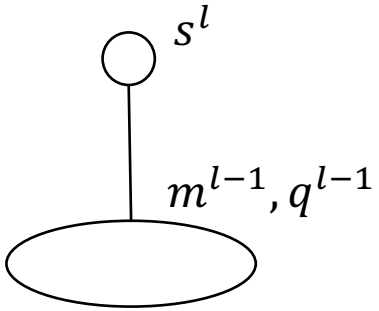


- Interested in the **typical** behavior of an ensemble of networks $\hat{\boldsymbol{w}} \sim P(\hat{\boldsymbol{w}})$, overbar – quenched average

$$\overline{\Gamma[\hat{\boldsymbol{\psi}}, \boldsymbol{\psi}]} := \sum_{\{\hat{\boldsymbol{w}}^l, \boldsymbol{w}^l\}} \Gamma[\hat{\boldsymbol{\psi}}, \boldsymbol{\psi}] P(\hat{\boldsymbol{w}})\, P(\boldsymbol{w})$$

$$= \int \prod_l \frac{dQ^l dq^l}{2\pi/N} e^{N\Psi[\boldsymbol{q},\boldsymbol{Q}]} \approx e^{N\Psi[\boldsymbol{q}^*,\boldsymbol{Q}^*]}, \text{ in the limit } N \to \infty,$$

Represented by macroscopic order parameters; the saddle point $\boldsymbol{q}^*, \boldsymbol{Q}^* = \mathrm{extr}_{\boldsymbol{q},\boldsymbol{Q}} \boldsymbol{\Psi}(\boldsymbol{q}, \boldsymbol{Q})$ satisfies certain self-consistent **mean-field** equation.

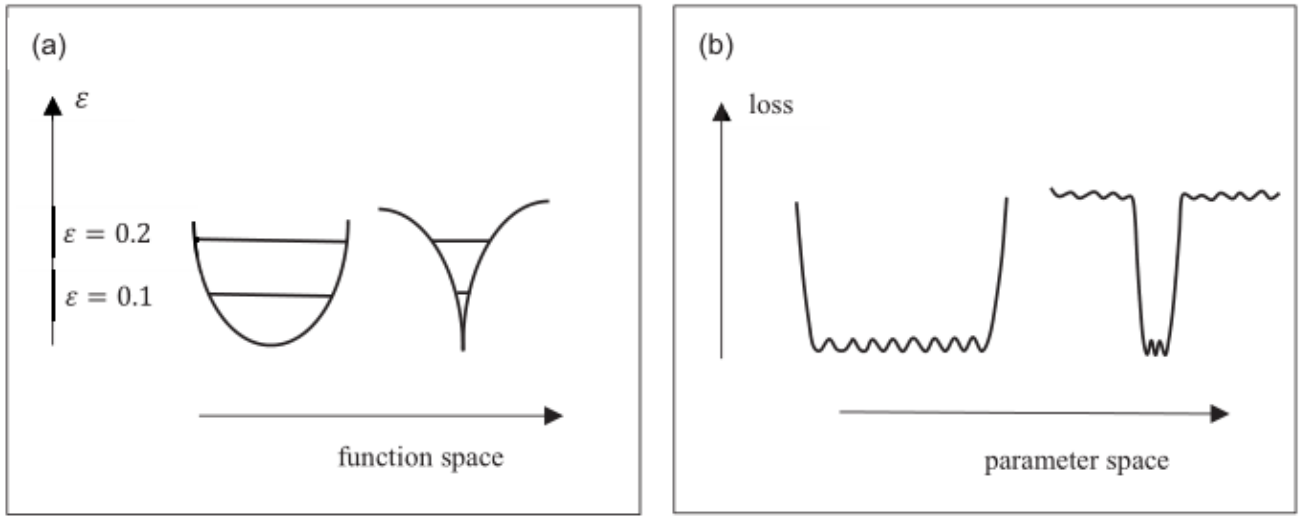


Function Error and Entropy

- For discrete spins, the **overlap** between internal representations the two systems is of interest $q^l(\hat{\boldsymbol{w}}, \boldsymbol{w}, \beta) = \frac{1}{N}\sum_i \langle \hat{s}_i^l s_i^l \rangle$, calculated using Generating Functional Analysis
- Function error is defined as the **expected Hamming distance** of output layers between $f_{\hat{\boldsymbol{w}}}$ and $f_{\boldsymbol{w}}$

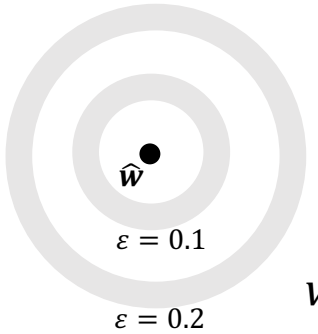
$$\varepsilon := \frac{1}{2N}\sum_{i=1}^{N} \overline{\left\langle \left| \hat{s}_i^L - s_i^L \right| \right\rangle} = \frac{1}{2}(1 - q^L),$$

which provides a distance measure between $f_{\hat{\boldsymbol{w}}}$ and $f_{\boldsymbol{w}}$.



Fully-connected Networks – Continuous/Binary Weights

Consider fully-connected networks, with $P(\hat{\boldsymbol{w}}_i^l) = \prod_j P(\hat{w}_{ij}^l)$



Continuous weights:

$$\hat{w}_{ij}^l \sim \mathcal{N}(0, \sigma^2)$$

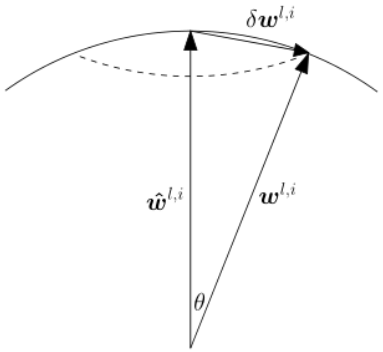
Perturbation strength at layer l

$$w_{ij}^l = \sqrt{1-(\eta^l)^2}\,\hat{w}_{ij}^l + \eta^l \delta w_{ij}^l$$

Binary weights:

$$P(\hat{w}_{ij}^l) = \frac{1}{2}\delta(\hat{w}_{ij}^l - 1) + \frac{1}{2}\delta(\hat{w}_{ij}^l + 1)$$

$$P(w_{ij}^l) = (1-p^l)\delta(w_{ij}^l - \hat{w}_{ij}^l) + p^l \delta(w_{ij}^l + \hat{w}_{ij}^l)$$



Typical overlaps: $q^l = \frac{2}{\pi}\sin^{-1}(\sqrt{1-(\eta^l)^2}\,q^{l-1})$ $\qquad q^l = \frac{2}{\pi}\sin^{-1}((1-2p^l)q^{l-1})$

Entropy of $f_{\boldsymbol{w}}$: $S_{\text{con}} = \frac{1}{L}\sum_l \log \eta^l$ $\qquad S_{\text{bin}} = \frac{1}{L}\sum_l -p^l \log p^l - (1-p^l)\log(1-p^l)$

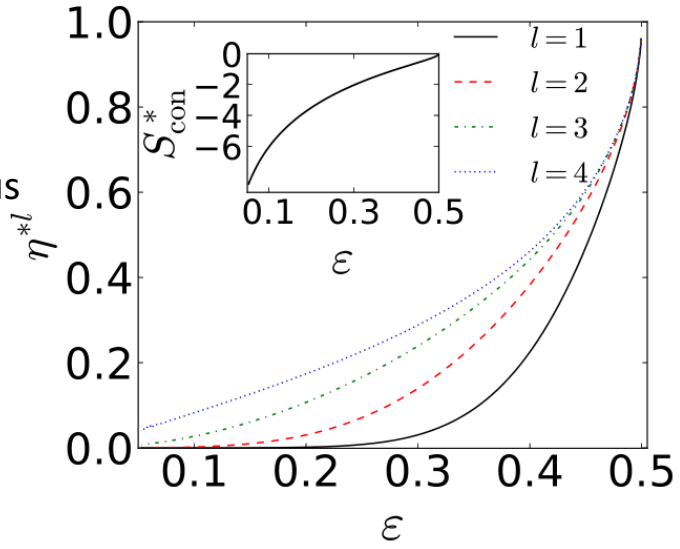
The distance-ε surface of $f_{\boldsymbol{w}}$ with volume $\Omega(\{\eta^l\}) = \exp N_p\, S_{\text{con}}\,(\{\eta^l\})$, is **exponentially** dominated by the maximum-entropy solutions when $N_p \to \infty$:

$$\eta^{*l} = \arg\max_{\eta^l} S_{\text{con}}\,(\{\eta^l\}), \quad \text{s.t.} \quad q^L(\{\eta^l\}) = 1 - 2\varepsilon$$

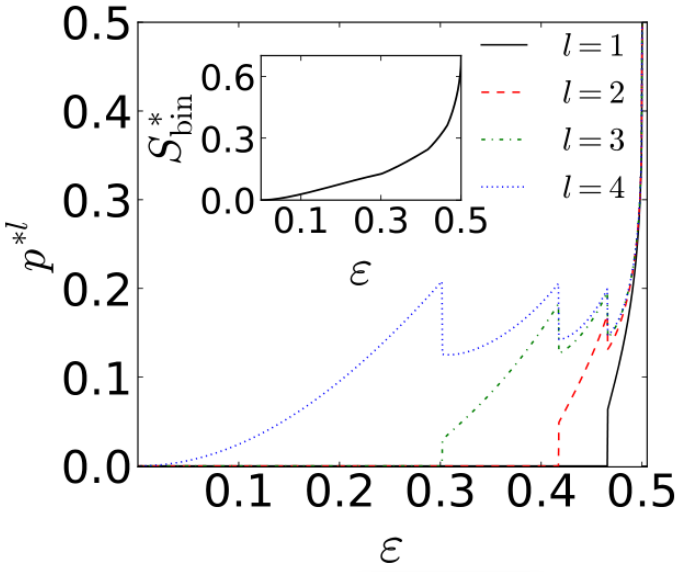
[B. Poole et al.,NIPS 2016]

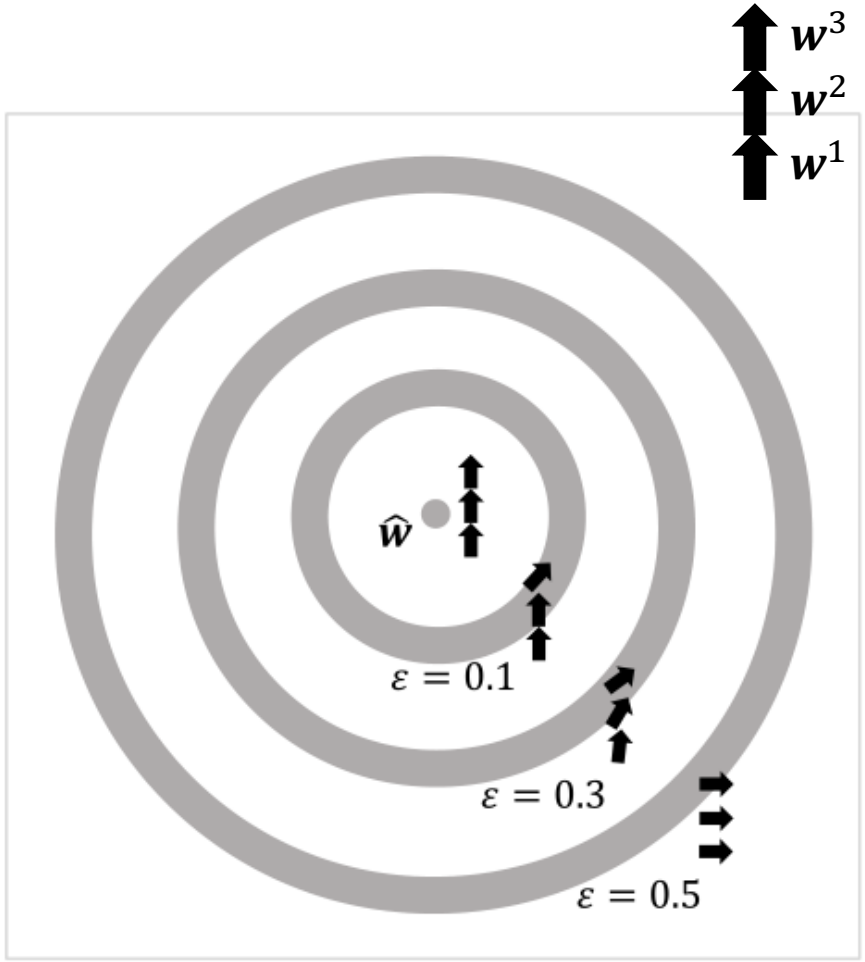
Earlier Layers Converge First When Decreasing ε

Continuous weights



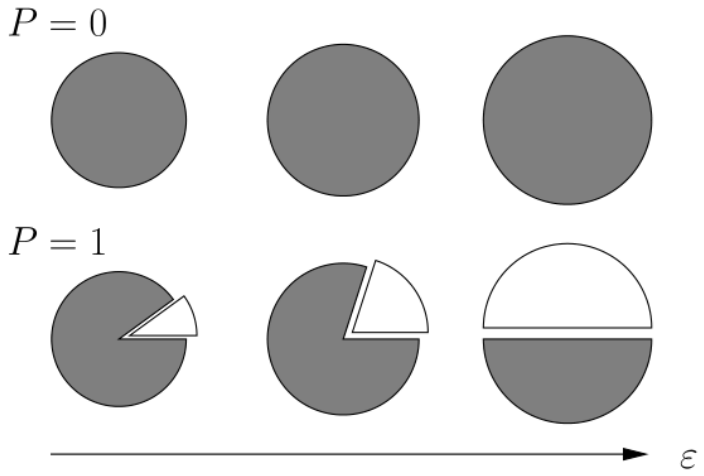
Binary weights





M. Raghu et al, *ICML* 2017

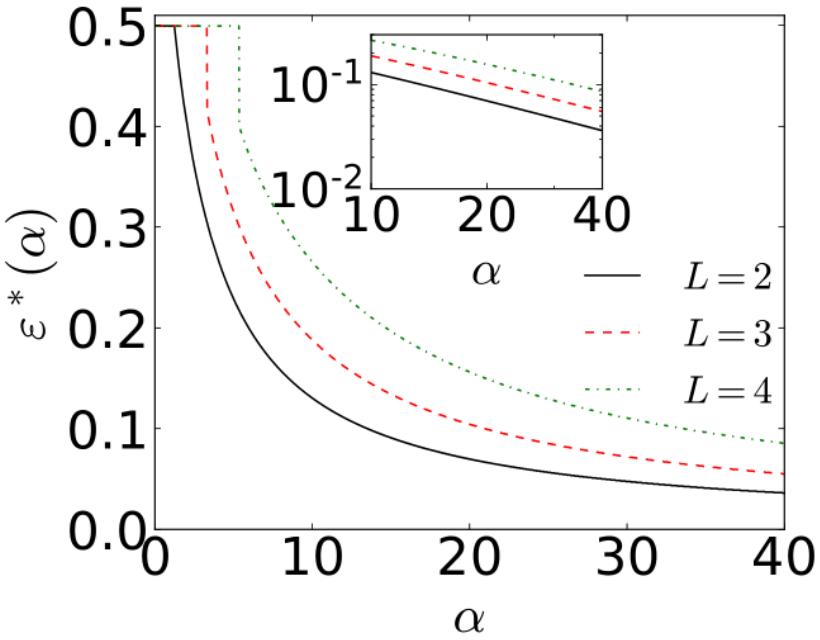
Approximate Generalization Curve (dense DLM with continuous weights)



$$\Omega_\alpha(\varepsilon) = \Omega_{\text{tot}}\left(\{\eta^{*l}\}, \varepsilon\right) = \Omega_0(\varepsilon)(1-\varepsilon)^P$$

$$P = \alpha L N^2 \qquad \varepsilon^*(\alpha) = \text{argmax}_\varepsilon \Omega_\alpha(\varepsilon)$$

Annealed theory of learning

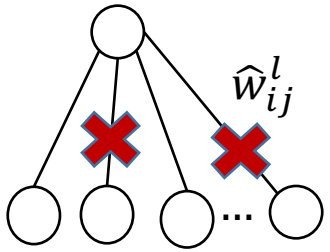


Relevant in small ε (large α) limit.

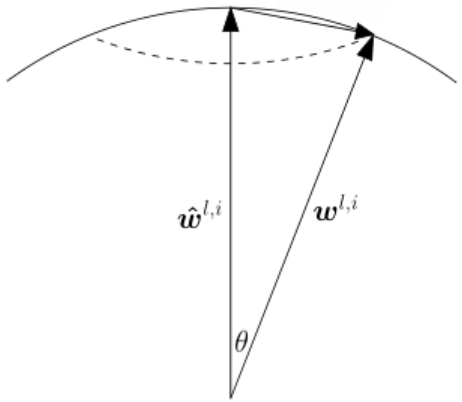
A. Engle and C. Van den Broeck, 2001

Perturbations Through Dilution/Discretization

Weight disconnection

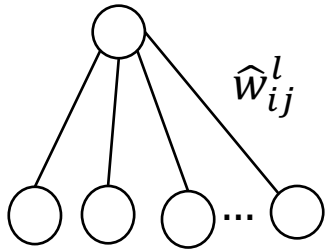


$\Leftrightarrow$



$\Leftrightarrow$

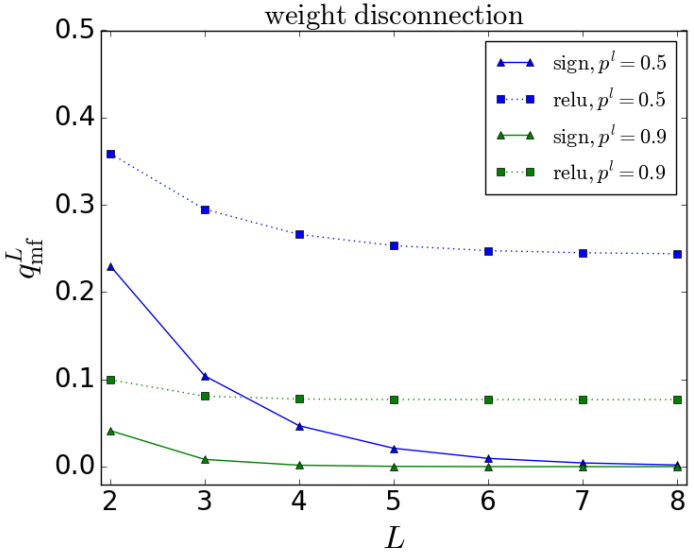
Weight discretization

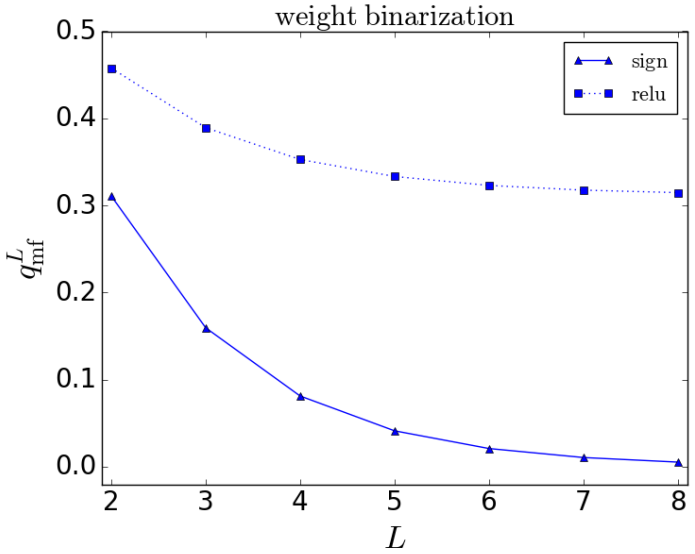


$w_{ij}^{l} = \mathrm{sgn}(\widehat{w}_{ij}^{l})$

$\theta^{l} = \sin^{-1}\sqrt{p^{l}}$

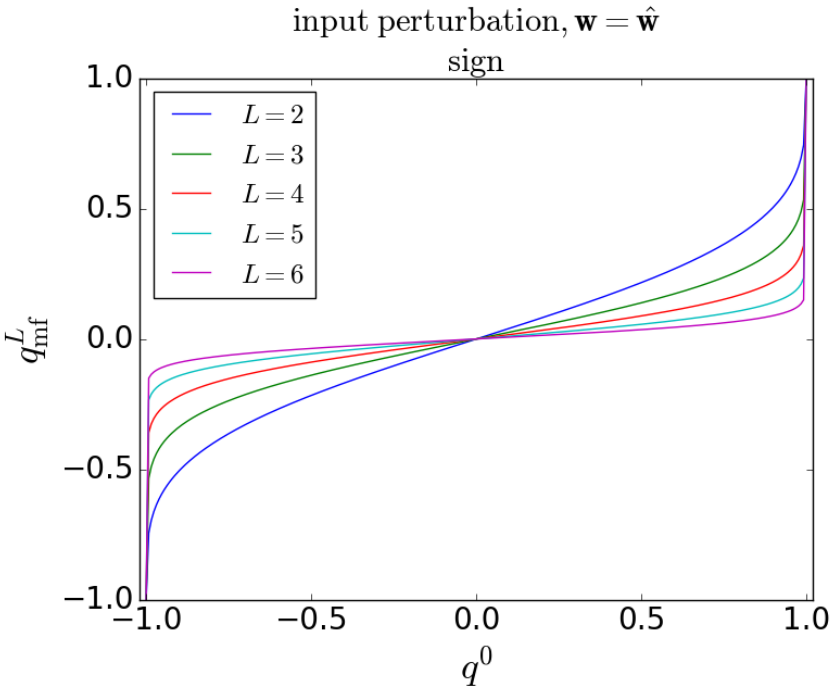
$\theta^{l} = \cos^{-1}\sqrt{2/\pi} \approx 37°$ [A. Anderson,et al.,ICLR 2018]

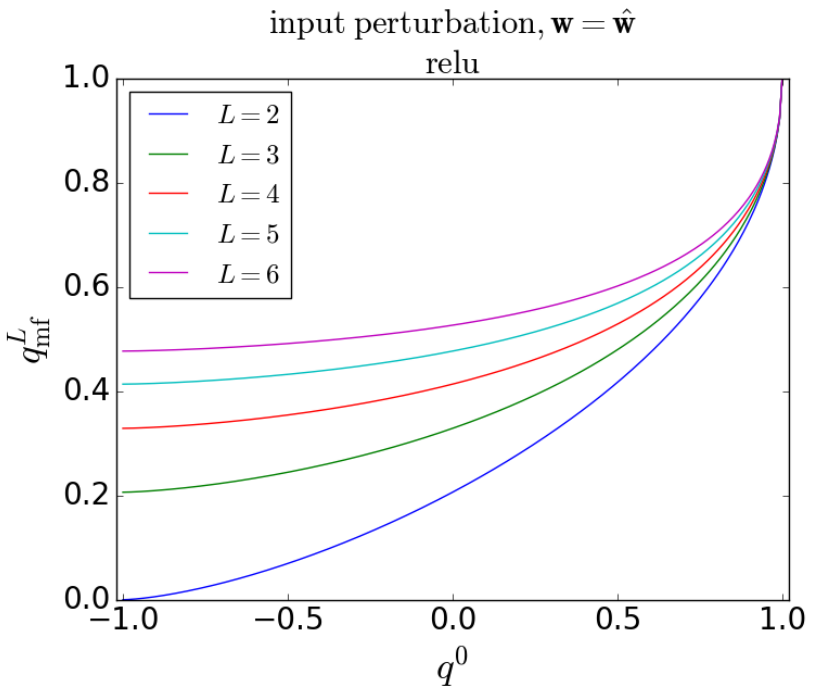


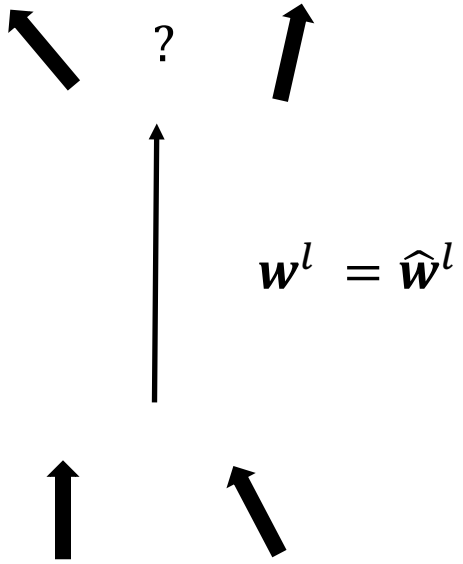


Deep ReLU networks with *random* weights are robust to disconnecting or binarizing weights.

Sign vs ReLU - Input Sensitivity







sensitivity under input perturbations

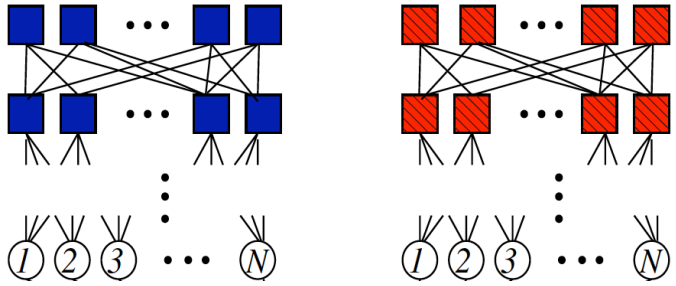
$$\boldsymbol{s}^0 = \mathrm{Perturb}(\hat{\boldsymbol{s}}^0)$$

Deep ReLU networks with *random* weights compute simple functions.

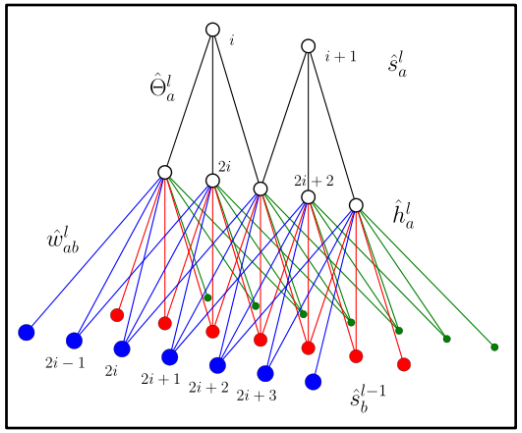
[B. Poole et al.,NIPS 2016]

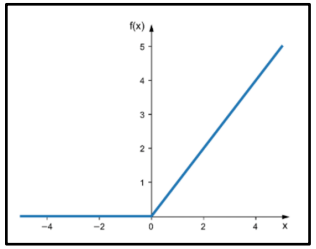
Interim Summary – Other Setups we Investigated

- **Sparse architectures -** as before, where each node is **randomly** connected to k units in the previous layer and $\widehat{w}_{ij}^{l} = 1$.



- **Weight dilution/discretization–** ReLU vs sign
- **Continuous variable values** and ReLU activation function $\phi(x) = \max(0, x)$.
- **Convolutional neural networks**
- **Correlated weights**
- **Input sensitivity –** ReLU vs sign





Space of Functions Generated by Sparse DLM

Consider Boolean functions computed by deep-layered machines,

$$f : \{+1, -1\}^n \to \{+1, -1\}.$$

E.g., for $n = 2$, there are $2^n = 4$ possible input patterns in total, which are

$$x_0 = (+1, +1), \quad x_1 = (+1, -1),$$
$$x_2 = (-1, +1), \quad x_3 = (-1, -1).$$

The Boolean function $f(\cdot)$ is fully specified by the string of length 4

$$\boldsymbol{f} \equiv \big(f(x_0), f(x_1), f(x_2), f(x_3)\big).$$

Boolean Input-Output Relation in DLM

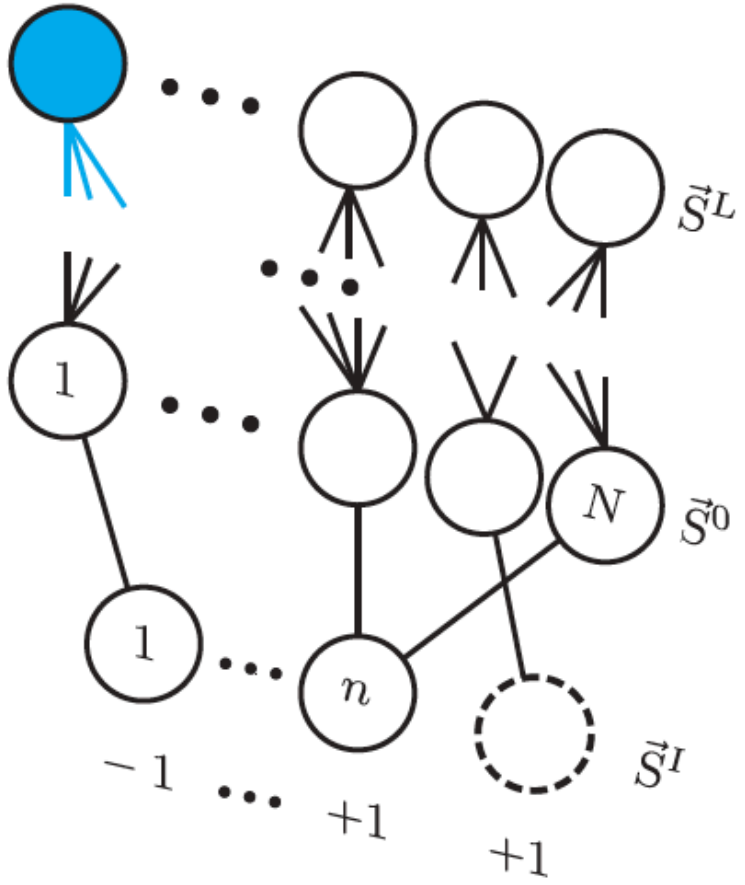
Computation of a specific input pattern $\vec{s} \in \{+1, -1\}^n$

$$P(\vec{S}^L|\vec{s}) = \sum_{\vec{S}^{L-1}\ldots\vec{S}^0} P(\vec{S}^0|\vec{s}) \prod_{\ell=1}^{L} P(\vec{S}^\ell|\vec{S}^{\ell-1}),$$

where

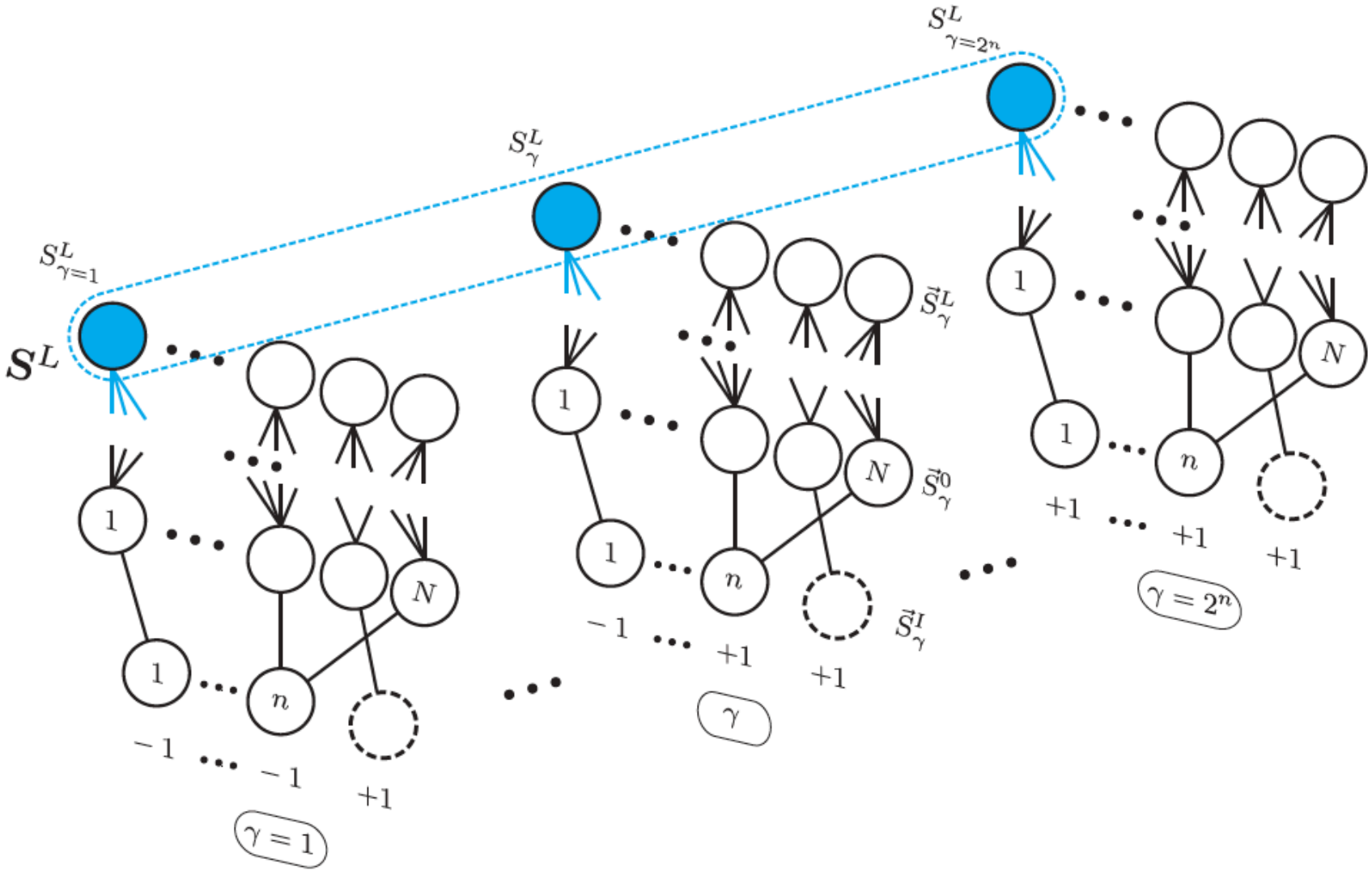
$$P^0(\vec{S}^0|\vec{s}) = \prod_{i=1}^{N} \delta\left[S_i^0, S_{n_i}^I(\vec{s})\right],$$

$$P(S_i^\ell|\vec{S}^{\ell-1}) = \delta\left[S_i^\ell, \alpha_i^\ell(\vec{S}^{\ell-1})\right].$$



Functions Generated by Layered Networks

Computation of all possible input patterns: $\prod_{\gamma=1}^{2^n} P(\vec{S}_\gamma^L | \vec{s}_\gamma)$



The Framework – Probability of Functions

Distribution of functions computed on the final layer

$$P_N^L(\boldsymbol{f}) = \frac{1}{N}\sum_{i=1}^{N}\left\langle \prod_{\gamma=1}^{2^n} \delta\left(f_\gamma, S_{i,\gamma}^L\right)\right\rangle .$$

To compute macroscopic observables such as $P_N^L(\boldsymbol{f})$, introduce the generating functional

$$\overline{\Gamma[\{\psi_{i,\gamma}^\ell\}]} = \overline{\sum_{\{S_{i,\gamma}^\ell\}\forall \ell,i,\gamma} \prod_{\gamma=1}^{2^n} P(\vec{S}_\gamma^L|\vec{s}_\gamma) e^{-\mathrm{i}\sum_{\ell,i,\gamma}\psi_{i,\gamma}^\ell S_{i,\gamma}^\ell}}.$$

GF Analysis – Saddle Point Equations

Weight disorder: $W_{ij}^{\ell} \sim \mathcal{N}(0, \sigma^2)$

By introducing the overlap $q_{\gamma\gamma'}^{\ell,\ell'} = (1/N)\sum_i \overline{\langle S_{i,\gamma}^{\ell} S_{i,\gamma'}^{\ell'} \rangle}$, the GF can be computed as

$$\overline{\Gamma} = \int \{\mathrm{d}\boldsymbol{q}\, \mathrm{d}\boldsymbol{Q}\} \mathrm{e}^{N\Psi[\boldsymbol{q},\boldsymbol{Q}]}.$$

The saddle points of $\Psi[\boldsymbol{q}, \boldsymbol{Q}]$ give rise to the typical behaviors

$$\frac{\partial \Psi}{\partial \boldsymbol{q}} = 0, \qquad \frac{\partial \Psi}{\partial \boldsymbol{Q}} = 0.$$

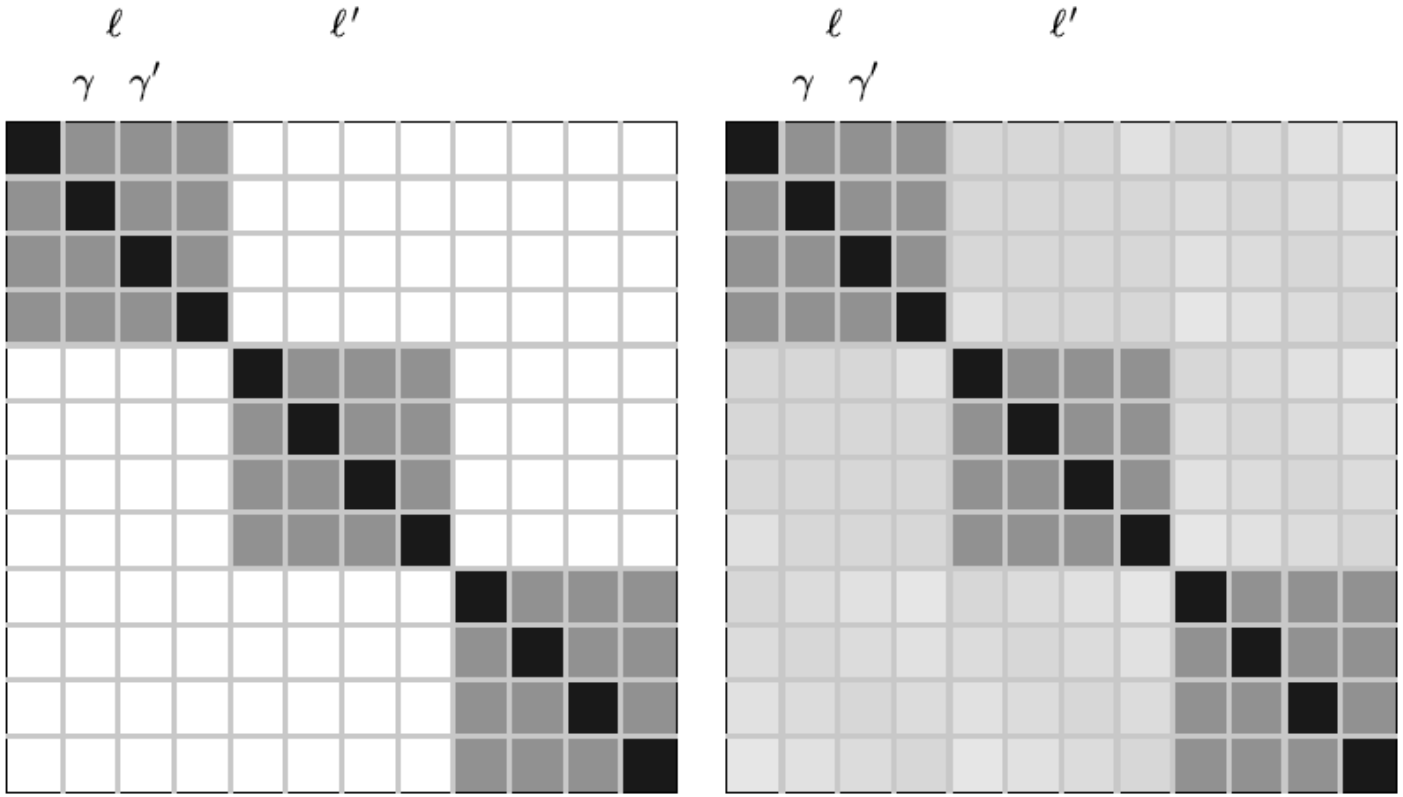
Layered vs Recurrent Networks

Overlaps for both architectures:

$$q^{\ell,\ell'}_{\gamma\gamma'} = \int \mathrm{d}\boldsymbol{H}\, \alpha^{\ell}(h^{\ell}_{\gamma})\, \alpha^{\ell'}(h^{\ell'}_{\gamma'})\, \mathcal{N}(\boldsymbol{H}|\boldsymbol{0}, \boldsymbol{C}), \qquad \ell, \ell' > 0.$$

Covariance matrix $\boldsymbol{C} \propto \boldsymbol{q}$:

The elements admit the same values for **both** architectures when $\ell = \ell'$.



layer-dependent recurrent

Layered vs Recurrent - Functions

To examine the distribution of functions, we only need to consider single-layer activities! (Remind: $P_N^L(\boldsymbol{f}) = \frac{1}{N}\sum_i \left\langle \prod_{\gamma=1} \delta\left(f_\gamma, S_{i,\gamma}^L\right) \right\rangle$.)

- For **both** architectures of neural networks,

$$P^L(\boldsymbol{f}) = \int \mathrm{d}\boldsymbol{h}\, \mathcal{N}(\boldsymbol{h}|\boldsymbol{0}, \boldsymbol{c}^L) \prod_{\gamma=1}^{2^n} \delta\left[f_\gamma, sgn(h_\gamma)\right].$$

- For **both** architectures of Boolean circuits,

$$P^{\ell+1}(\boldsymbol{f}) = \sum_{\boldsymbol{f}_1,\ldots,\boldsymbol{f}_k} \left\{ \prod_{j=1}^{k} P^\ell(\boldsymbol{f}_j) \right\} \prod_{\gamma=1}^{2^n} \delta\left[f_\gamma, \alpha(f_{1,\gamma}, \ldots, f_{k,\gamma})\right],$$

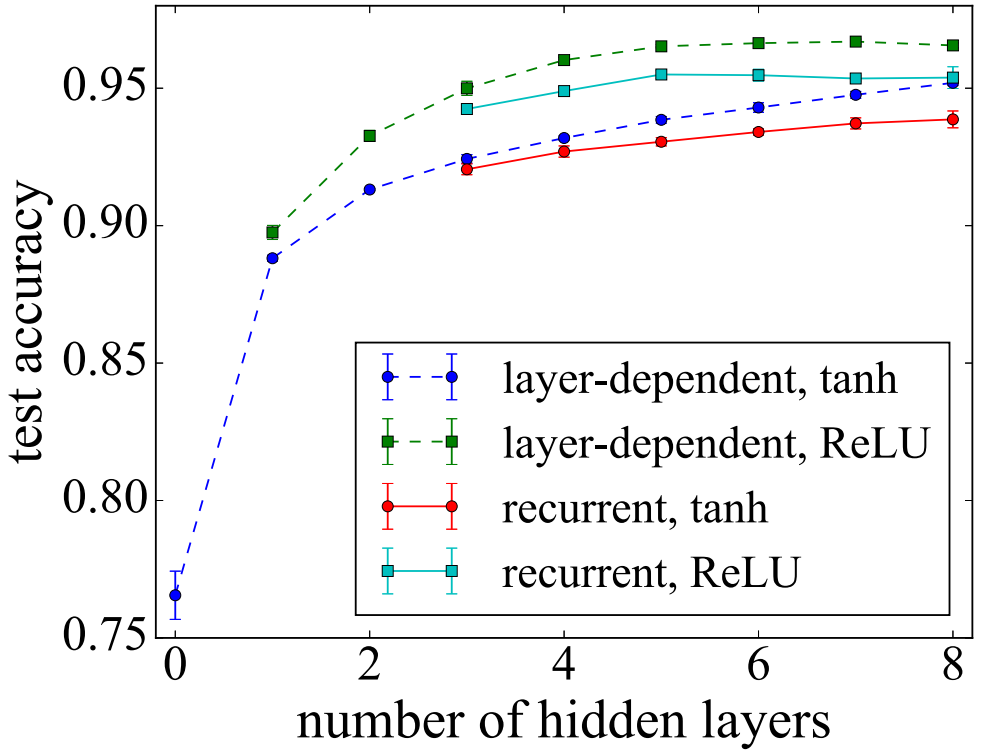
Numerical Results

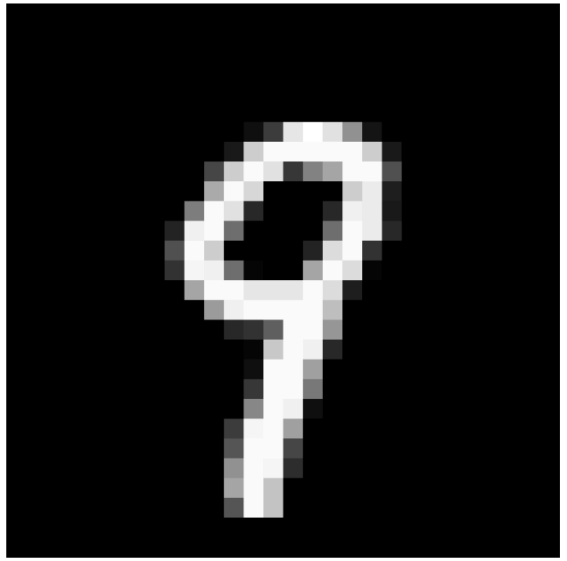
Computation with weight-sharing across layers to save parameters?

Though the theory only applies to **random** machines, we consider training experiments with neural networks on MNIST data.

For recurrent architectures

$$\vec{S}^0 \xrightarrow{W^{\text{in}}} \vec{S}^1 \xrightarrow{W^{\text{hid}}} \vec{S}^2 \xrightarrow{W^{\text{hid}}} \cdots \xrightarrow{W^{\text{hid}}} \vec{S}^{L-1} \xrightarrow{W^{\text{out}}} \vec{S}^L$$





Entropy In Sparse Layered Networks

Effect of gate $\alpha(\cdot)$. Consider $\vec{S}^I = (\vec{s}, -\vec{s}, 1, -1)$.

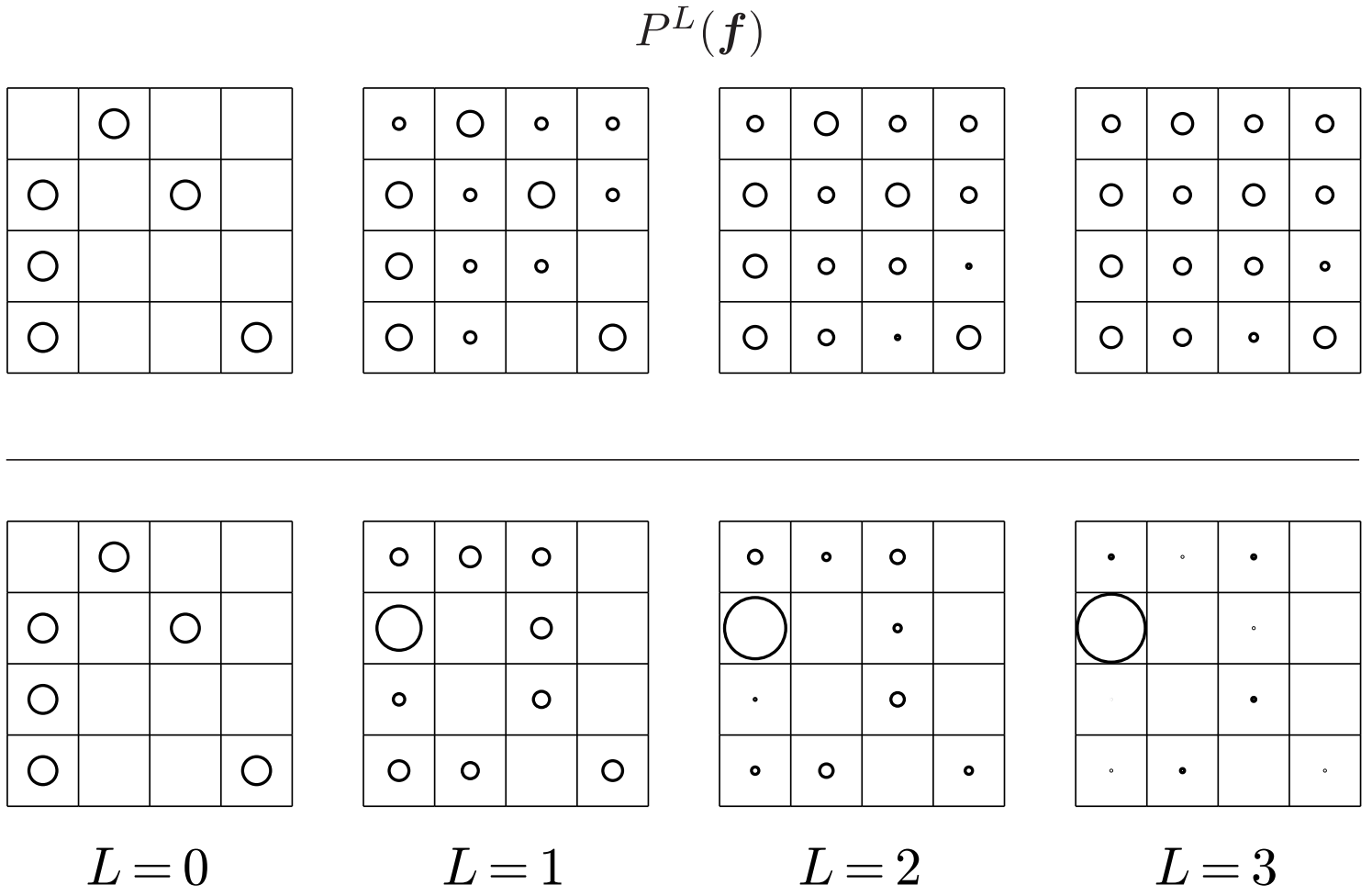
$L \to \infty$,

MAJ3

$P^L(\boldsymbol{f}) = 1/2^{2^n}$

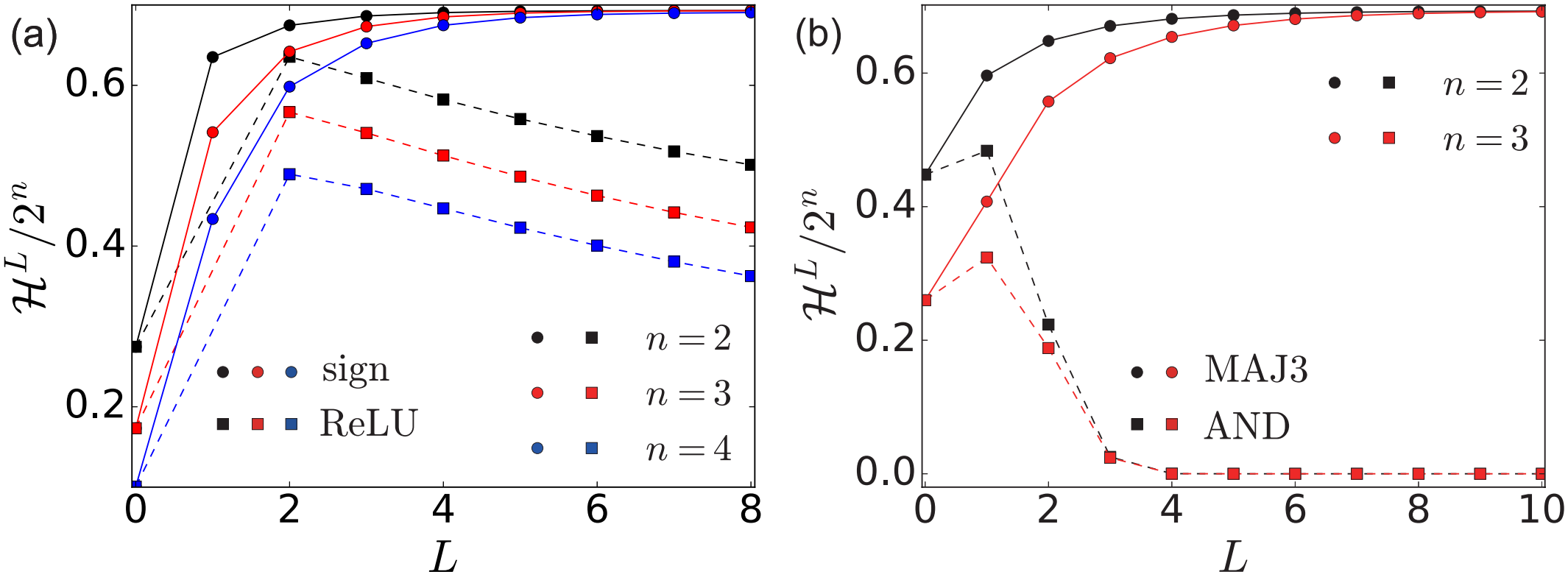
AND

$f(\cdot) = \mathit{False}$



Entropy of Functions – ReLU vs sign

Shannon entropy of Boolean functions $\mathcal{H}^L = -\sum_{\boldsymbol{f}} P^L(\boldsymbol{f}) \log P^L(\boldsymbol{f})$



Simplicity bias of neural networks with ReLU activation when initialized randomly, which arguably plays a role in their generalization ability [G. Valle-Perez et al., ICML 2019; G. De Palma et al., NeurIPS 2019].

Current and Future Work

- We proposed a framework for analyzing functions computed by random DLM.
- Show equivalence between the space of functions generated by random DLM and recurrent architectures.
- Possible computation with a less parameters by weight/connection sharing (may sacrifice accuracy)
- Depending on gate/activation functions, simpler or more complex functions are being represented as layer depth increases.

Future work:

- The role of **over-parametrization** in function landscape, error and generalization.
- Exploring the effect of non-trivially **correlated inputs**, e.g. from a generative model, and go beyond the random reference functions.
- Optimizing **variable hidden layer size**.