FUNCTIONAL DIMENSION OF RELU NEURAL NETWORKS

Boston Symmetry Day
April 7, 2023

J. Elisenda Grigsby
Boston College

BASED ON:

- Joint work with K. Lindsey, R. Meyerhoff, and C. Wu: *"Functional dimension of feedforward ReLU neural networks,"* arXiv: math.MG/2209.04036
- Joint work with K. Lindsey, D. Rolnick: *"Hidden symmetries of ReLU networks,"* (to appear)

SUPERVISED LEARNING PROBLEM:

SUPERVISED LEARNING PROBLEM:

Given a finite data set $\mathscr{D} = \{(x^{(i)}, y^{(i)}) \in \mathbb{R}^{n_0} \times \mathbb{R}^{n_d}\}_{i=1}^{N}$

sampled from an unknown probability distribution $\mathscr{P}(x, y)$

SUPERVISED LEARNING PROBLEM:

Given a finite data set $\mathscr{D} = \{(x^{(i)}, y^{(i)}) \in \mathbb{R}^{n_0} \times \mathbb{R}^{n_d}\}_{i=1}^{N}$

sampled from an unknown probability distribution $\mathscr{P}(x, y)$

1) Choose a parameterized hypothesis class of functions

$$h_\theta : \mathbb{R}^{n_0} \to \mathbb{R}^{n_d} \qquad \theta \in (\Omega = \mathbb{R}^D)$$

SUPERVISED LEARNING PROBLEM:

Given a finite data set $\mathscr{D} = \{(x^{(i)}, y^{(i)}) \in \mathbb{R}^{n_0} \times \mathbb{R}^{n_d}\}_{i=1}^{N}$

sampled from an unknown probability distribution $\mathscr{P}(x, y)$

1) Choose a parameterized hypothesis class of functions

$$h_\theta : \mathbb{R}^{n_0} \to \mathbb{R}^{n_d} \qquad \theta \in (\Omega = \mathbb{R}^D)$$

2) Use an optimization algorithm to find optimal predictor

of labels on *unseen data* drawn from $\mathscr{P}(x, y)$

MOTIVATING QUESTION:

MOTIVATING QUESTION:

Given a parameterized function class for learning, how well does its parameter space model the function class?

MOTIVATING QUESTION:

Given a parameterized function class for learning, how well does its parameter space model the function class?

WHY WE SHOULD CARE:

MOTIVATING QUESTION:

Given a parameterized function class for learning, how well does its parameter space model the function class?

WHY WE SHOULD CARE:

- (Empirical) loss depends only on the function (and the sample data), BUT optimization algorithms proceed in parameter space

MOTIVATING QUESTION:

Given a parameterized function class for learning, how well does its parameter space model the function class?

WHY WE SHOULD CARE:

- (Empirical) loss depends only on the function (and the sample data), BUT optimization algorithms proceed in parameter space
- Function **redundancy** or **inhomogeneity** will bias optimization algorithms

RELU NEURAL NETWORKS

RELU NEURAL NETWORKS

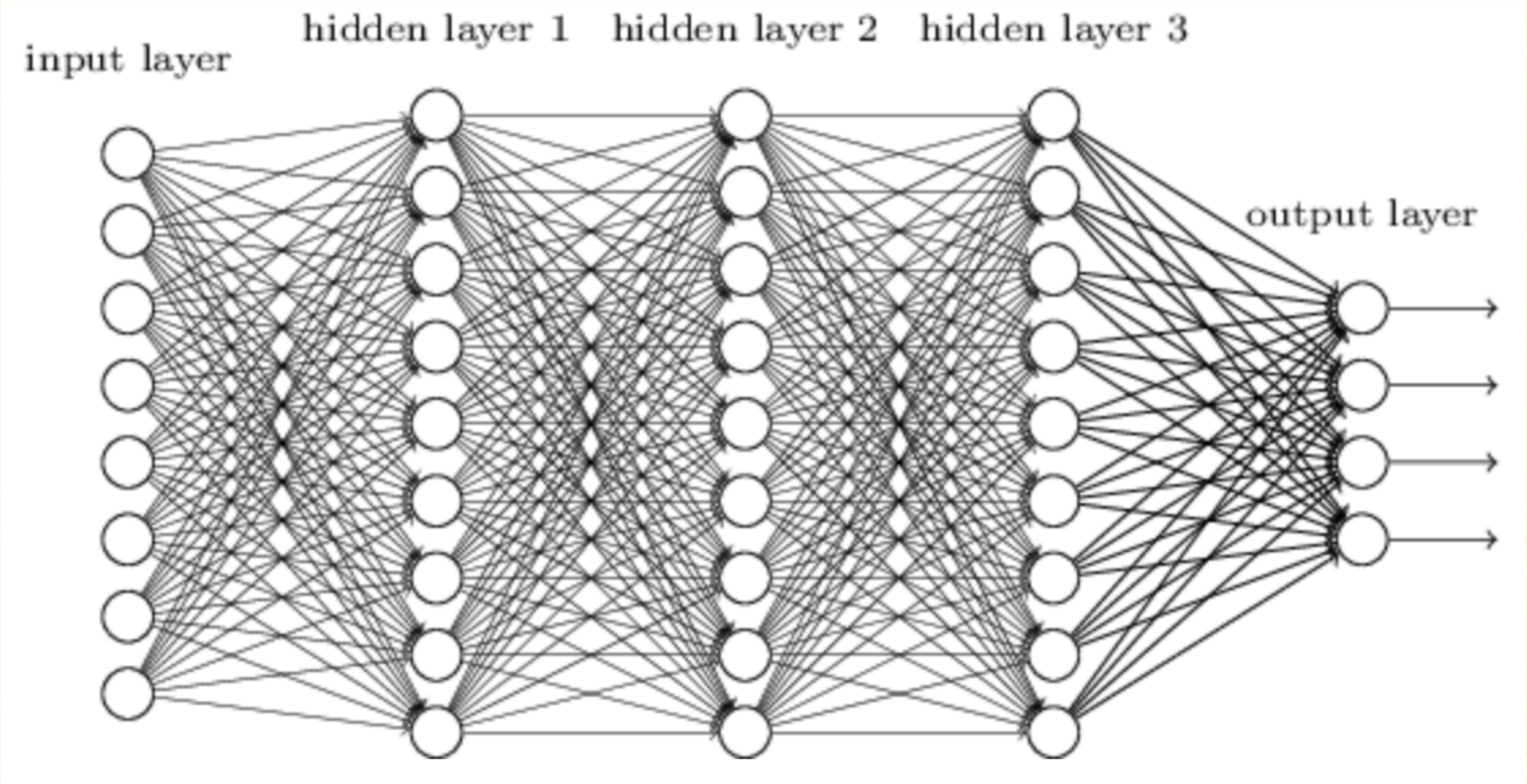
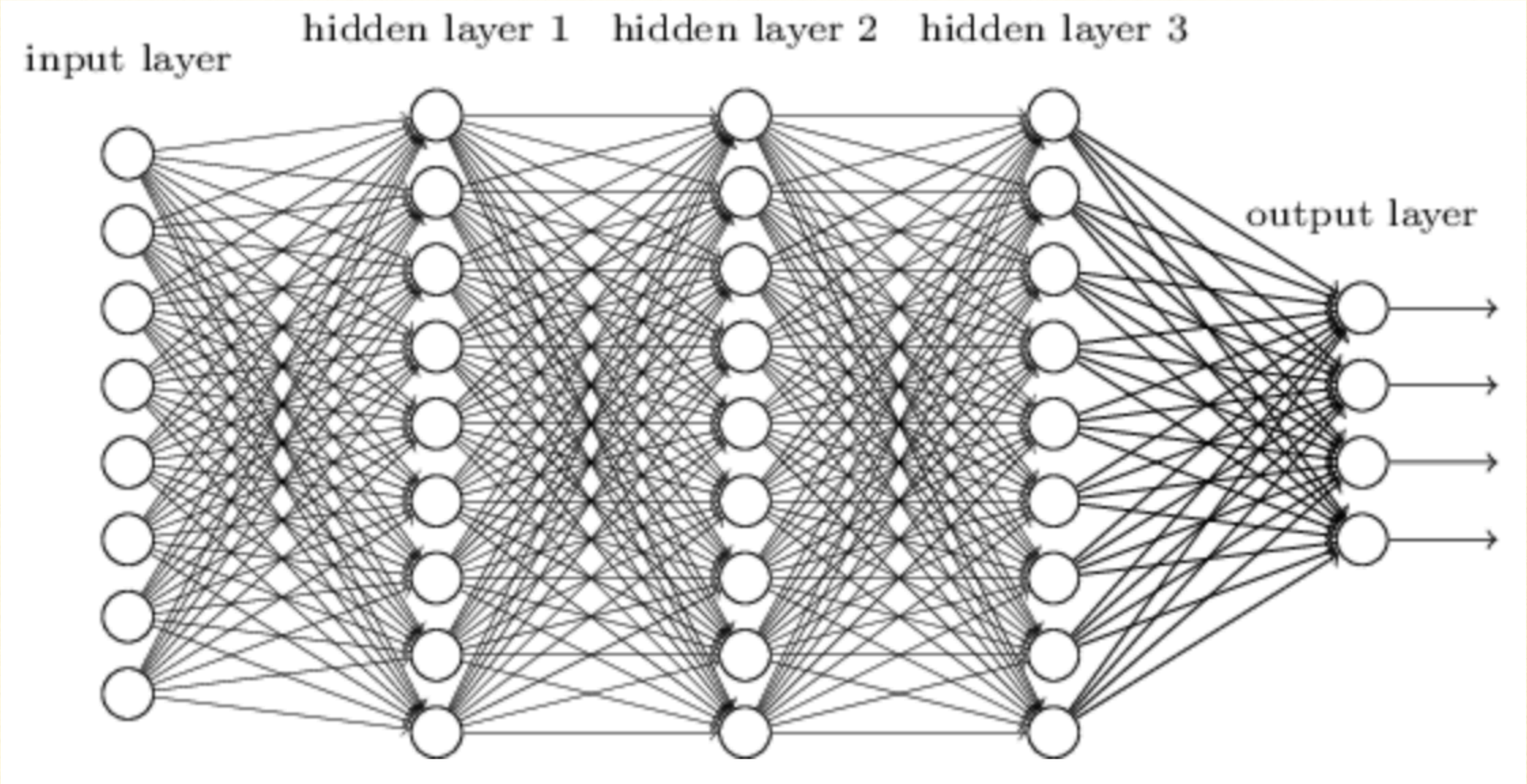


Image credit: M. Nielsen, *Neural networks and deep learning*

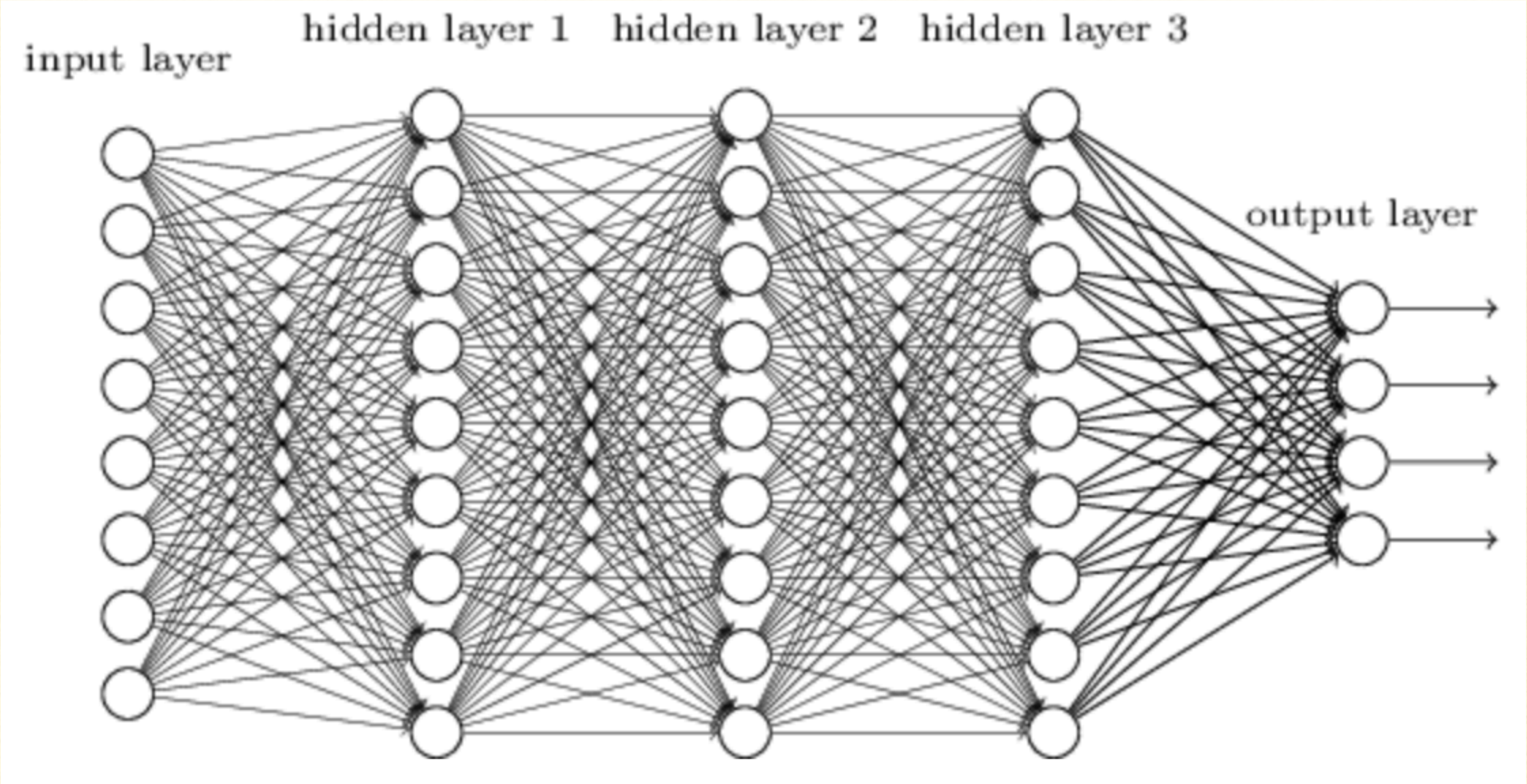
RELU NEURAL NETWORKS



$$\mathbb{R}^{n_0} \longrightarrow \mathbb{R}^{n_d}$$

Image credit: M. Nielsen, *Neural networks and deep learning*

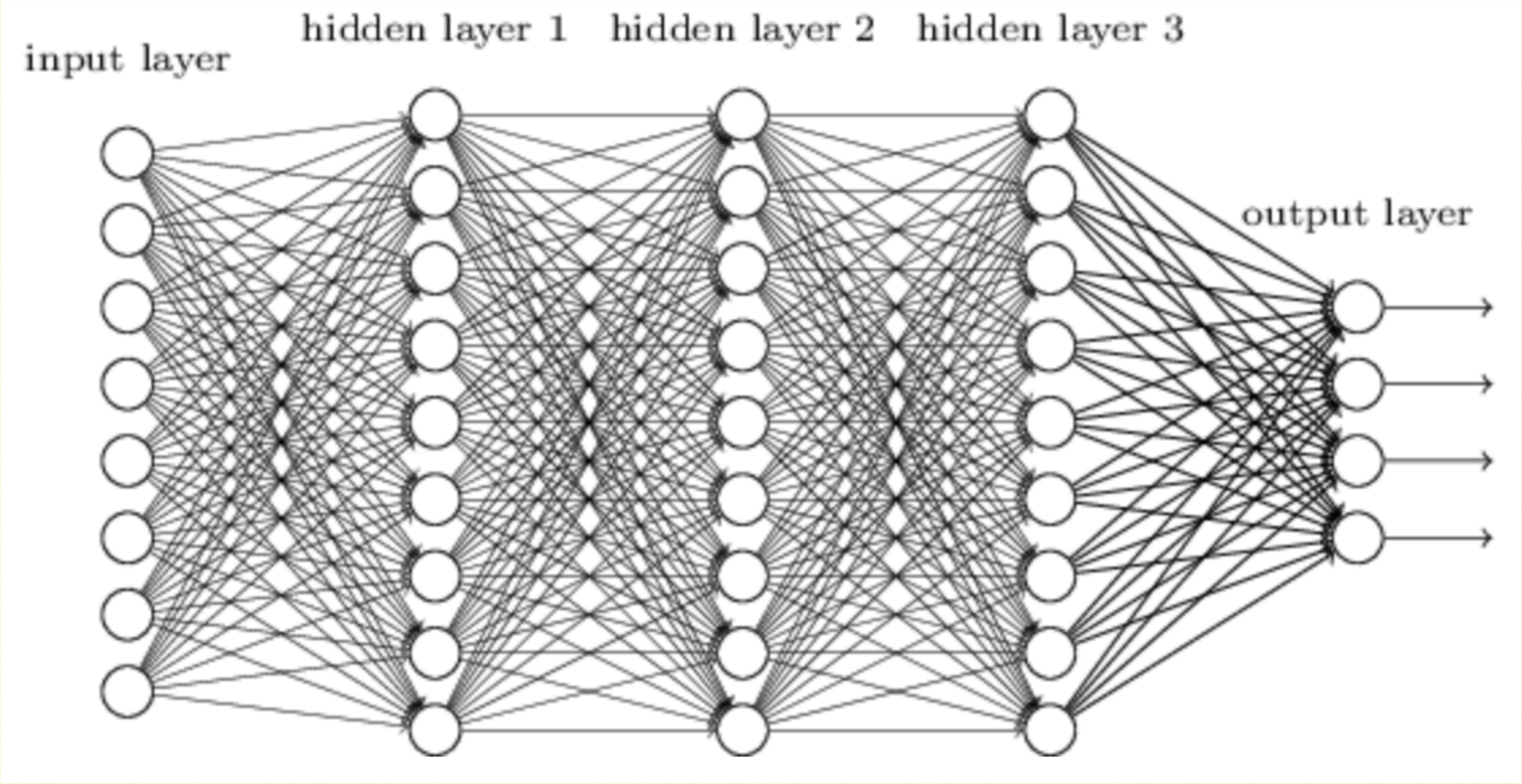
RELU NEURAL NETWORKS



$$\mathbb{R}^{n_0} \to \mathbb{R}^{n_1} \to \mathbb{R}^{n_2} \to \mathbb{R}^{n_3} \to \mathbb{R}^{n_d}$$

Image credit: M. Nielsen, *Neural networks and deep learning*

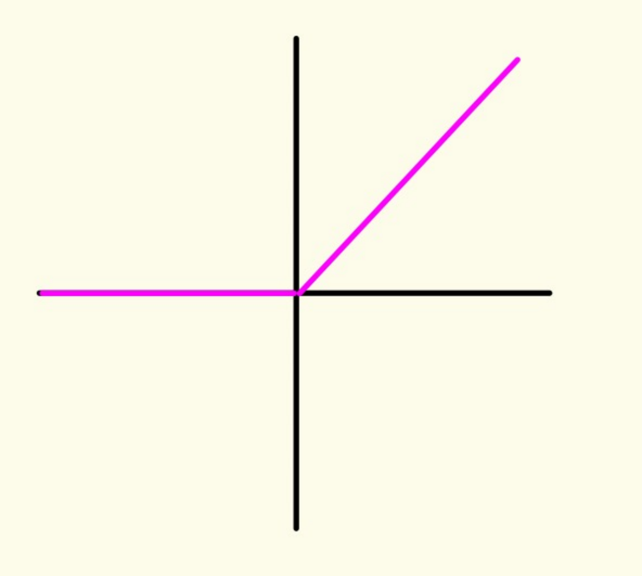
RELU NEURAL NETWORKS



$$\mathbb{R}^{n_0} \to \mathbb{R}^{n_1} \to \mathbb{R}^{n_2} \to \mathbb{R}^{n_3} \to \mathbb{R}^{n_d}$$

Architecture $(n_0, n_1, \ldots, n_d)$

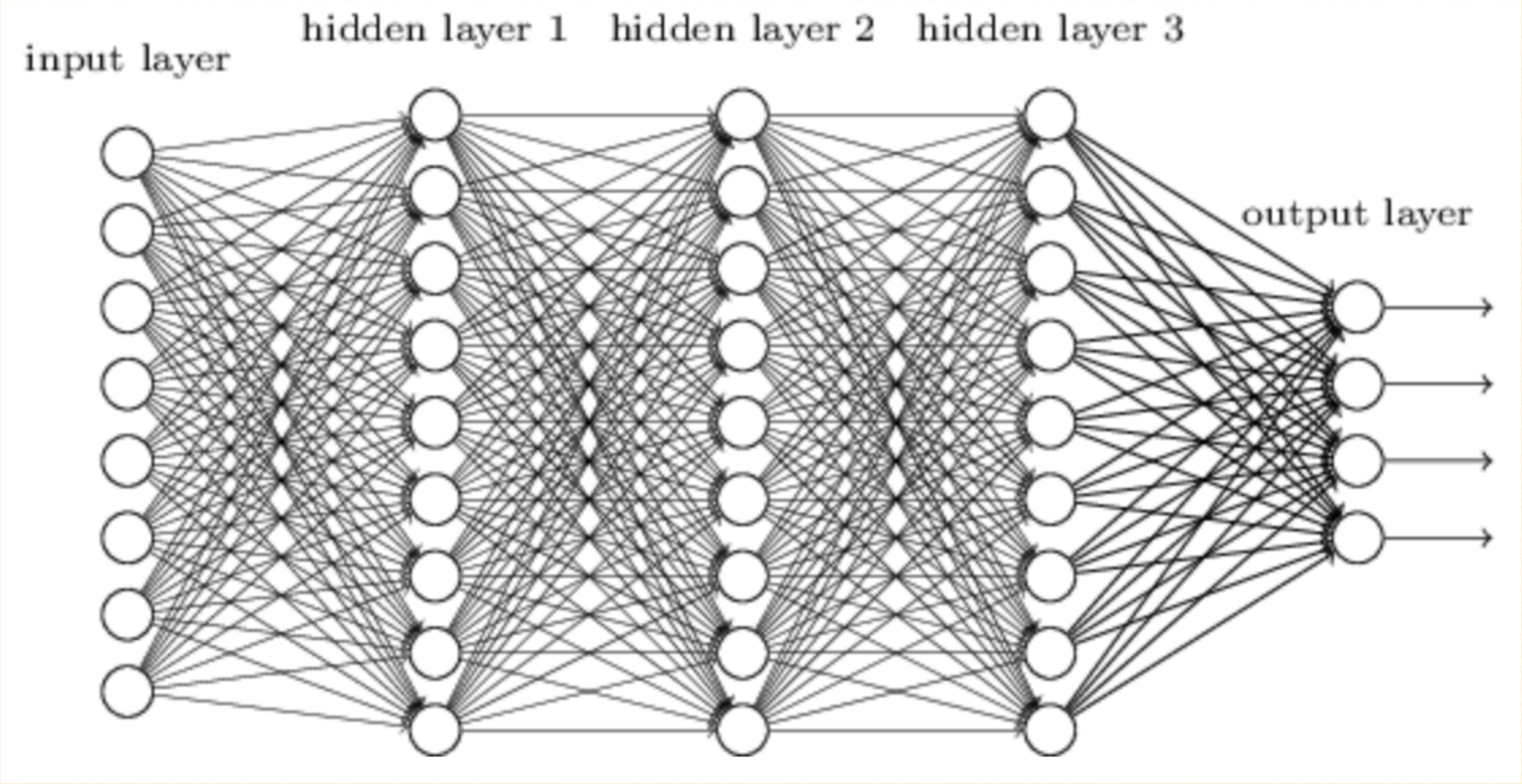
$$\text{ReLU}(x) := \max\{0, x\}$$



Modern activation function of choice

Image credit: M. Nielsen, *Neural networks and deep learning*

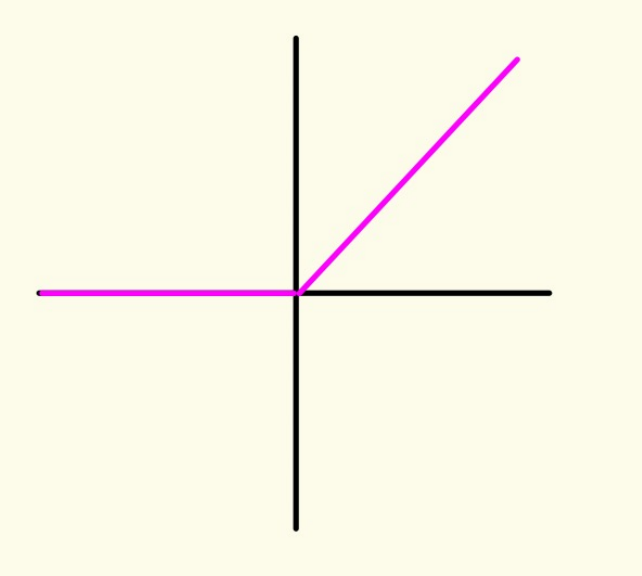
RELU NEURAL NETWORKS



$$\mathbb{R}^{n_0} \to \mathbb{R}^{n_1} \to \mathbb{R}^{n_2} \to \mathbb{R}^{n_3} \to \mathbb{R}^{n_d}$$

Architecture $(n_0, n_1, \ldots, n_d)$

ReLU$(x) := \max\{0, x\}$

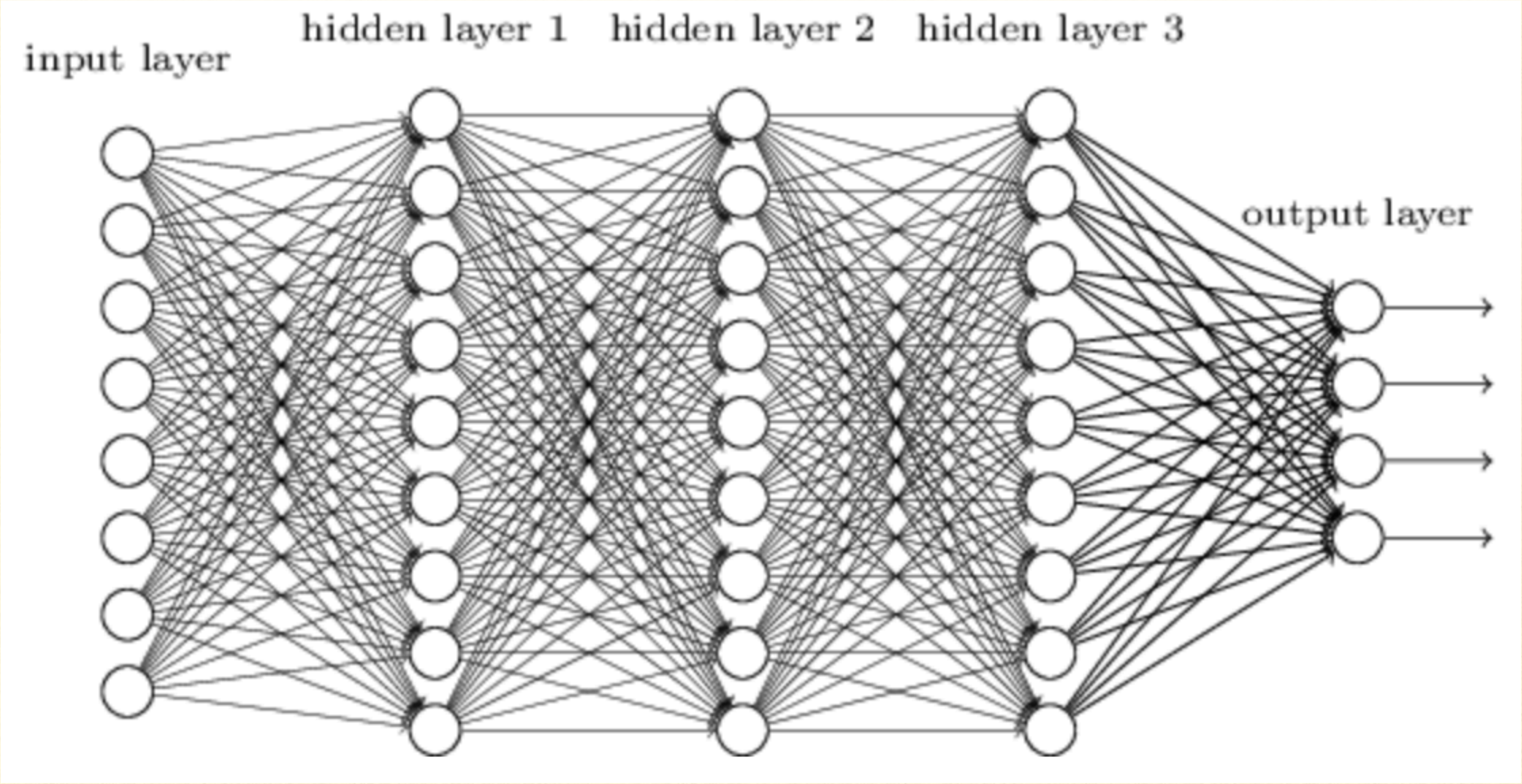


Modern activation function of choice

Arora-Basu-Mianjy-Mukherjee (ICLR '18):

Image credit: M. Nielsen, *Neural networks and deep learning*

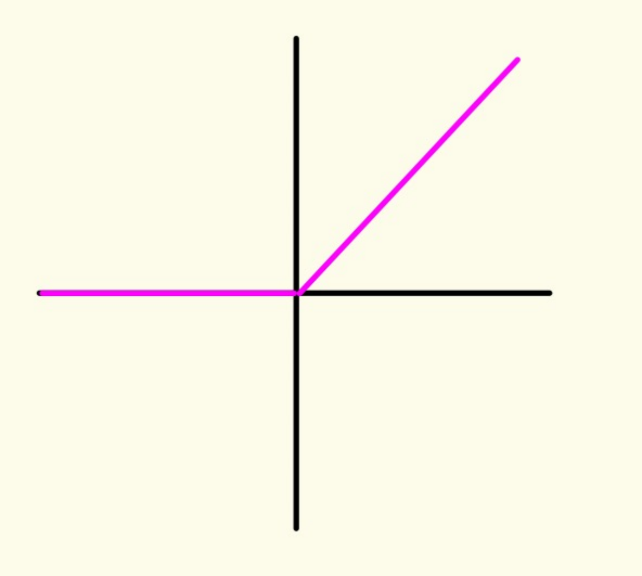
RELU NEURAL NETWORKS



$$\mathbb{R}^{n_0} \to \mathbb{R}^{n_1} \to \mathbb{R}^{n_2} \to \mathbb{R}^{n_3} \to \mathbb{R}^{n_d}$$

Architecture $(n_0, n_1, \ldots, n_d)$

$\mathrm{ReLU}(x) := \max\{0,x\}$



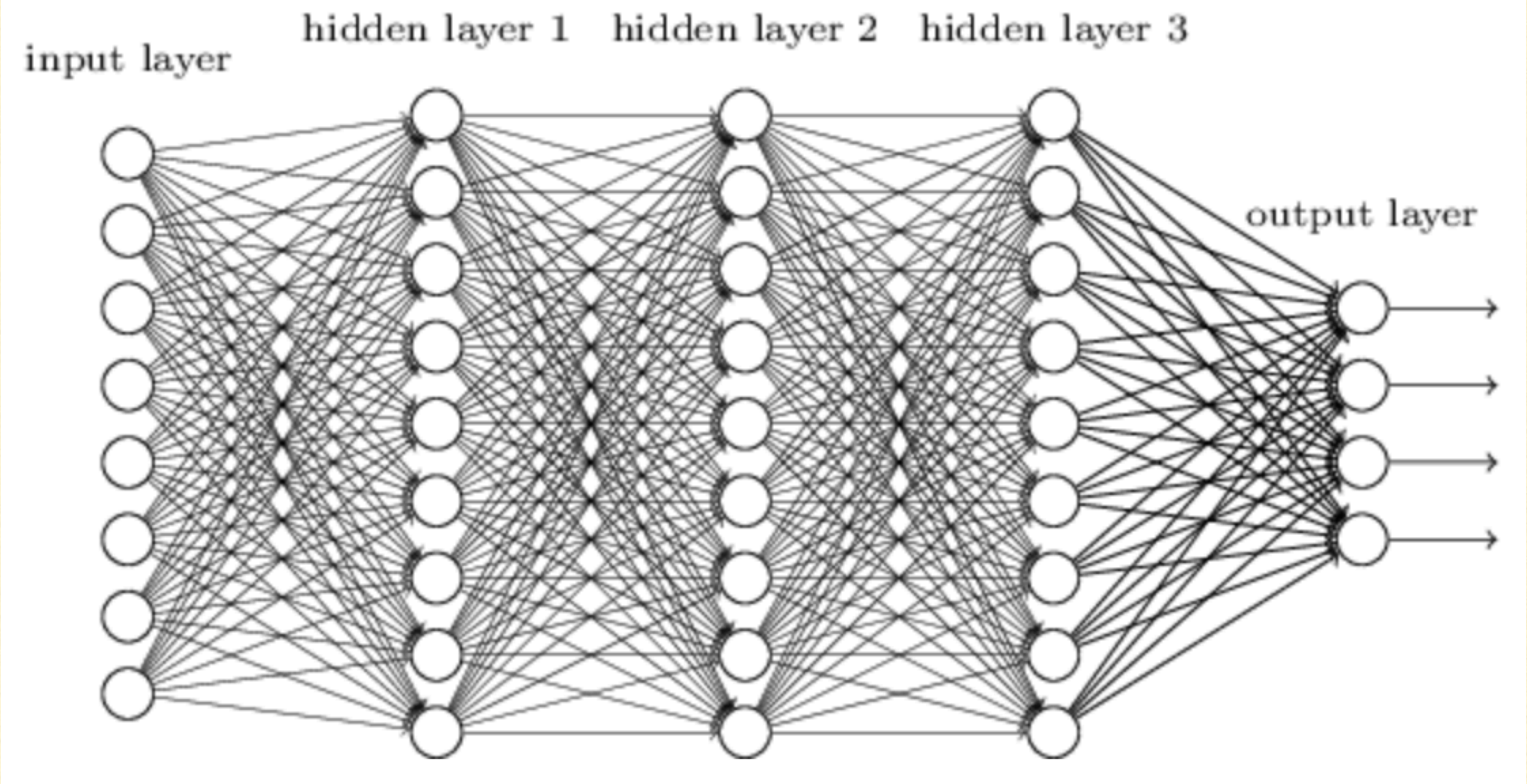
Modern activation function of choice

Arora-Basu-Mianjy-Mukherjee (ICLR '18):

$\boxed{\text{ReLU neural network functions}} = \boxed{\text{Finite piecewise-linear (PL) functions}}$

Image credit: M. Nielsen, *Neural networks and deep learning*

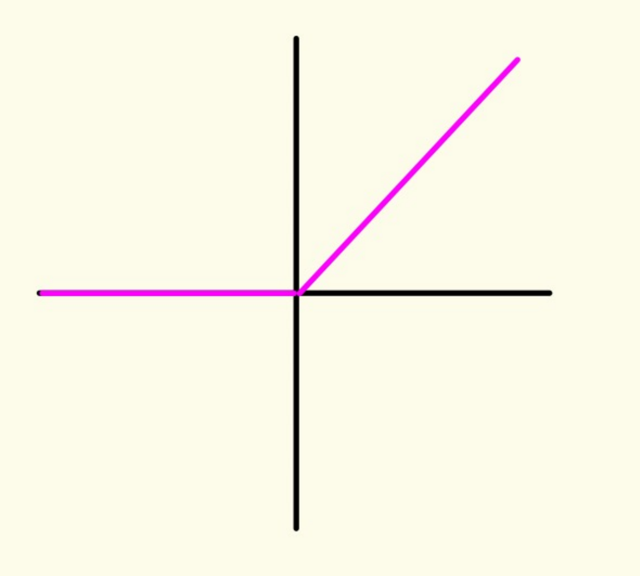
RELU NEURAL NETWORKS



$$\mathbb{R}^{n_0} \to \mathbb{R}^{n_1} \to \mathbb{R}^{n_2} \to \mathbb{R}^{n_3} \to \mathbb{R}^{n_d}$$

Architecture $(n_0, n_1, \ldots, n_d)$

ReLU$(x) := \max\{0, x\}$



Modern activation function of choice

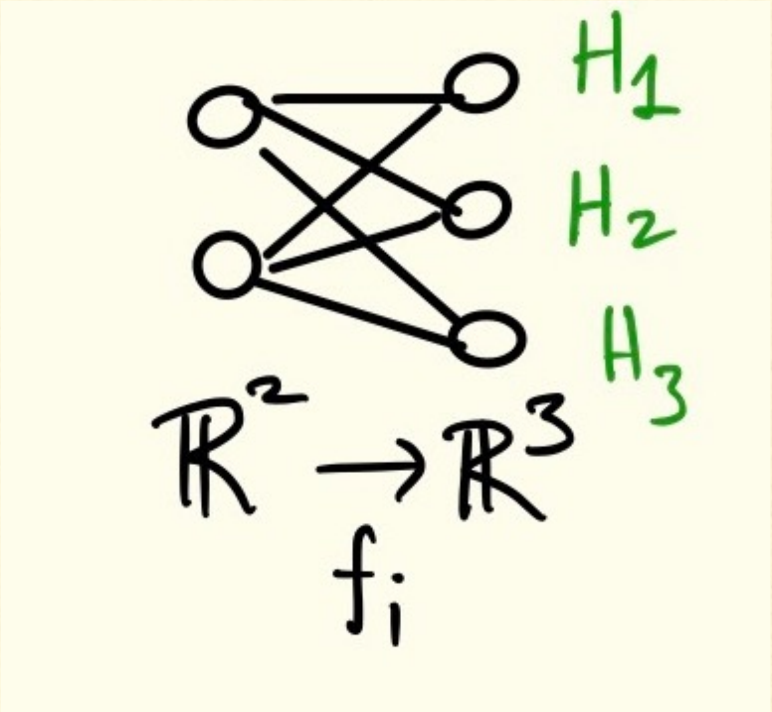
ReLU neural network functions of architecture $(n_0, n_1, \ldots, n_d)$ $\hookrightarrow$????? Finite piecewise-linear (PL) functions

Image credit: M. Nielsen, *Neural networks and deep learning*

RELU NEURAL NETWORKS

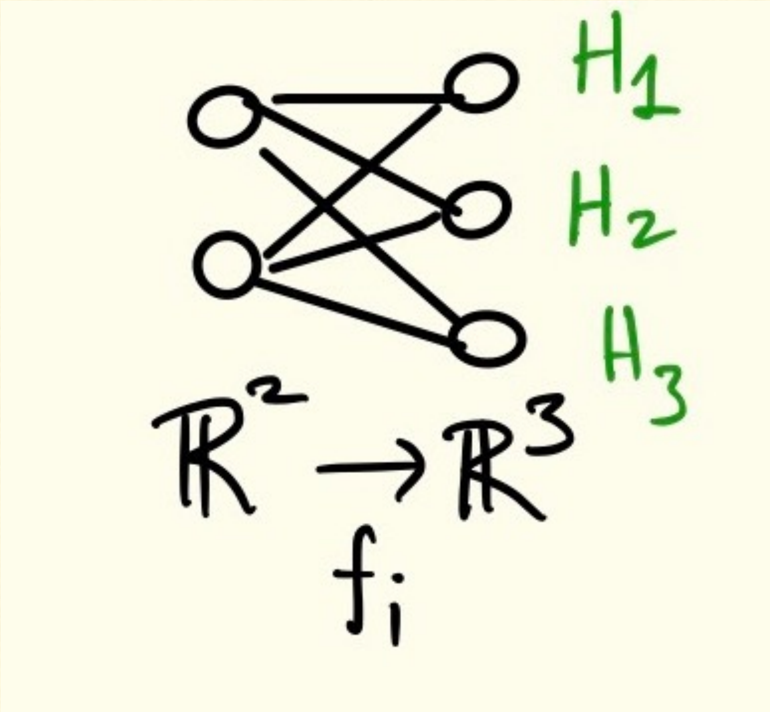
RELU NEURAL NETWORKS

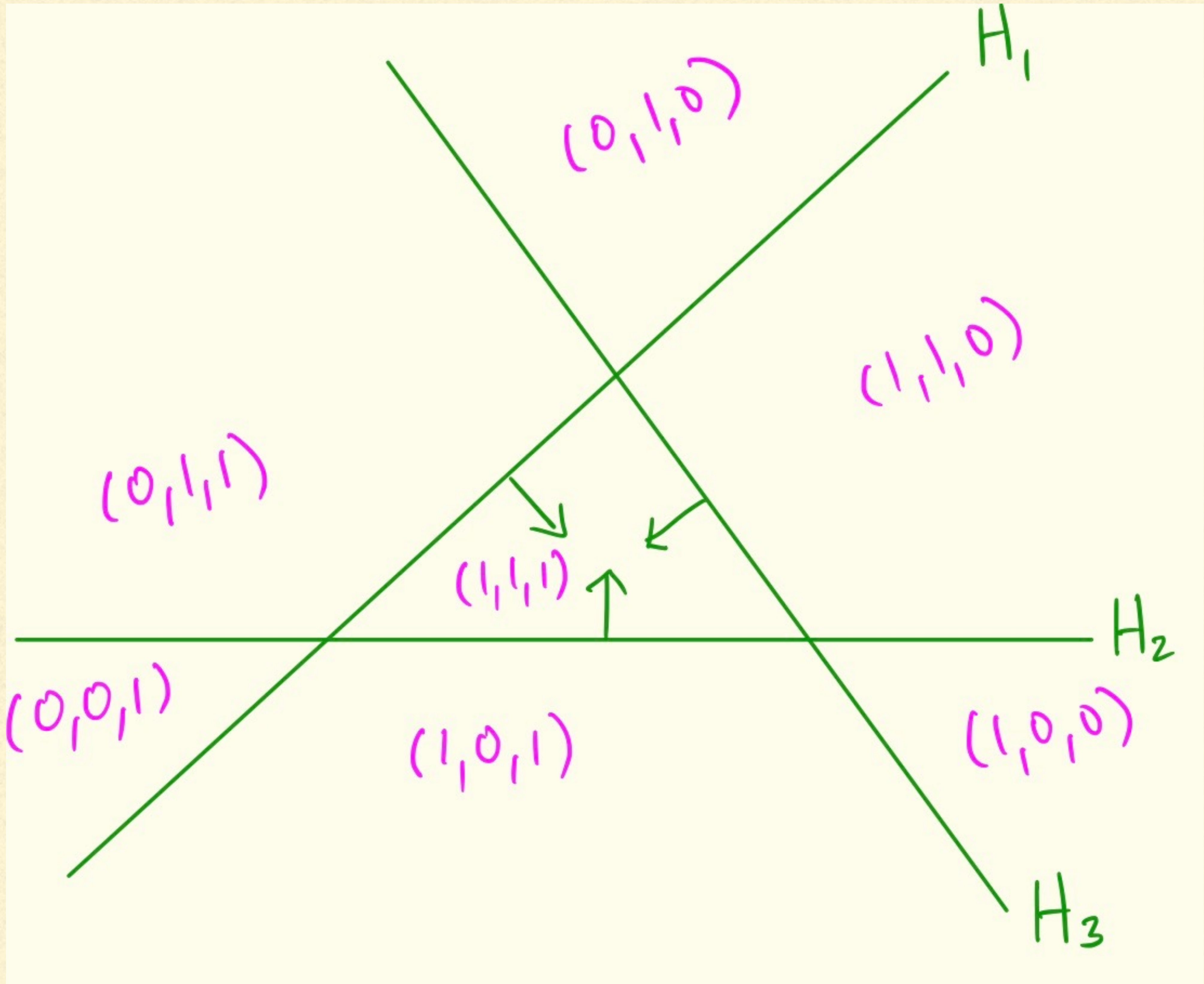
$$F_i = \sigma \circ A_i : \mathbb{R}^{n_{i-1}} \to \mathbb{R}^{n_i}$$



RELU NEURAL NETWORKS

$$F_i = \sigma \circ A_i : \mathbb{R}^{n_{i-1}} \to \mathbb{R}^{n_i}$$

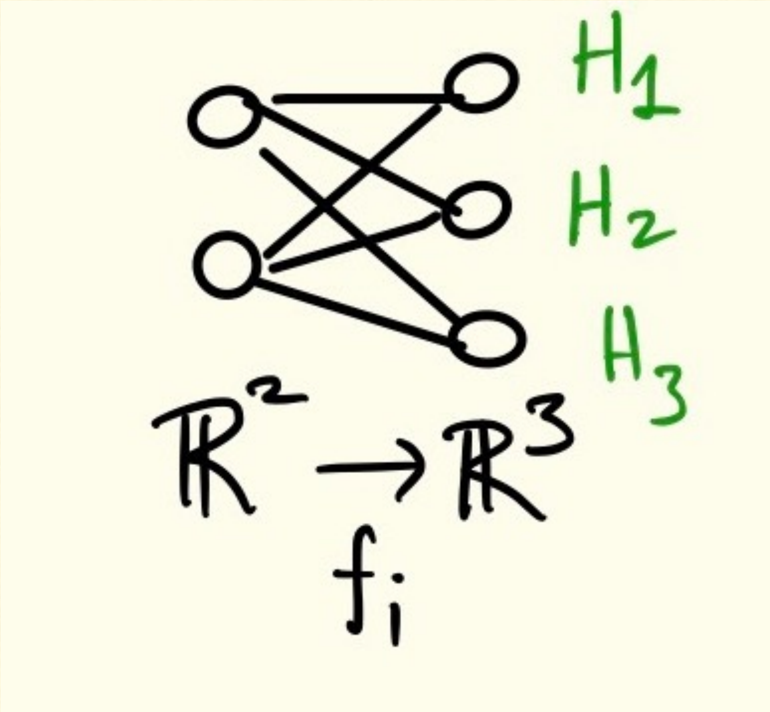


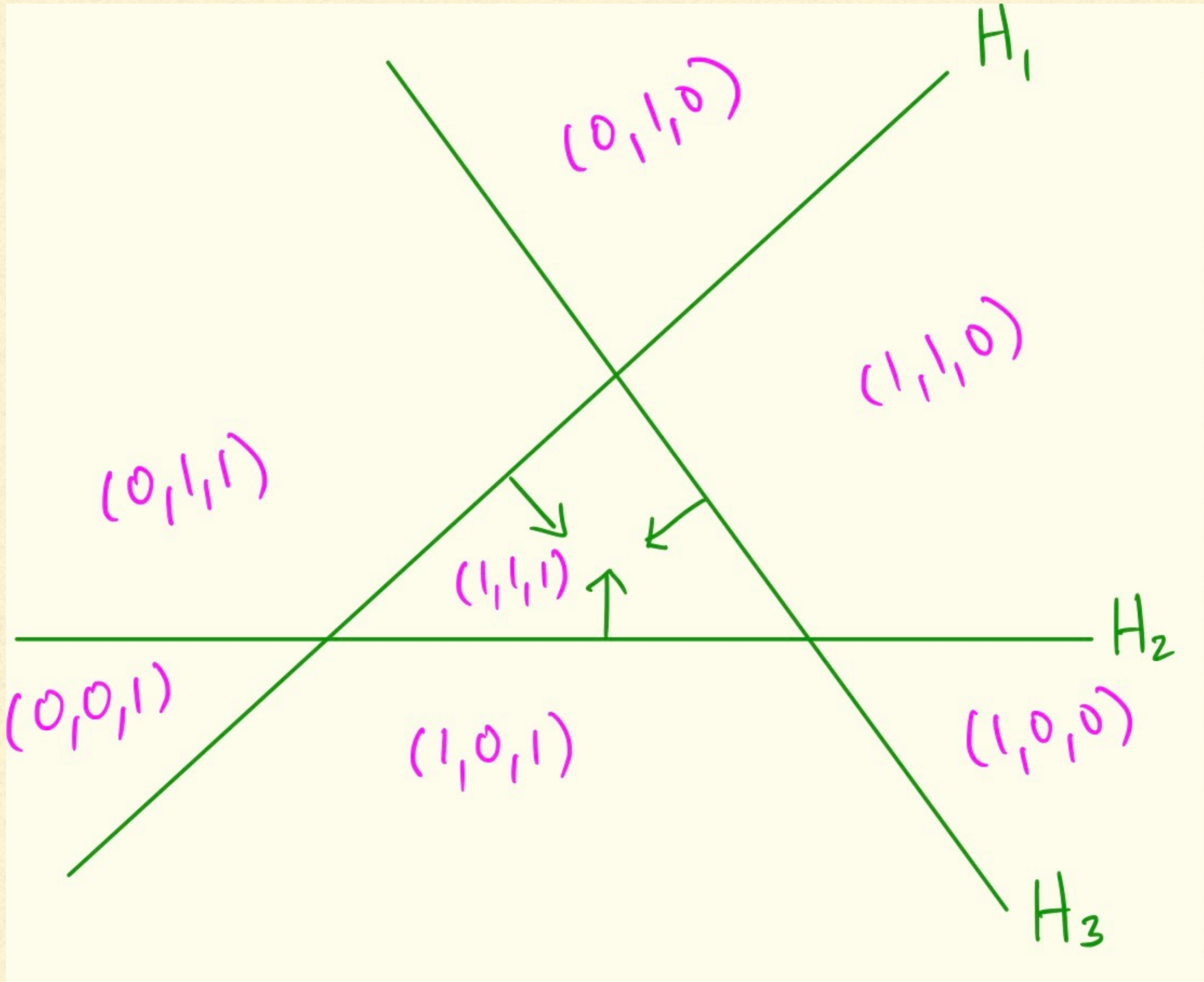


Co-oriented hyperplane arrangement

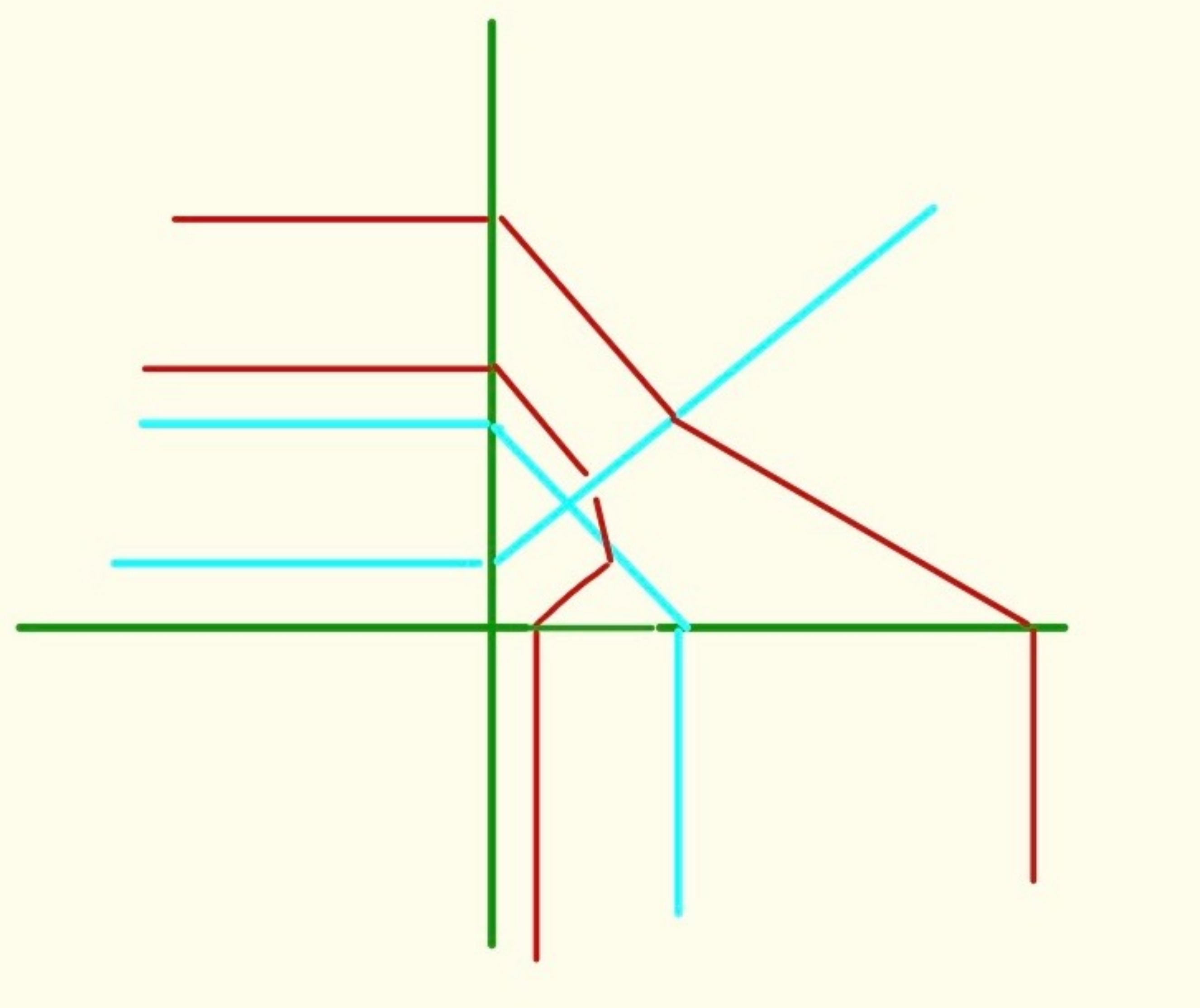
RELU NEURAL NETWORKS

$$F_i = \sigma \circ A_i : \mathbb{R}^{n_{i-1}} \to \mathbb{R}^{n_i}$$





Co-oriented hyperplane arrangement



"Bent" hyperplane arrangement

RELU NEURAL NETWORKS

RELU NEURAL NETWORKS

- For any fixed architecture of depth at least 2, parameter space is a highly **redundant** and **inhomogeneous** proxy for the true hypothesis class

RELU NEURAL NETWORKS

- For any fixed architecture of depth at least 2, parameter space is a highly **redundant** and **inhomogeneous** proxy for the true hypothesis class
- I believe this is a feature, not a bug

OUTLINE:

1. Parameter space $\neq$ Function space for ReLU networks
2. (Effective) functional dimension
3. Theoretical and experimental results

OUTLINE:

1. Parameter space $\neq$ Function space for ReLU networks

2. (Effective) functional dimension

3. Theoretical and experimental results

PARAMETER SPACE SYMMETRIES FOR RELU NETWORKS

PARAMETER SPACE SYMMETRIES FOR RELU NETWORKS

Parameter Space for Architecture

$(n_0, n_1, \ldots, n_{d-1}, n_d)$

PARAMETER SPACE SYMMETRIES FOR RELU NETWORKS

Parameter Space for Architecture

$(n_0, n_1, \ldots, n_{d-1}, n_d)$

$$\parallel$$

$$\mathbb{R}^D$$

where $\displaystyle D = \sum_{i=0}^{d-1} (n_i + 1) n_{i+1}$

PARAMETER SPACE SYMMETRIES FOR RELU NETWORKS

Parameter Space for Architecture

$(n_0, n_1, \ldots, n_{d-1}, n_d)$

$$\neq$$

Function Space for Architecture

$(n_0, n_1, \ldots, n_{d-1}, n_d)$

$$\|$$

$$\mathbb{R}^D$$

where $\quad D = \sum_{i=0}^{d-1} (n_i + 1)n_{i+1}$

PARAMETER SPACE SYMMETRIES FOR RELU NETWORKS

Parameter Space for Architecture

$(n_0, n_1, \ldots, n_{d-1}, n_d)$

$$\parallel$$

$$\mathbb{R}^D$$

$$\neq$$

Function Space for Architecture

$(n_0, n_1, \ldots, n_{d-1}, n_d)$

$$\subseteq$$

$$\mathrm{PL}(\mathbb{R}^{n_0} \to \mathbb{R}^{n_d})$$

where $D = \sum_{i=0}^{d-1} (n_i + 1) n_{i+1}$

PARAMETER SPACE SYMMETRIES FOR RELU NETWORKS

Have a realization map

PARAMETER SPACE SYMMETRIES FOR RELU NETWORKS

Have a realization map

$$\rho : \mathbb{R}^D \to \mathrm{PL}(\mathbb{R}^{n_0} \to \mathbb{R}^{n_d})$$

PARAMETER SPACE SYMMETRIES FOR RELU NETWORKS

Have a realization map

$$\rho : \mathbb{R}^D \to \mathrm{PL}(\mathbb{R}^{n_0} \to \mathbb{R}^{n_d})$$

Parameter Space

Function Space

PARAMETER SPACE SYMMETRIES FOR RELU NETWORKS

Have a realization map

$$\rho : \mathbb{R}^D \to \text{PL}(\mathbb{R}^{n_0} \to \mathbb{R}^{n_d})$$

Parameter Space

Function Space

- Not injective ("Many to one")

PARAMETER SPACE SYMMETRIES FOR RELU NETWORKS

Have a realization map

$$\rho : \mathbb{R}^D \to \mathrm{PL}(\mathbb{R}^{n_0} \to \mathbb{R}^{n_d})$$

Parameter Space

Function Space

- Not injective ("Many to one")
- Positive-dimensional spaces of symmetries

HISTORY/RELATED WORK:

- **Fefferman-Markel, Albertini-Sontag (1990's):** Parameters of a multilayer perceptron with sigmoidal activation can be recovered up to finite known symmetries

- **Armenta-Jodoin, et al. ('18):** *Quiver representation theory* (framework for understanding moduli spaces and global symmetries in general (for arbitrary activation functions)

- **Kording-Rolnick, Phuong-Lampert ('20):** For ReLU networks, give geometric conditions under which parameters are obtainable up to known global symmetries

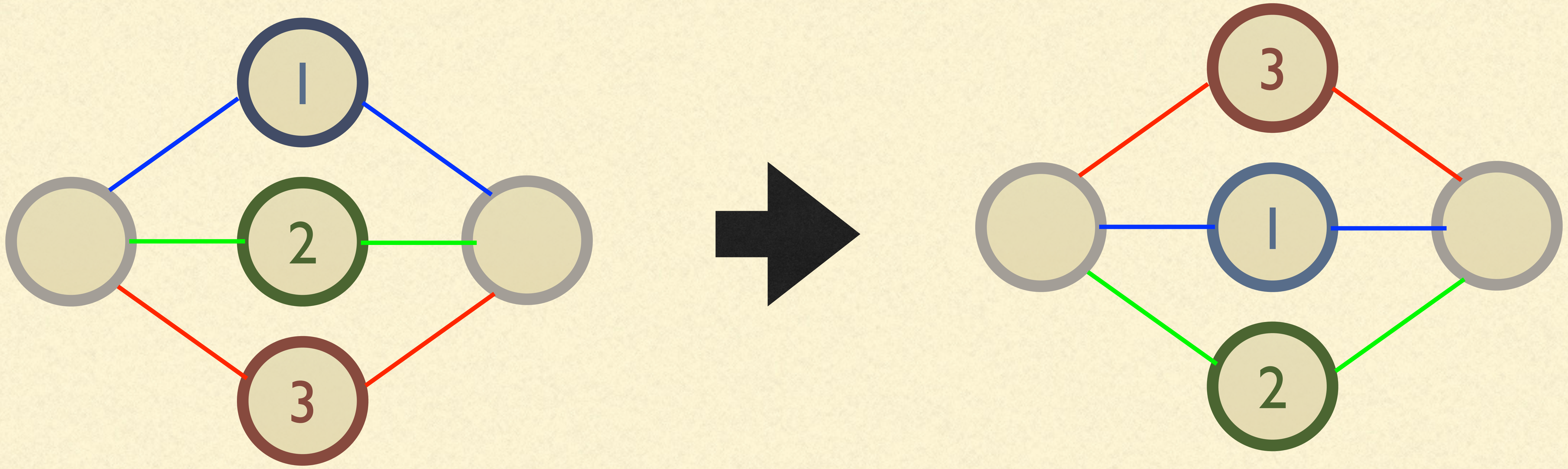
WELL-KNOWN GLOBALLY-DEFINED SYMMETRIES (*CF. KORDING-ROLNICK, PHUONG-LAMPERT*):

WELL-KNOWN GLOBALLY-DEFINED SYMMETRIES (*CF. KORDING-ROLNICK, PHUONG-LAMPERT*):

PERMUTATION:

WELL-KNOWN GLOBALLY-DEFINED SYMMETRIES *(CF. KORDING-ROLNICK, PHUONG-LAMPERT)*:

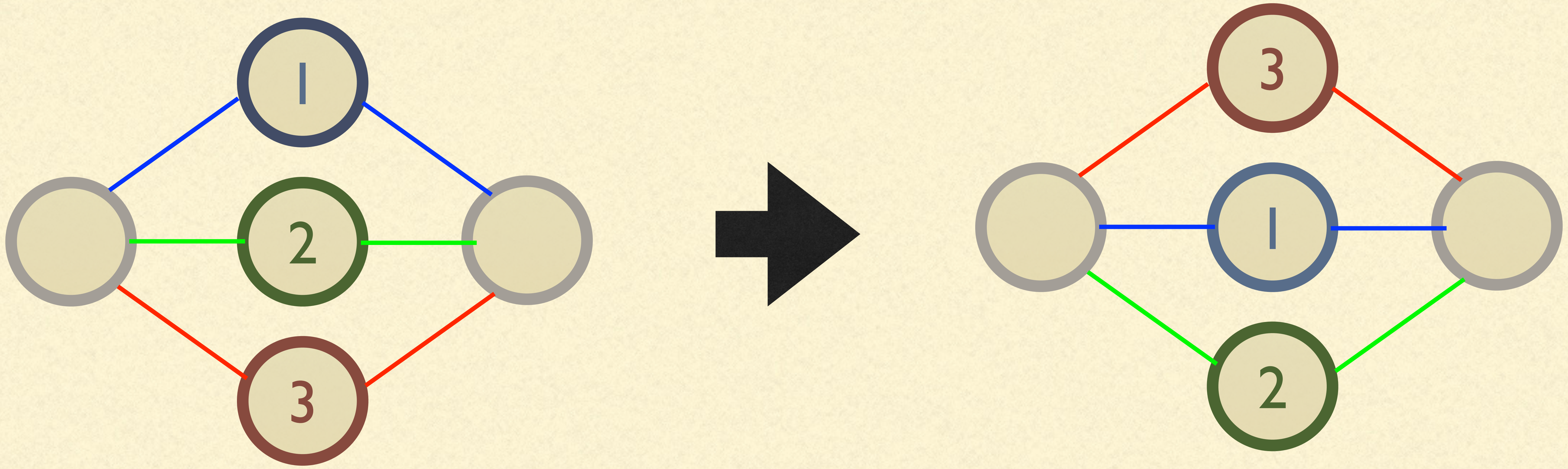
PERMUTATION:



WELL-KNOWN GLOBALLY-DEFINED SYMMETRIES *(CF. KORDING-ROLNICK, PHUONG-LAMPERT)*:

PERMUTATION:

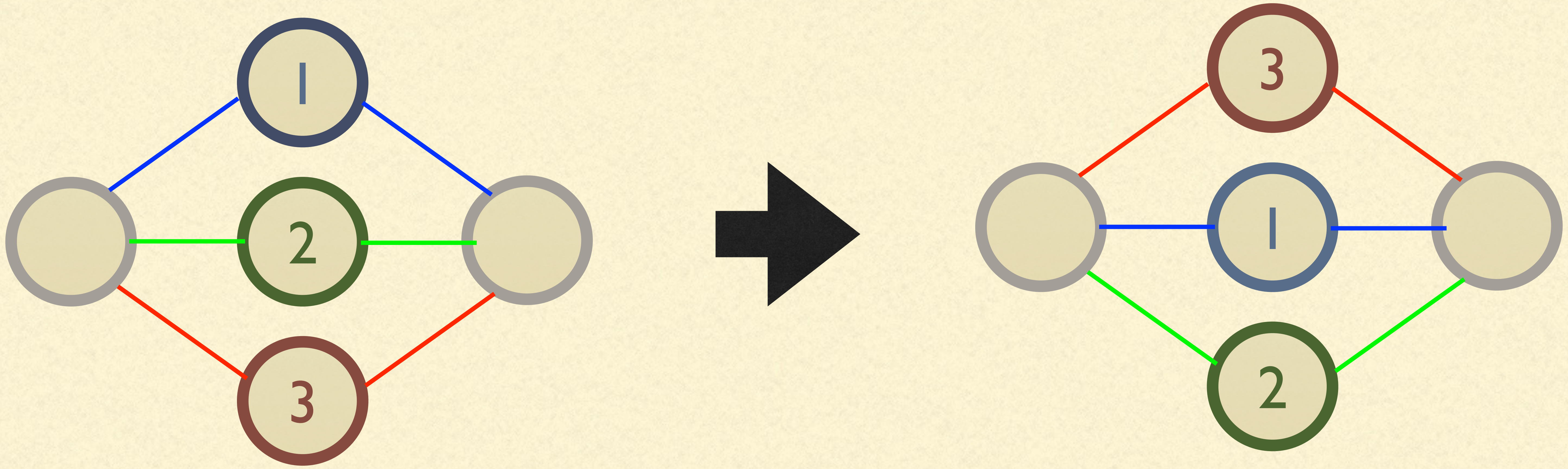
Discrete, aka
0-dimensional



WELL-KNOWN GLOBALLY-DEFINED SYMMETRIES *(CF. KORDING-ROLNICK, PHUONG-LAMPERT)*:

PERMUTATION:

Discrete, aka
0-dimensional

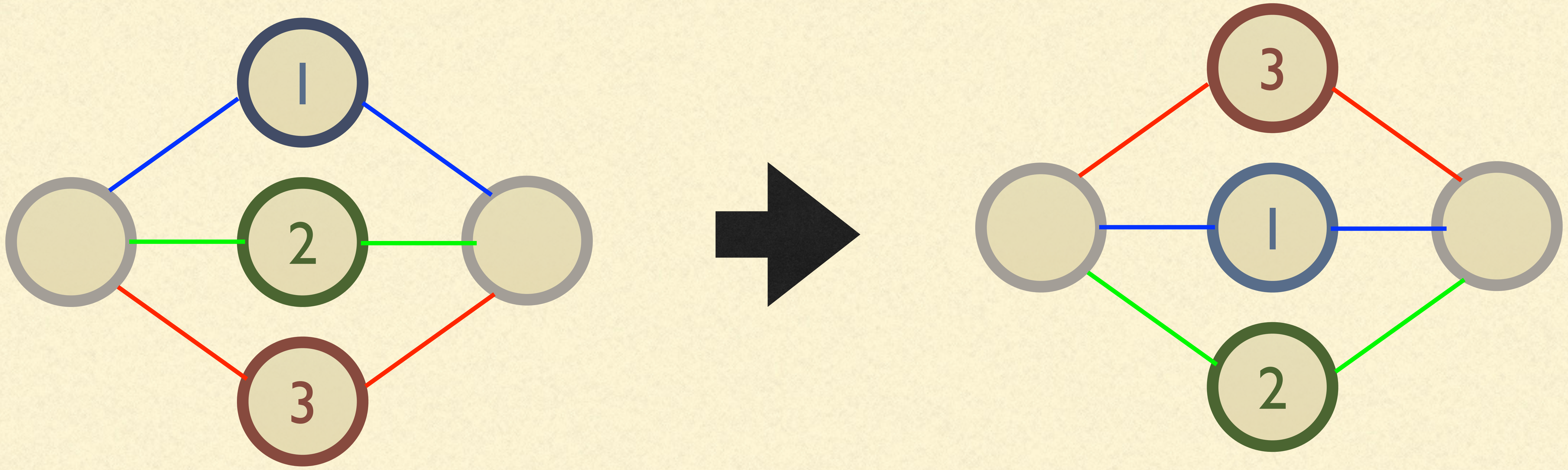


POSITIVE SCALING:

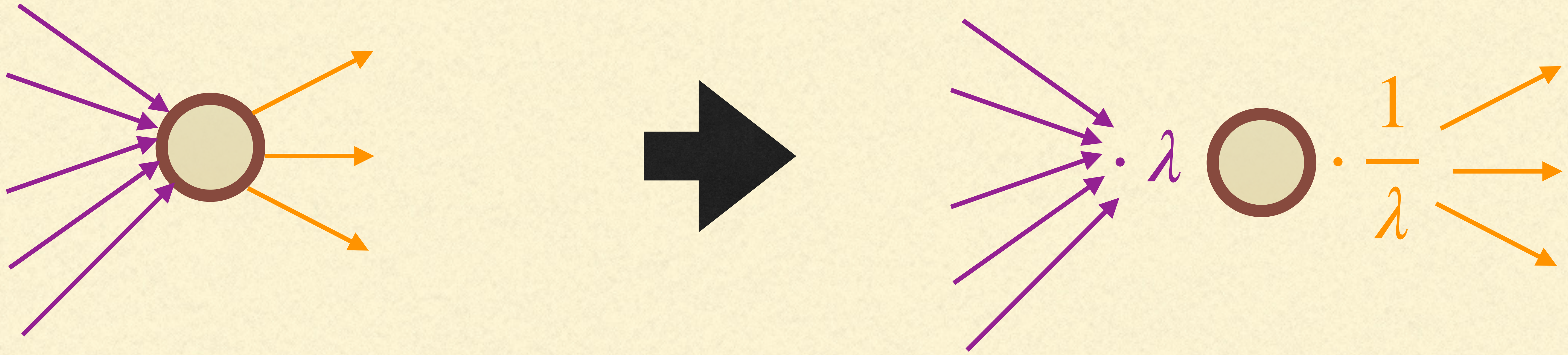
WELL-KNOWN GLOBALLY-DEFINED SYMMETRIES *(CF. KORDING-ROLNICK, PHUONG-LAMPERT)*:

PERMUTATION:

Discrete, aka
0-dimensional



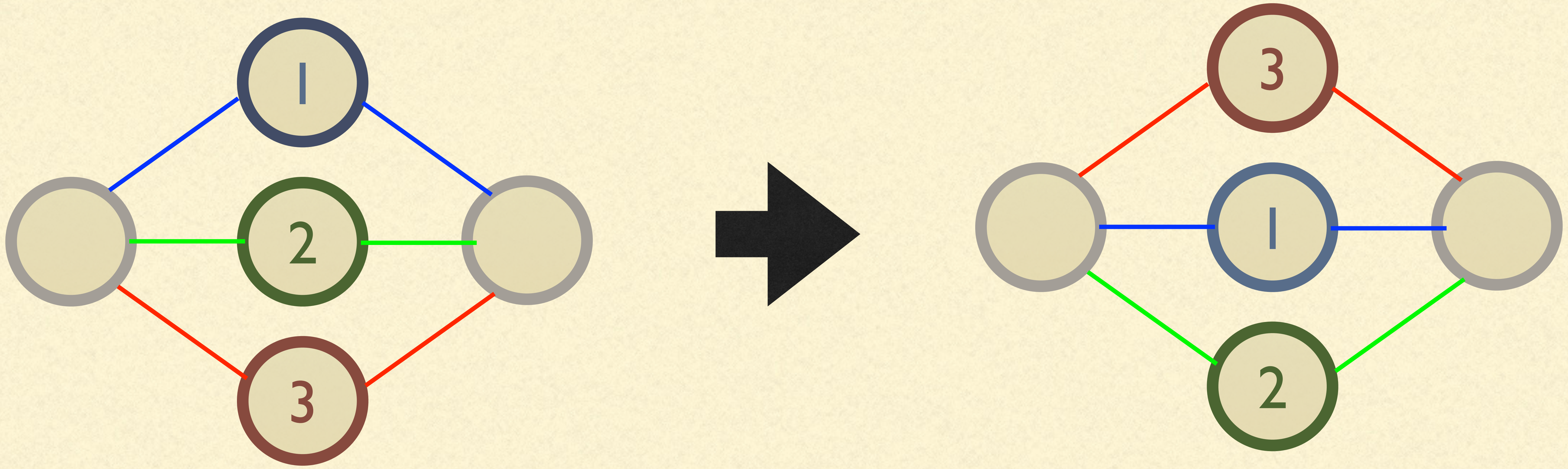
POSITIVE SCALING:



WELL-KNOWN GLOBALLY-DEFINED SYMMETRIES *(CF. KORDING-ROLNICK, PHUONG-LAMPERT)*:

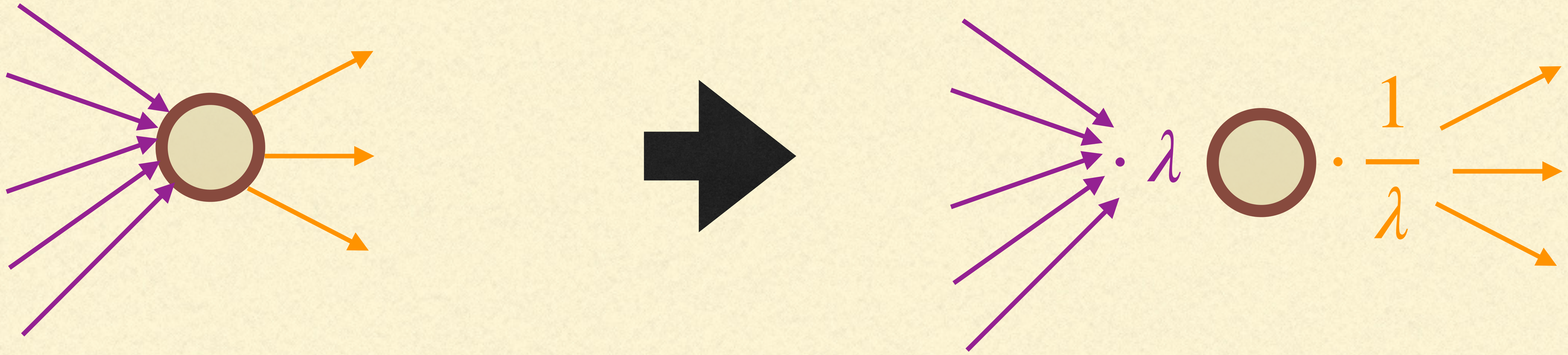
PERMUTATION:

Discrete, aka
0-dimensional



POSITIVE SCALING:

Positive-
dimensional



EASY CONSEQUENCE:

EASY CONSEQUENCE:

Lemma: The function space of a ReLU network of architecture $(n_0, \ldots, n_d)$ has dimension at most

$$D' := \sum_{i=0}^{d-1} (n_i + 1)n_{i+1} - \sum_{i=1}^{d-1} n_i$$

EASY CONSEQUENCE:

Lemma: The function space of a ReLU network of architecture $(n_0, \ldots, n_d)$ has dimension at most

$$D' := \underbrace{\sum_{i=0}^{d-1} (n_i + 1) n_{i+1}}_{D} - \underbrace{\sum_{i=1}^{d-1} n_i}$$

(Parametric dimension) (# of hidden neurons)

EASY CONSEQUENCE:

Lemma: The function space of a ReLU network of architecture $(n_0, \ldots, n_d)$ has dimension at most

$$D' := \underbrace{\sum_{i=0}^{d-1} (n_i + 1)n_{i+1}}_{D} - \underbrace{\sum_{i=1}^{d-1} n_i}$$

Theoretical upper bound on functional dimension

(Parametric dimension)

(# of hidden neurons)

OUTLINE:

1. Parameter space $\neq$ Function space for ReLU networks

2. (Effective) functional dimension

3. Theoretical and experimental results

FUNCTIONAL DIMENSION

FUNCTIONAL DIMENSION

Local, near a parameter

FUNCTIONAL DIMENSION

Local, near a parameter

$$\theta_0 \in (\Omega = \mathbb{R}^D)$$

FUNCTIONAL DIMENSION

Local, near a parameter

$$\theta_0 \in (\Omega = \mathbb{R}^D) \xrightarrow{\text{Gives rise to}} F_{\theta_0} : \mathbb{R}^{n_0} \to \mathbb{R}^{n_d}$$

FUNCTIONAL DIMENSION

Local, near a parameter

$$\theta_0 \in (\Omega = \mathbb{R}^D) \xrightarrow{\text{Gives rise to}} F_{\theta_0} : \mathbb{R}^{n_0} \to \mathbb{R}^{n_d}$$

Fix

$$Z = \{z_1, \ldots, z_N\} \in \mathbb{R}^{n_0}$$

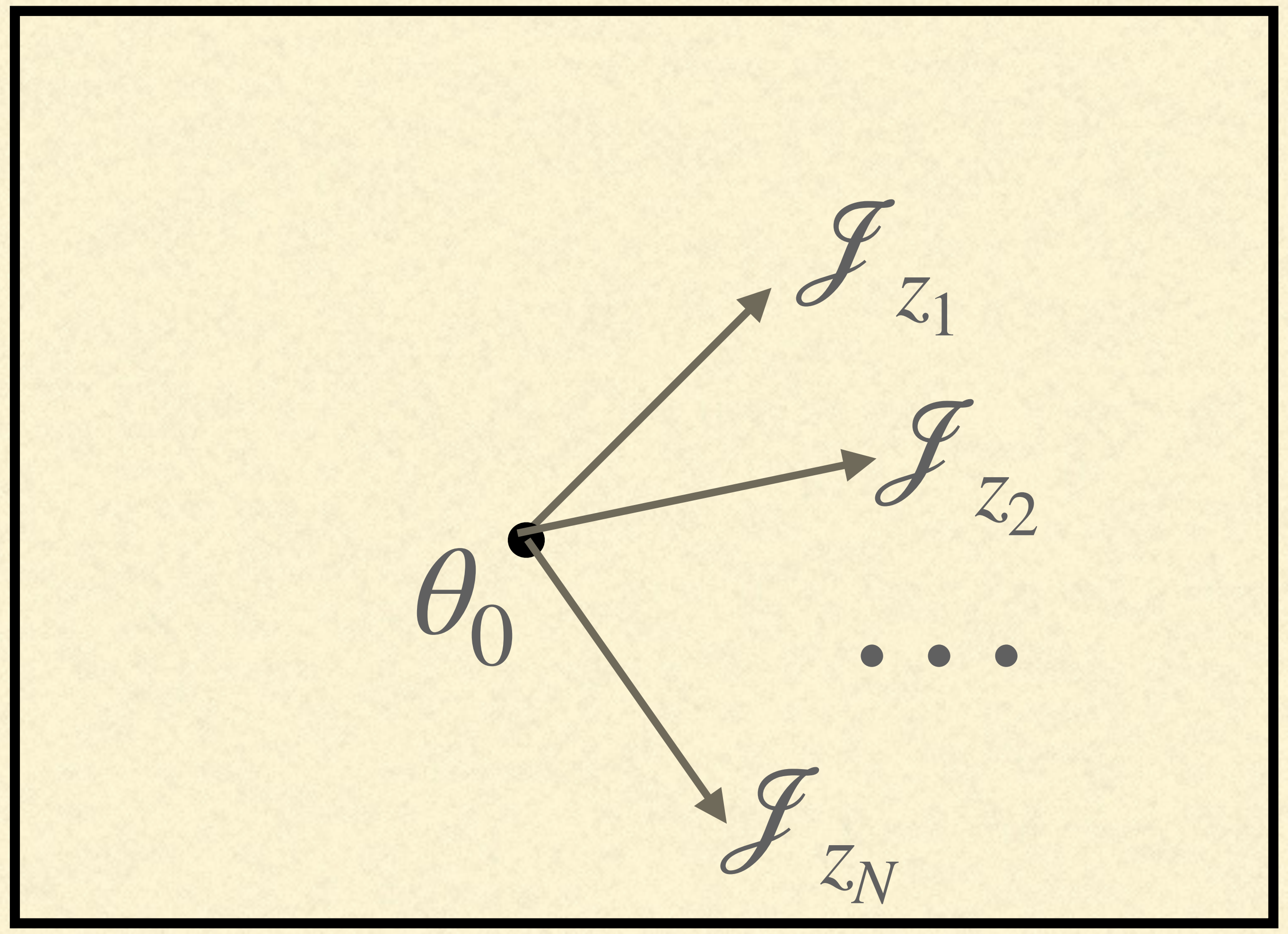
FUNCTIONAL DIMENSION

Local, near a parameter

$$\theta_0 \in (\Omega = \mathbb{R}^D) \xrightarrow{\text{Gives rise to}} F_{\theta_0} : \mathbb{R}^{n_0} \to \mathbb{R}^{n_d}$$

Fix

$$Z = \{z_1, \ldots, z_N\} \in \mathbb{R}^{n_0}$$



$T_{\theta_0}\left(\mathbb{R}^D\right)$: Tangent space at θ_0

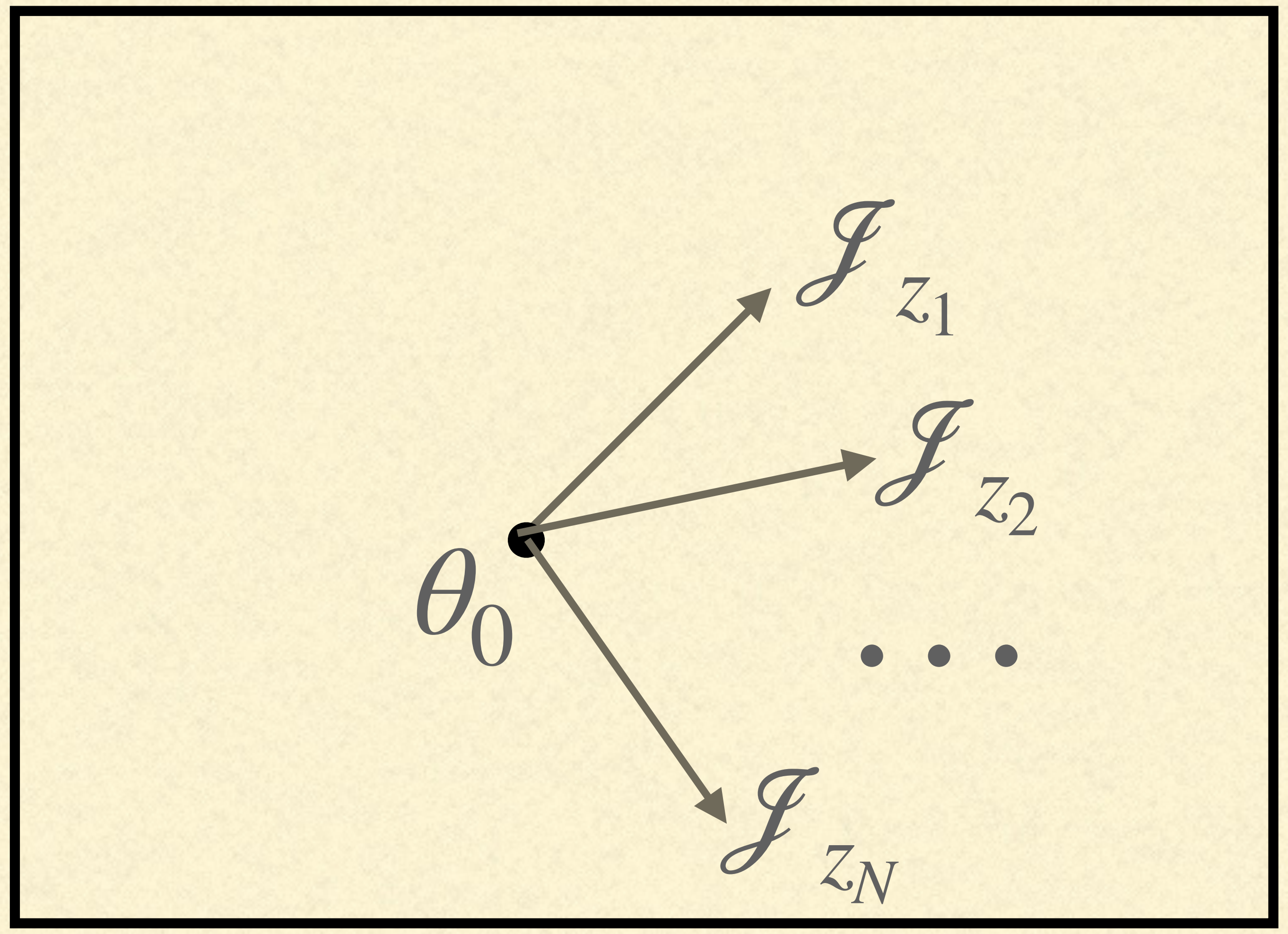
FUNCTIONAL DIMENSION

Local, near a parameter

$$\theta_0 \in (\Omega = \mathbb{R}^D) \xrightarrow{\text{Gives rise to}} F_{\theta_0} : \mathbb{R}^{n_0} \to \mathbb{R}^{n_d}$$

Fix

$$Z = \{z_1, \ldots, z_N\} \in \mathbb{R}^{n_0}$$



$T_{\theta_0}\left(\mathbb{R}^D\right)$: Tangent space at θ_0

$\mathcal{J}_Z$: (Span of) directions in which we can perturb near θ_0 to change the value of F_θ for at least one point in Z

FORMALLY:

FORMALLY:

Fix $Z = \{z_1, \ldots, z_k\} \subseteq \mathbb{R}^{n_0}$

FORMALLY:

Fix $Z = \{z_1, \ldots, z_k\} \subseteq \mathbb{R}^{n_0}$

$$\mathrm{Ev}_Z : \mathbb{R}^D \longrightarrow M_{k \times n_d}$$

FORMALLY:

Fix $Z = \{z_1, \ldots, z_k\} \subseteq \mathbb{R}^{n_0}$

$$\begin{array}{ccc} \mathrm{Ev}_Z : \mathbb{R}^D & \longrightarrow & M_{k \times n_d} \\ \rotatebox{90}{$\in$} & & \\ \theta & & \begin{pmatrix} F_\theta(z_1) \\ \vdots \\ F_\theta(z_k) \end{pmatrix} \end{array}$$

FORMALLY:

Fix $Z = \{z_1, \ldots, z_k\} \subseteq \mathbb{R}^{n_0}$

$$
\begin{tikzcd}
\mathrm{Ev}_Z : \mathbb{R}^D \arrow[r] & M_{k\times n_d} \arrow[r] & \mathbb{R}^{kn_d} \\
\rotatebox{90}{$\in$} & & \\
\theta & \begin{pmatrix} F_\theta(z_1) \\ \vdots \\ F_\theta(z_k) \end{pmatrix} \arrow[ruu, bend right, "\text{Vectorize (unroll)}"'] &
\end{tikzcd}
$$

FORMALLY:

Fix $Z = \{z_1, \ldots, z_k\} \subseteq \mathbb{R}^{n_0}$

$$\begin{tikzcd}[column sep=huge] \mathrm{Ev}_Z : \mathbb{R}^D \arrow[r] & M_{k \times n_d} \arrow[r] & \mathbb{R}^{kn_d} \\ \theta \arrow[u, phantom, "\Cup" description] & \begin{pmatrix} F_\theta(z_1) \\ \vdots \\ F_\theta(z_k) \end{pmatrix} \arrow[ur, bend right, "\text{Vectorize (unroll)}"'] & \end{tikzcd}$$

Functional dimension at θ_0 relative to Z :

FORMALLY:

Fix $Z = \{z_1, \ldots, z_k\} \subseteq \mathbb{R}^{n_0}$

$$\begin{array}{ccccc} \mathrm{Ev}_Z : \mathbb{R}^D & \longrightarrow & M_{k\times n_d} & \longrightarrow & \mathbb{R}^{kn_d} \\ \Downarrow & & & & \\ \theta & & \begin{pmatrix} F_\theta(z_1) \\ \vdots \\ F_\theta(z_k) \end{pmatrix} & \xrightarrow{\text{Vectorize (unroll)}} & \end{array}$$

Functional dimension at θ_0 relative to Z : Rank $\left(J_\theta(\mathrm{Ev}_Z)\,|_{\theta_0} = \left(\frac{\partial \mathrm{Ev}_Z}{\partial \theta} \right)\Bigg|_{\theta_0} \right)$

FORMALLY:

Fix $Z = \{z_1, \ldots, z_k\} \subseteq \mathbb{R}^{n_0}$

$$
\begin{tikzcd}
\mathrm{Ev}_Z : \mathbb{R}^D \arrow[r] & M_{k\times n_d} \arrow[r] & \mathbb{R}^{kn_d} \\
\theta \arrow[u, phantom, "\in" rotate=90] & \begin{pmatrix} F_\theta(z_1) \\ \vdots \\ F_\theta(z_k) \end{pmatrix} \arrow[ur, bend right, "\text{Vectorize (unroll)}"'] &
\end{tikzcd}
$$

Functional dimension at θ_0 relative to Z : Rank $\left(J_\theta(\mathrm{Ev}_Z)\,|_{\theta_0} = \left(\dfrac{\partial \mathrm{Ev}_Z}{\partial \theta} \right)\Bigg|_{\theta_0} \right)$

Dimension of space of tangent vectors at θ_0 impacting the value of F_θ on Z

FORMALLY:

Fix $Z = \{z_1, \ldots, z_k\} \subseteq \mathbb{R}^{n_0}$

$$\begin{array}{ccccc} \mathrm{Ev}_Z : \mathbb{R}^D & \longrightarrow & M_{k \times n_d} & \longrightarrow & \mathbb{R}^{kn_d} \\ \rotatebox{90}{$\in$} & & & & \\ \theta & & \begin{pmatrix} F_\theta(z_1) \\ \vdots \\ F_\theta(z_k) \end{pmatrix} & \xrightarrow{\text{Vectorize (unroll)}} & \end{array}$$

Functional dimension at θ_0 relative to Z : Rank $\left(J_\theta(\mathrm{Ev}_Z)\,|_{\theta_0} = \left(\dfrac{\partial \mathrm{Ev}_Z}{\partial \theta} \right) \Bigg|_{\theta_0} \right)$

"Batch" functional dimension for "Batch" Z

FORMALLY:

Fix $Z = \{z_1, \ldots, z_k\} \subseteq \mathbb{R}^{n_0}$

$$\mathrm{Ev}_Z : \mathbb{R}^D \longrightarrow M_{k \times n_d} \longrightarrow \mathbb{R}^{kn_d}$$

$$\theta \in \mathbb{R}^D, \qquad \begin{pmatrix} F_\theta(z_1) \\ \vdots \\ F_\theta(z_k) \end{pmatrix} \xrightarrow{\text{Vectorize (unroll)}} \mathbb{R}^{kn_d}$$

Functional dimension at θ_0 ~~relative to Z~~ : $\sup_{Z} \text{Rank} \left(J_\theta(\mathrm{Ev}_Z)\big|_{\theta_0} = \left(\frac{\partial \mathrm{Ev}_Z}{\partial \theta} \right) \bigg|_{\theta_0} \right)$

FORMALLY:

Fix $Z = \{z_1, \dots, z_k\} \subseteq \mathbb{R}^{n_0}$

$$
\begin{tikzcd}
\mathrm{Ev}_Z : \mathbb{R}^D \arrow[r] & M_{k \times n_d} \arrow[r] & \mathbb{R}^{kn_d} \\
\theta \arrow[u, phantom, sloped, "\in"] & \begin{pmatrix} F_\theta(z_1) \\ \vdots \\ F_\theta(z_k) \end{pmatrix} \arrow[ur, bend right, "\text{Vectorize (unroll)}"'] &
\end{tikzcd}
$$

Functional dimension at θ_0 ~~relative to Z~~ : $\sup\limits_{Z} \operatorname{Rank} \left(J_\theta(\mathrm{Ev}_Z)\,|_{\theta_0} = \left(\frac{\partial \mathrm{Ev}_Z}{\partial \theta} \right) \Big|_{\theta_0} \right)$

Dimension of space of tangent vectors at θ_0 impacting the value of F_θ anywhere

FORMALLY:

Fix $Z = \{z_1, \ldots, z_k\} \subseteq \mathbb{R}^{n_0}$

$$\mathrm{Ev}_Z : \underset{\theta}{\mathbb{R}^D} \ni \theta \longrightarrow M_{k \times n_d} \longrightarrow \mathbb{R}^{kn_d}$$

$$\begin{pmatrix} F_\theta(z_1) \\ \vdots \\ F_\theta(z_k) \end{pmatrix}$$

Vectorize (unroll)

Functional dimension at θ_0 ~~relative to Z~~ : $\sup_Z$ Rank $\left(J_\theta(\mathrm{Ev}_Z)\,|_{\theta_0} = \left(\frac{\partial \mathrm{Ev}_Z}{\partial \theta} \right) \Big|_{\theta_0} \right)$

This is the (local) functional dimension at θ_0

TECHNICAL POINTS:

TECHNICAL POINTS:

- Choosing Z to contain $n_0 + 1$ points in each linear region guarantees that we achieve the supremum of the rank of $J_\theta(\mathrm{Ev}_Z)|_{\theta_0}$ over all finite sets Z

TECHNICAL POINTS:

- Choosing Z to contain $n_0 + 1$ points in each linear region guarantees that we achieve the supremum of the rank of $J_\theta(\mathsf{Ev}_Z)\,|_{\theta_0}$ over all finite sets Z
- Guaranteeing that a batch Z satisfies the above assumptions is computationally challenging

TECHNICAL POINTS:

- Choosing Z to contain $n_0 + 1$ points in each linear region guarantees that we achieve the supremum of the rank of $J_\theta(\mathrm{Ev}_Z)\,|_{\theta_0}$ over all finite sets Z
- Guaranteeing that a batch Z satisfies the above assumptions is computationally challenging
- The placement of the batch Z relative to the decomposition of the domain into linear regions is highly relevant

WHY CARE ABOUT (BATCH) FUNCTIONAL DIMENSION?

WHY CARE ABOUT (BATCH) FUNCTIONAL DIMENSION?

Low functional dimension

$=$

High local redundancy

$=$

Low complexity

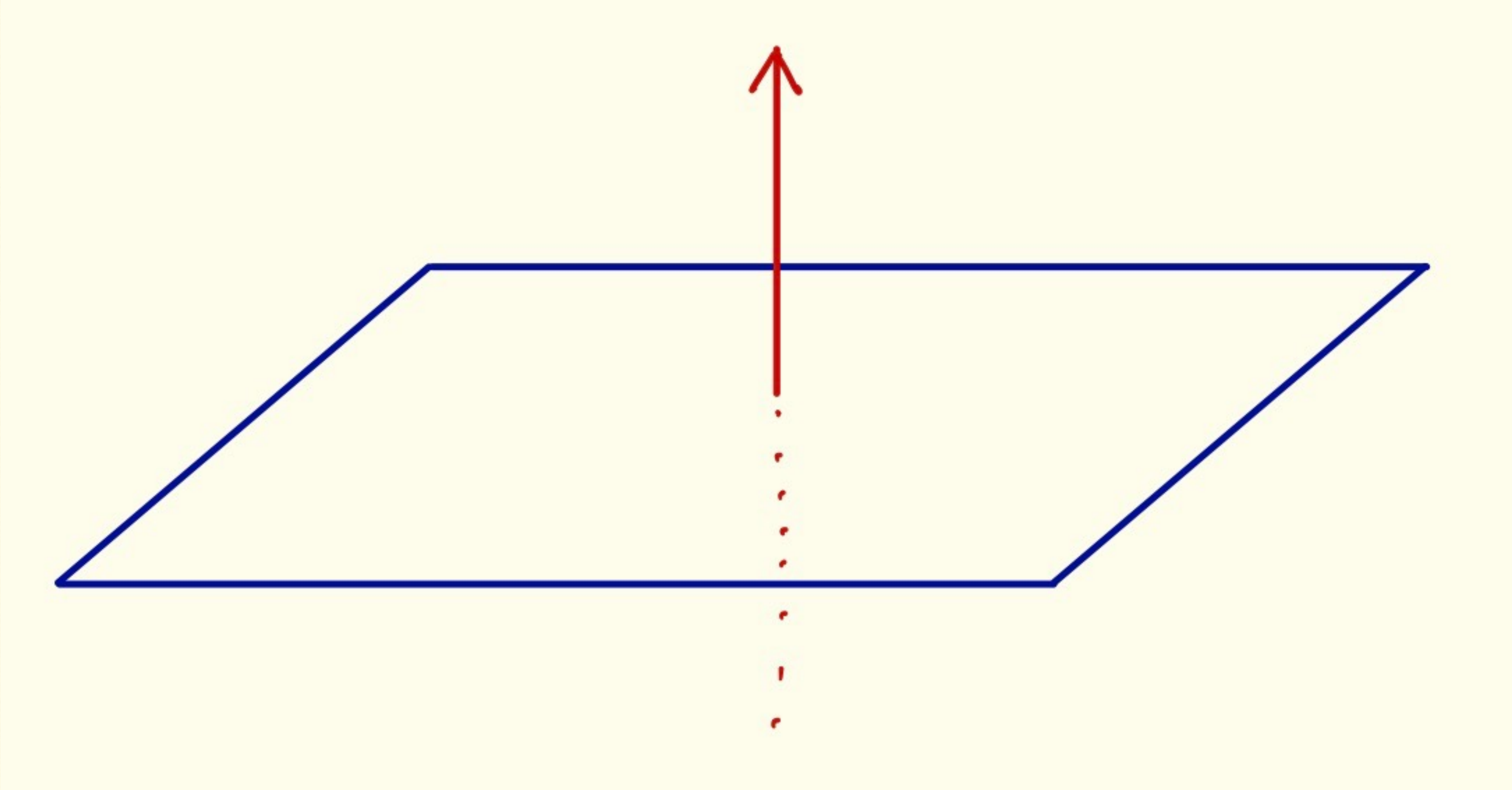
$\Downarrow$

*Global minima of loss landscape corresponding to parameters with low functional dimension should be flat in more directions**

*This is a heuristic; not a theorem yet

LOW FUN. DIM. ⇒ HIGH-DIMENSIONAL LEVEL SETS

Direction(s) in which
function can change



"Level set" for function
$\subseteq$
Level set for loss

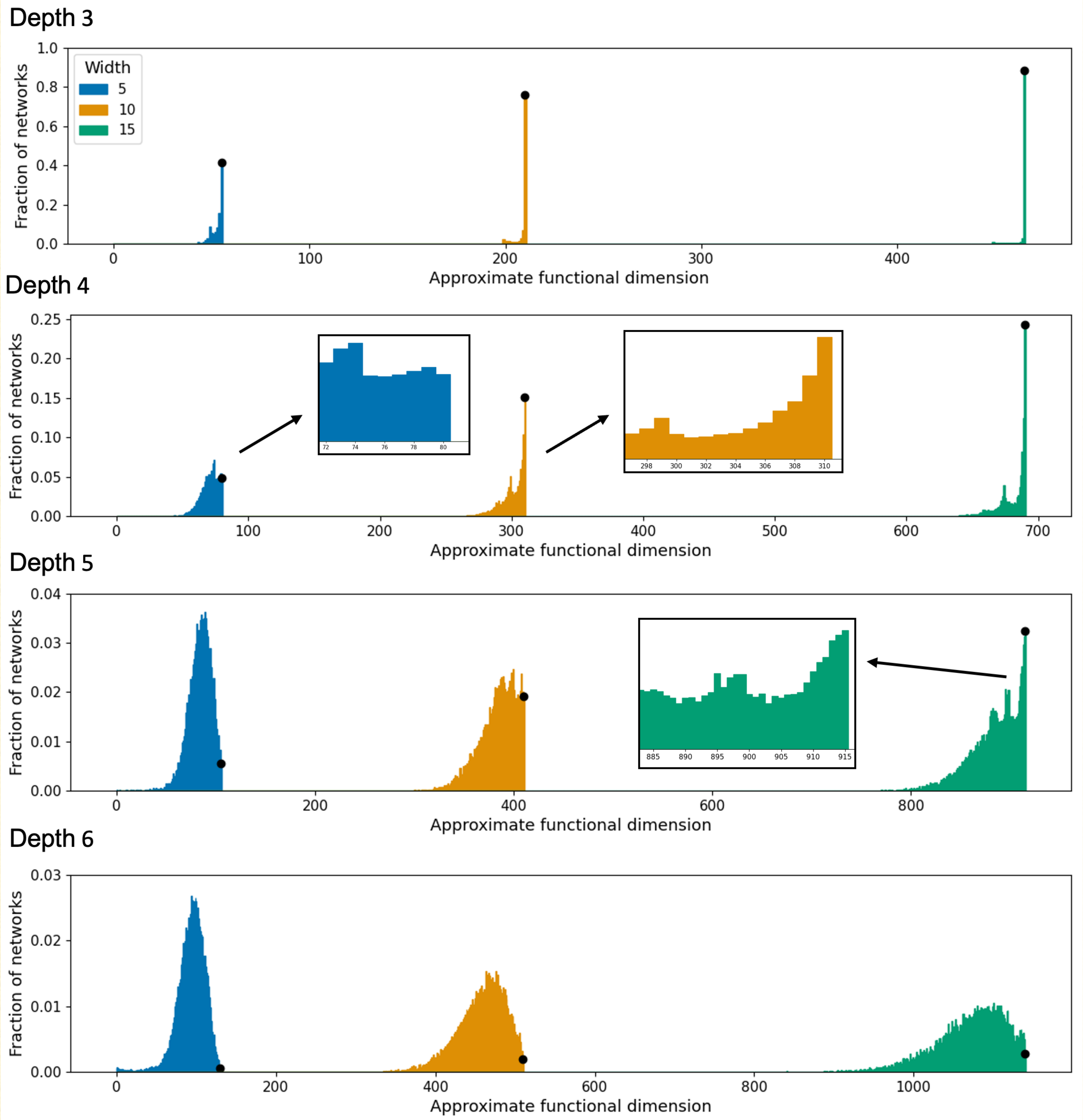
FLATTER GLOBAL MINIMA OF THE LOSS FUNCTION ARE PREFERRED

- (Y. Cooper, '18): *The loss landscape of overparameterized neural networks*
- (Ma-Ying, '22): *On Linear Stability of SGD and Input-Smoothness of Neural Networks*
- (Li-Wang-Arora, '22): *What happens after SGD reaches zero loss? A mathematical framework*
- (Blanc-Gupta-Valiant-Valiant, '20): *Implicit regularization for deep neural networks driven by an Ornstein-Uhlenbeck like process*

(Partial list: please help!)

OUTLINE:

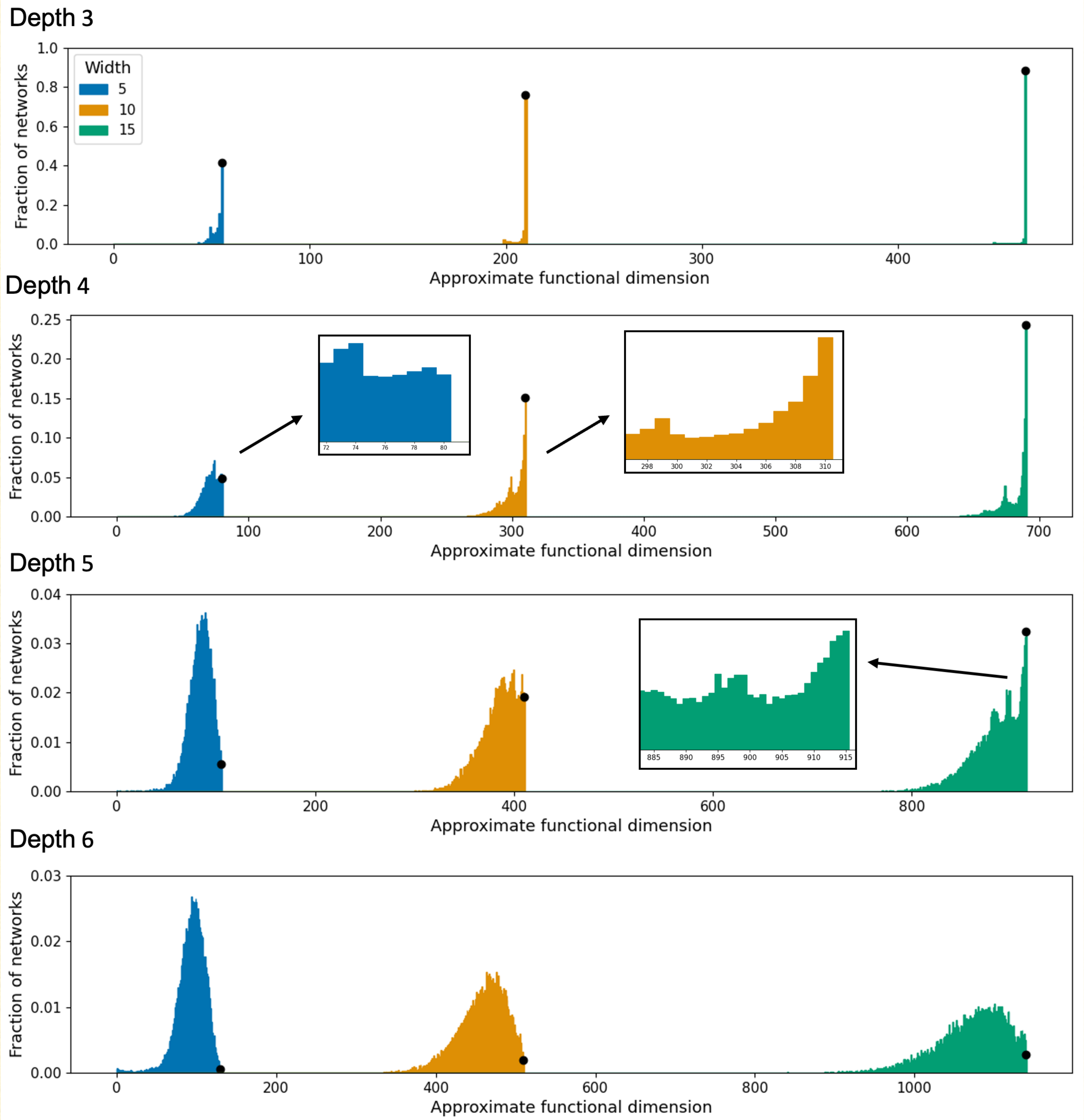
1. Parameter space $\neq$ Function space for ReLU networks
2. (Effective) functional dimension
3. Theoretical and experimental results



EXPERIMENTS

- Width = 5,10,15, Depth = 3,4,5,6
- # of points in each batch: $|Z| = m = 2D'$
- *Batch not guaranteed to achieve sup, so APPROXIMATE functional dimension*
- 20K networks in each run
- Weights sampled i.i.d. w/ variance 2/fan-in, bias w/ variance 0.01
- Black dot represents percentage of sample networks achieving the theoretical upper bound

Image credit: D. Rolnick



TAKE-AWAYS:

EXPECTED DEFICIT FROM UPPER BOUND:

- ↓ with width, ↑ with depth

VARIANCE:

- ↑ with width, ↑ with depth

MODES:

- Multimodal, especially at ↑ width, ↓ depth
- Modes appear to be separated (roughly) by width

Image credit: D. Rolnick

THEORETICAL RESULTS:

THEORETICAL RESULTS:

- For *every* architecture, a positive measure subset of parameter space **fails to achieve** the theoretical upper bound on functional dimension

THEORETICAL RESULTS:

- For *every* architecture, a positive measure subset of parameter space **fails to achieve** the theoretical upper bound on functional dimension
- For *every* architecture *whose hidden layers are at least as wide as the input layer*, a positive measure subset of parameter space **achieves** the upper bound on functional dimension*

*Conjecture (90% theorem)

MECHANISMS INSURING UPPER BOUND IS ACHIEVED*:

Theorem (G-Lindsey-Rolnick '22): For every architecture $(n_0, \ldots, n_d)$ with $n_i \geq n_0$ for $i = 1, \ldots, d-1$, there exists a positive measure subset of parameter space that admits no hidden symmetries (parameters can be recovered up to permutation and positive scaling)

MECHANISMS INSURING UPPER BOUND IS ACHIEVED*:

Theorem (G-Lindsey-Rolnick '22): For every architecture $(n_0, \ldots, n_d)$ with $n_i \geq n_0$ for $i = 1, \ldots, d-1$, there exists a positive measure subset of parameter space that admits no hidden symmetries (parameters can be recovered up to permutation and positive scaling)

*Conjecture (90% Theorem)

MECHANISMS INSURING UPPER BOUND IS ACHIEVED*:

Proof sketch:

*Conjecture (90% Theorem)

MECHANISMS INSURING UPPER BOUND IS ACHIEVED*:

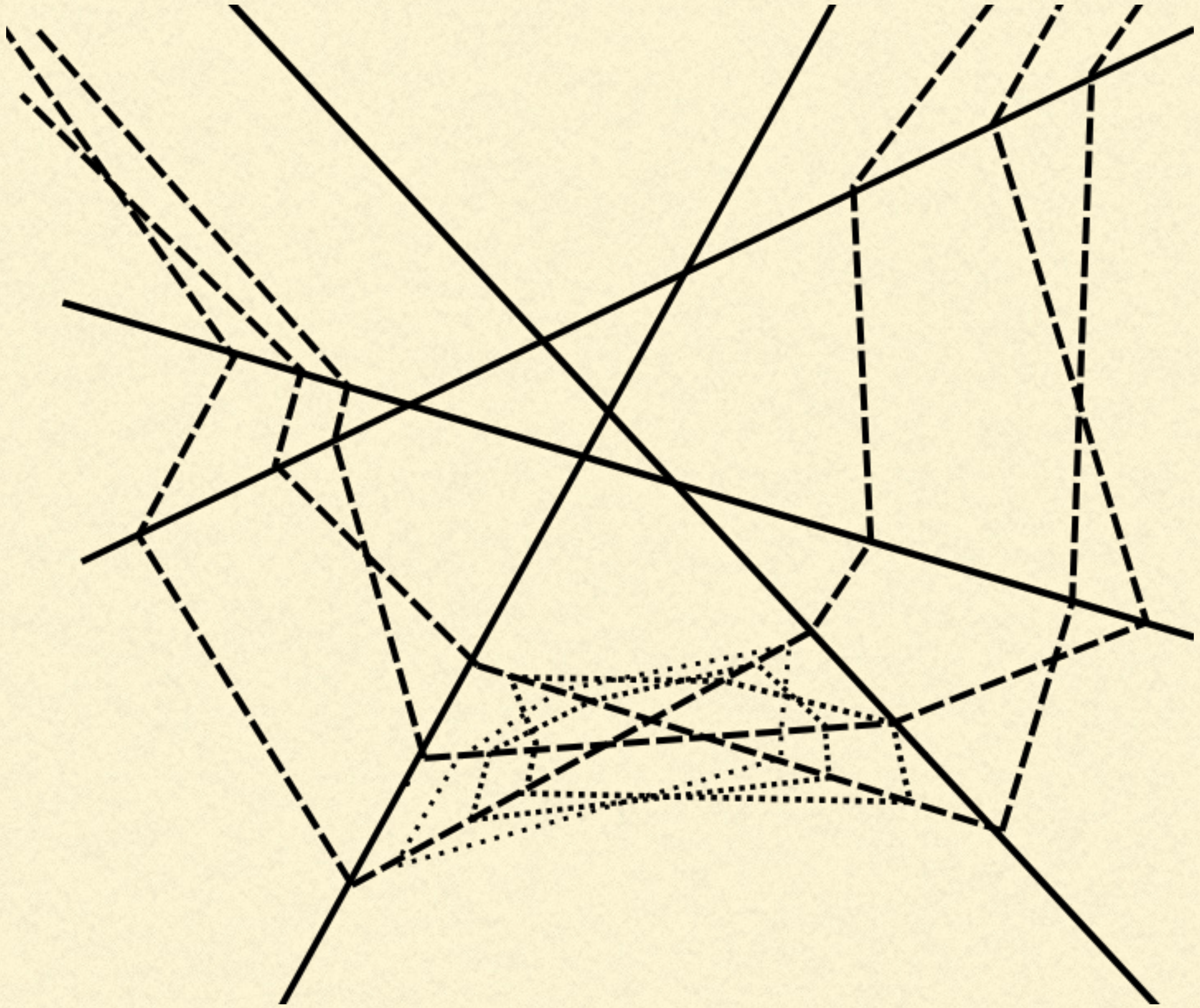
Proof sketch:

- Kording-Rolnick proved: If *every* pair of bent hyperplanes from every pair of adjacent layers intersects *transversely* (with expected dimension), then the parameters can be recovered up to permutation and positive scaling
- We give a construction ensuring the Kording-Rolnick condition is satisfied
- *Remark: The construction is fiddly! I don't have a great sense of how often the transverse pairwise intersection condition is satisfied in general, especially for deep networks.*

*Conjecture (90% Theorem)

MECHANISMS INSURING UPPER BOUND IS ACHIEVED*:

Illustration of construction
For architecture (2,5,3,3)



FURTHER QUESTIONS:

FURTHER QUESTIONS:

- How does functional dimension evolve during training?

FURTHER QUESTIONS:

- How does functional dimension evolve during training?
- How about in the overparameterized setting?

FURTHER QUESTIONS:

- How does functional dimension evolve during training?
- How about in the overparameterized setting?
- Better understanding of the mechanisms affecting (effective) functional dimension?

FURTHER QUESTIONS:

- How does functional dimension evolve during training?
- How about in the overparameterized setting?
- Better understanding of the mechanisms affecting (effective) functional dimension?
- Dependence of (batch) functional dimension on symmetries/geometry of data-generating distribution?

THANK YOU FOR BEING HERE!