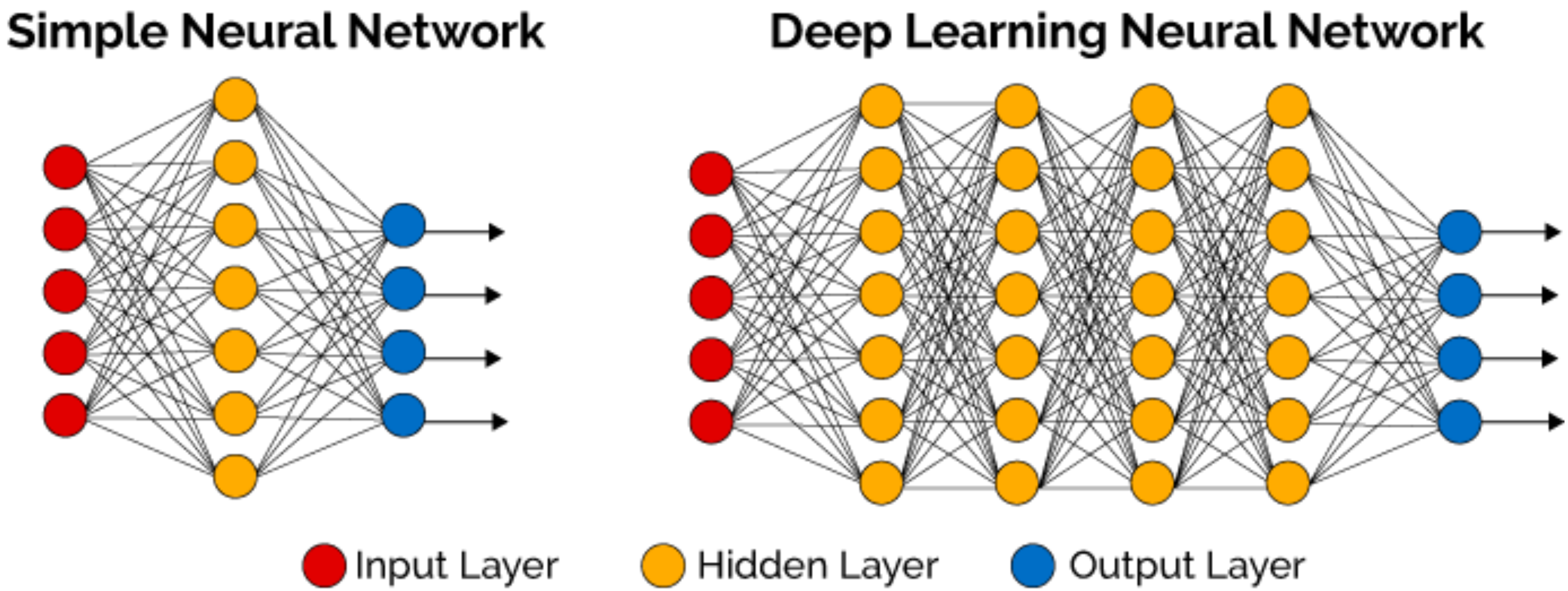
Dynamical Isometry is Achieved in Residual Networks in a Universal Way for any Activation Function

Wojciech Tarnowski, Piotr Warchoł, Stanisław Jastrzębski,
Jacek Tabor, Maciej Nowak

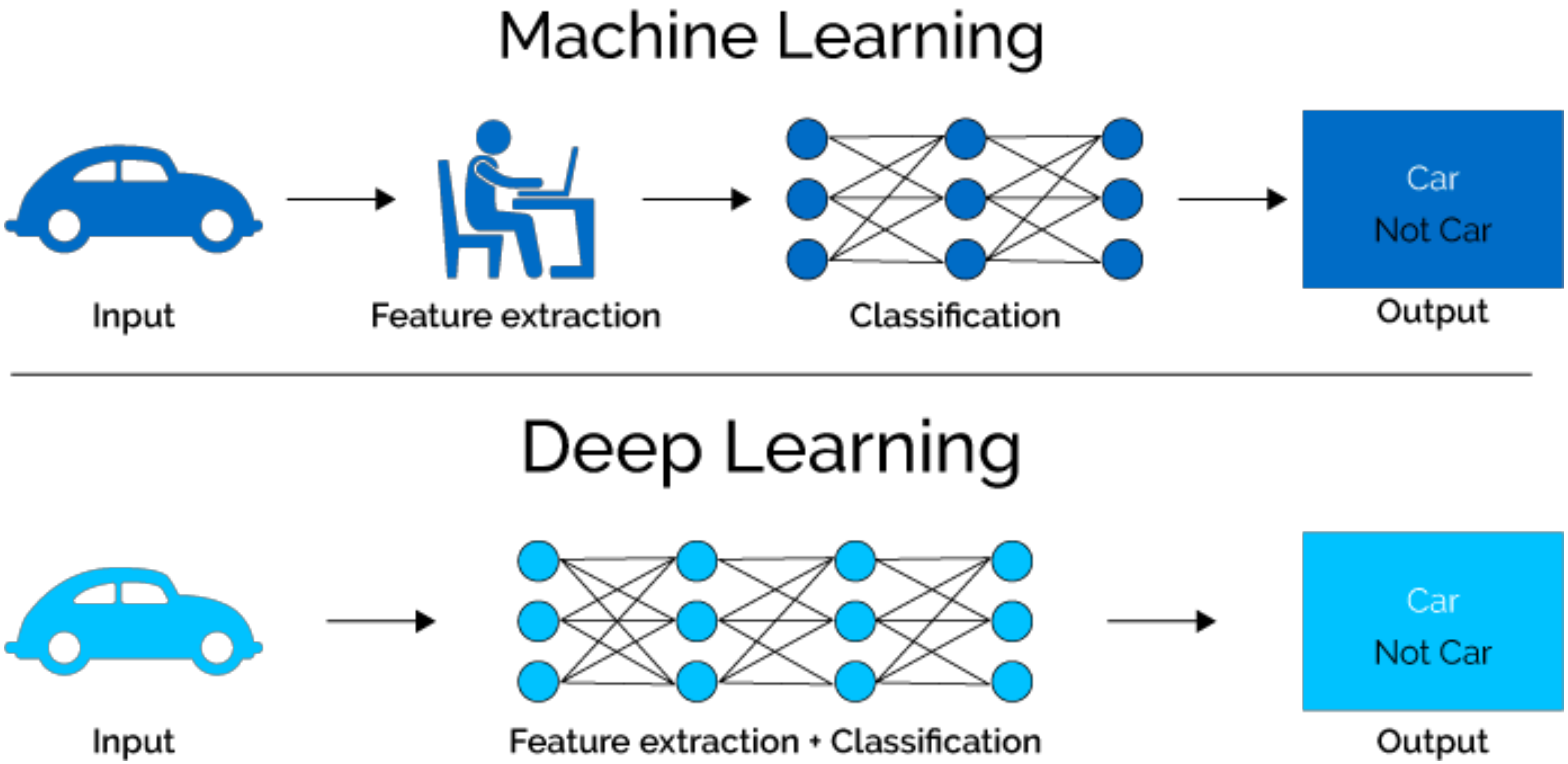
JAGIELLONIAN UNIVERSITY
IN KRAKÓW

Paper accepted to AISTATS 2019

- (Deep) artificial neural networks - a short introduction



- (Deep) artificial neural networks - a short introduction

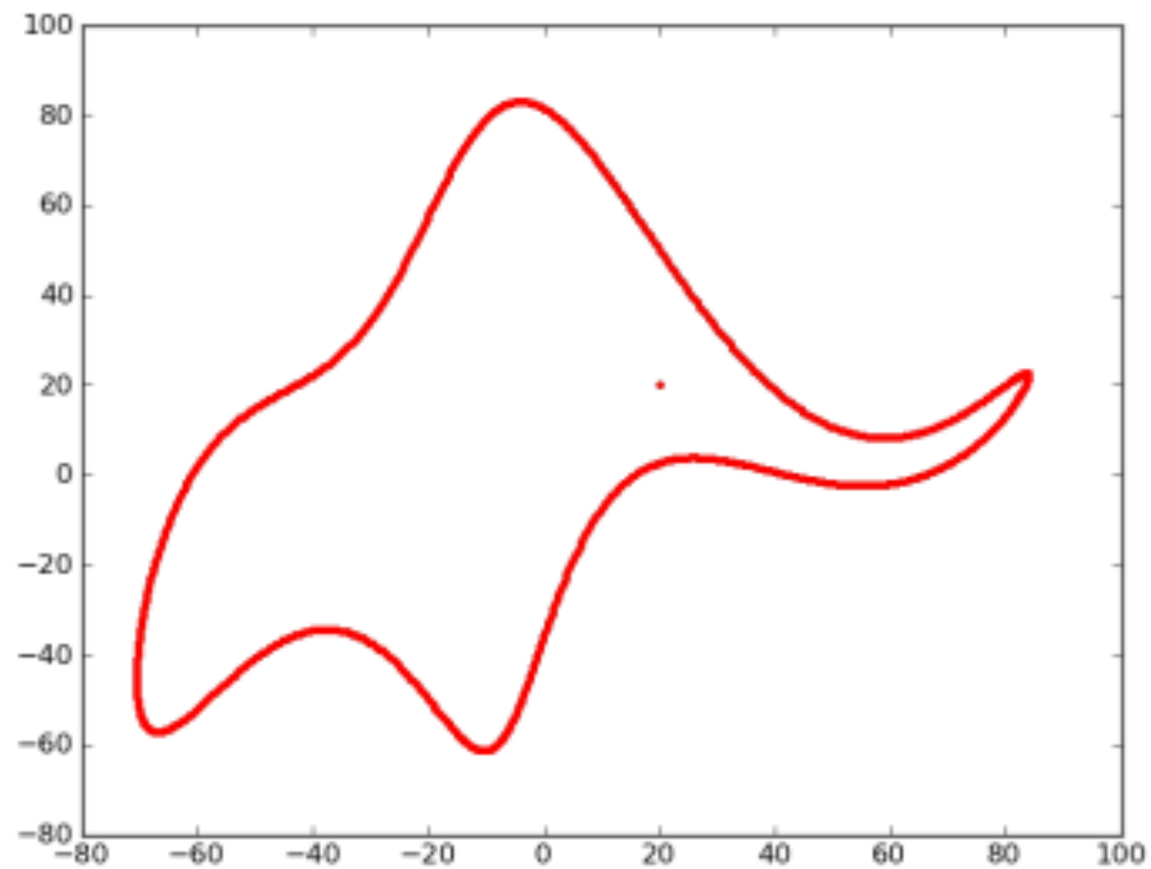


• (Deep) artificial neural networks - a short introduction

Johnny von Neumann: with four parameters I can fit an elephant,
and with five I can make him wiggle his trunk.

• (Deep) artificial neural networks - a short introduction

Johnny von Neumann: with four parameters I can fit an elephant, and with five I can make him wiggle his trunk.



“Drawing an elephant with four complex parameters” by Jurgen Mayer, Khaled Khairy, and Jonathon Howard, Am. J. Phys. 78, 648 (2010), DOI:10.1119/1.3254017.

State-of-the-art deep neural nets sometimes contain millions or even billions of parameters!

- (Deep) artificial neural networks - a short introduction

Fundamental:

- **Massive over parametrization… so why don't deep NN overfit?**
- …

Technical:

- What activation functions to use?
- How to train more efficiently
- How to initialize?
- How to choose architecture?
- …

Practical:

- Explainability
- Robustness
- …

- The problem in short

We tackle the problem of initialization
of deep Residual Neural Networks
with Random Matrix and Free Probability Theories.

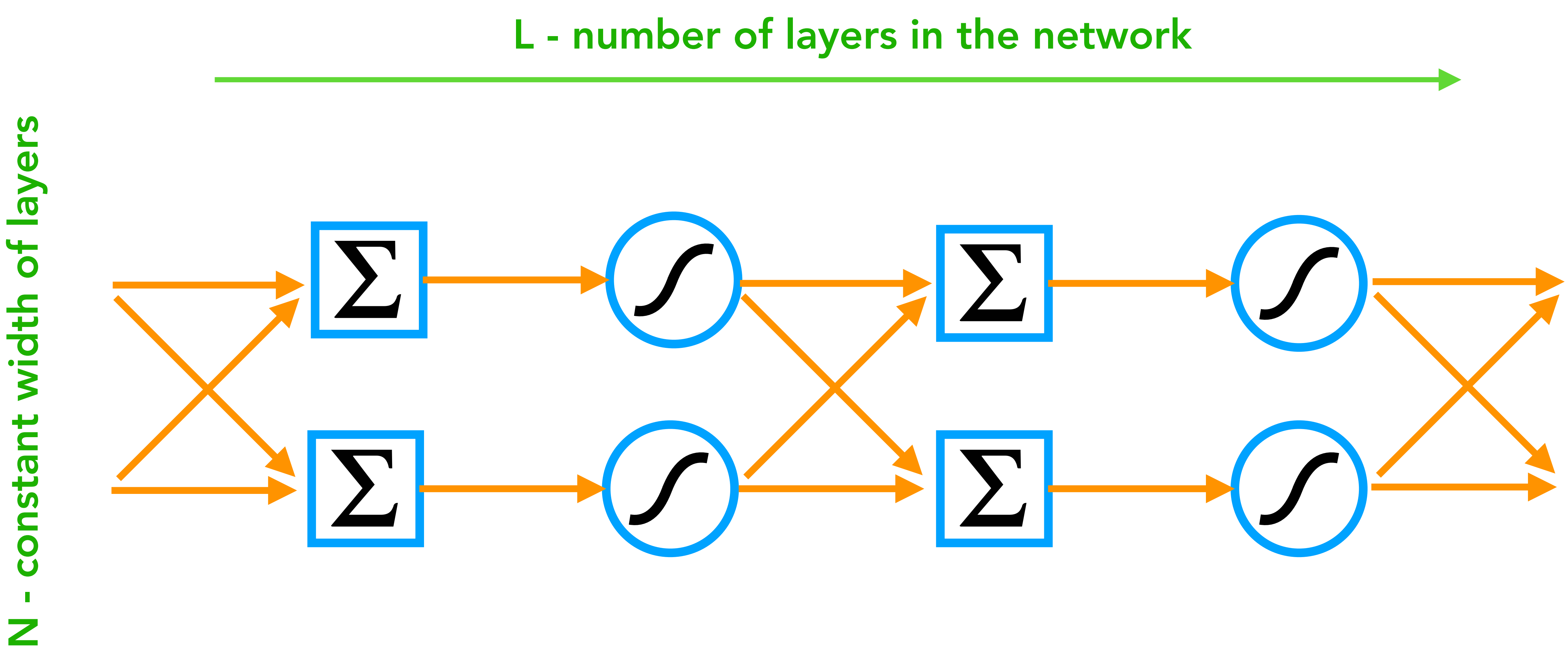
This is done by making sure the singular spectrum of the input-output Jacobian is concentrated around one.
This is called dynamical isometry.

$$J_{ik} = \frac{\partial x_i^L}{\partial x_k^0}$$

Presentation plan:

- ~~(Deep) artificial neural networks - a short introduction~~
- ~~The problem in short~~
- Stating the problem in more detail
- The case of Feed Forward Networks
- What we find for ResNets

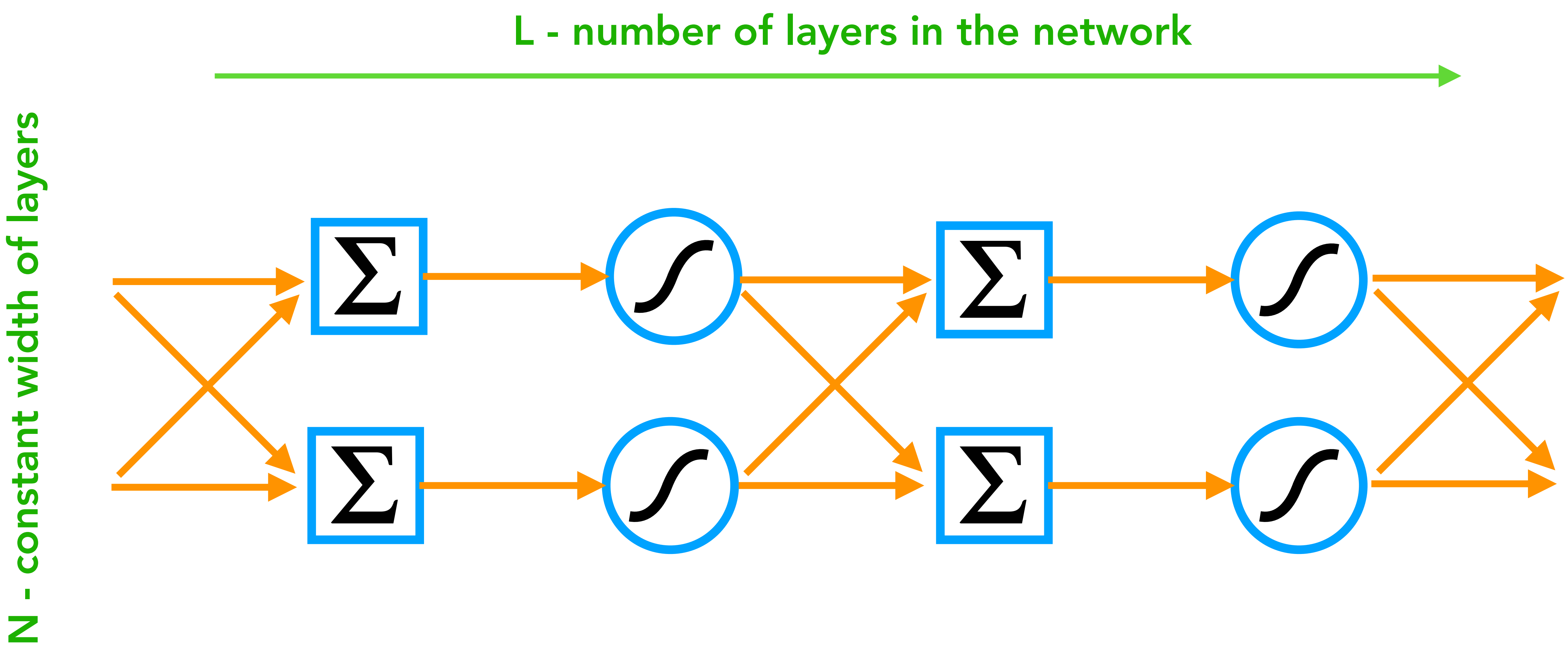
• Stating the problem



Signal propagation:

$$\mathbf{x^l} = \phi(\mathbf{h^l}), \quad \mathbf{h^l} = \mathbf{W^l x^{l-1}} + \mathbf{b^l}$$

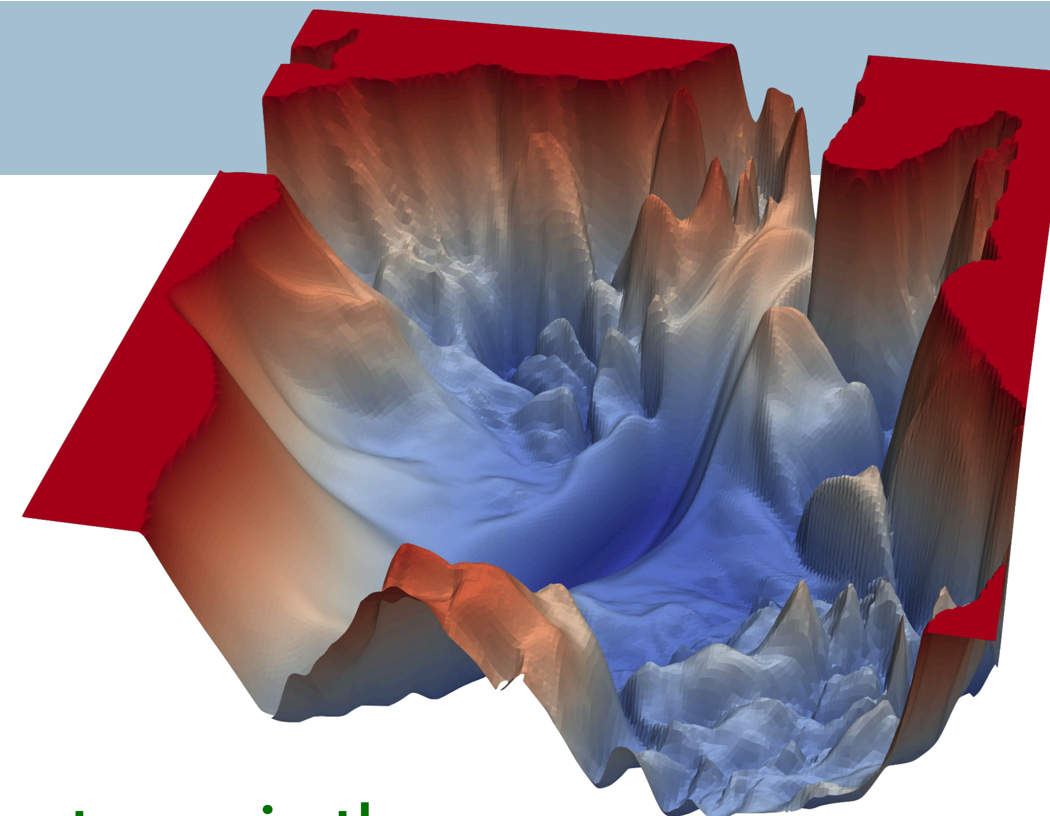
- Stating the problem



Signal propagation: $$\mathbf{x}^{\mathbf{l}} = \phi(\mathbf{h}^{\mathbf{l}}), \quad \mathbf{h}^{\mathbf{l}} = \mathbf{W}^{\mathbf{l}}\mathbf{x}^{\mathbf{l-1}} + \mathbf{b}^{\mathbf{l}}$$

Question: Can we say sth about how weights (bias) initialization will effect learning?

- Stating the problem



$$\Delta W^l_{ij} = -\eta \frac{\partial E(\boldsymbol{x}^L, \boldsymbol{y})}{\partial W^l_{ij}}$$

The learning process is based on gradually modifying the weights of the network

$$\Delta W^l_{ij} = -\eta \sum_{k,t} \frac{\partial x^l_t}{\partial W^l_{ij}} \frac{\partial x^L_k}{\partial x^l_t} \frac{\partial E(\boldsymbol{x}^L, \boldsymbol{y})}{\partial x^L_k}$$

All the terms in the sum of products must be bounded

$$J_{ik} = \frac{\partial x^L_i}{\partial x^0_k}$$

The input-output Jacobian is the most problematic one

It can be rewritten as:

$$\boldsymbol{J} = \prod_{l=1}^{L} \left(\boldsymbol{D}^l \boldsymbol{W}^l \quad \right),$$

with

$$D^l_{ij} = \phi'(h^l_i)\delta_{ij}$$

• Stating the problem

For a given activation function and network depth L,
how to initialize the weights?

Use Random Matrix and Free Probability Theories to find the singular values of

$$\boldsymbol{J} = \prod_{l=1}^{L} \left(\boldsymbol{D}^{l} \boldsymbol{W}^{l} \quad \right), \qquad \text{with} \qquad D^{l}_{ij} = \phi'(h^{l}_{i})\delta_{ij}$$

Study Signal propagation in the network to find the statistics of

• The case of Feed Forward Networks

(2017, Pennigton et al.)

Activation function: $\phi(h) = \tanh(h)$

Weight matrix at initialization orthogonal: $W^T W = \mathbf{1}$

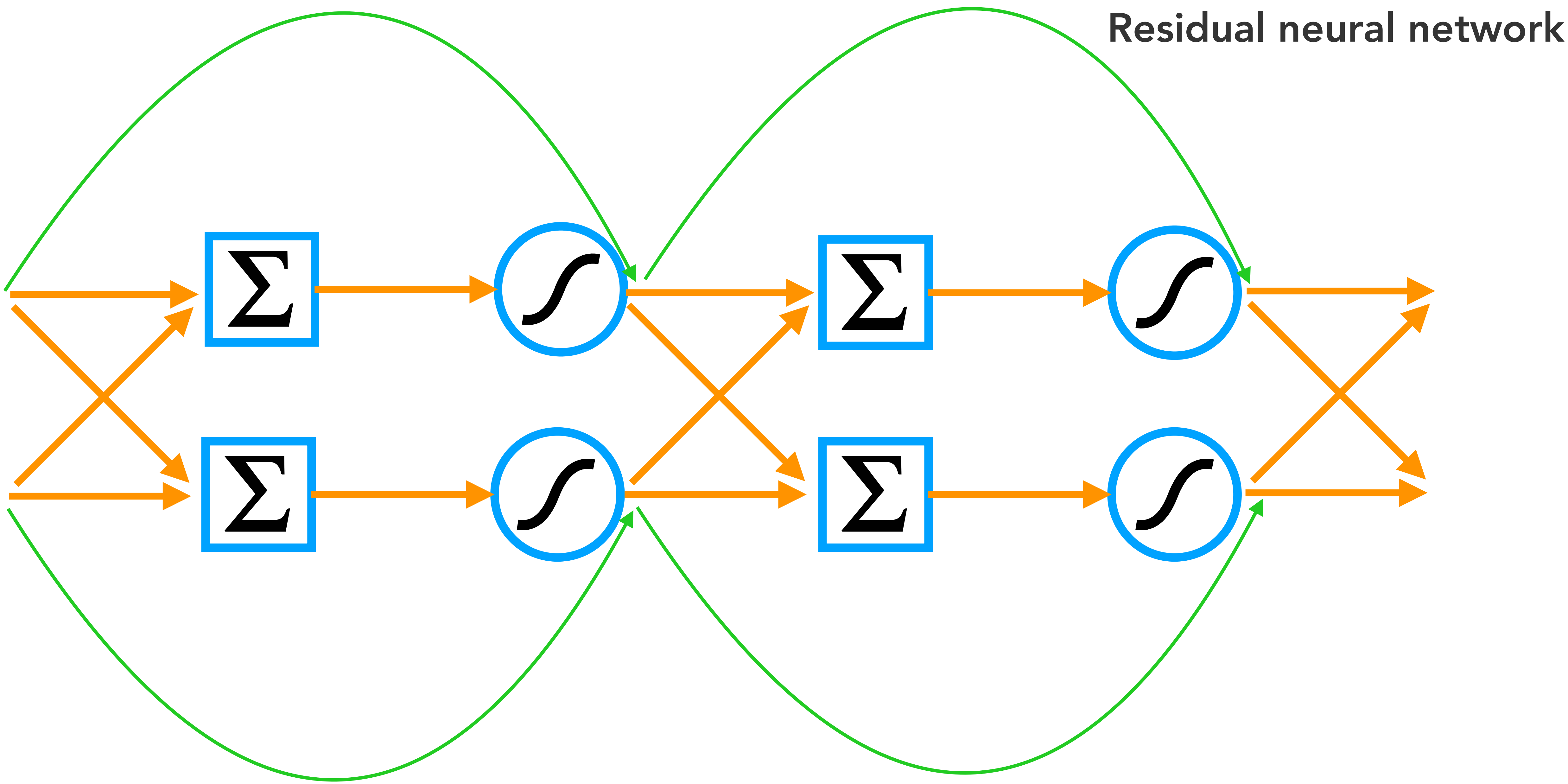
Fixed point related to signal propagation.

Dynamical Isometry in feed forward neural network

"Orders of magnitude" faster learning of DEEP feed forward neural networks. Training of 10000 layer vanilla CNN.

Not possible at all for ReLU (in feed forward networks).

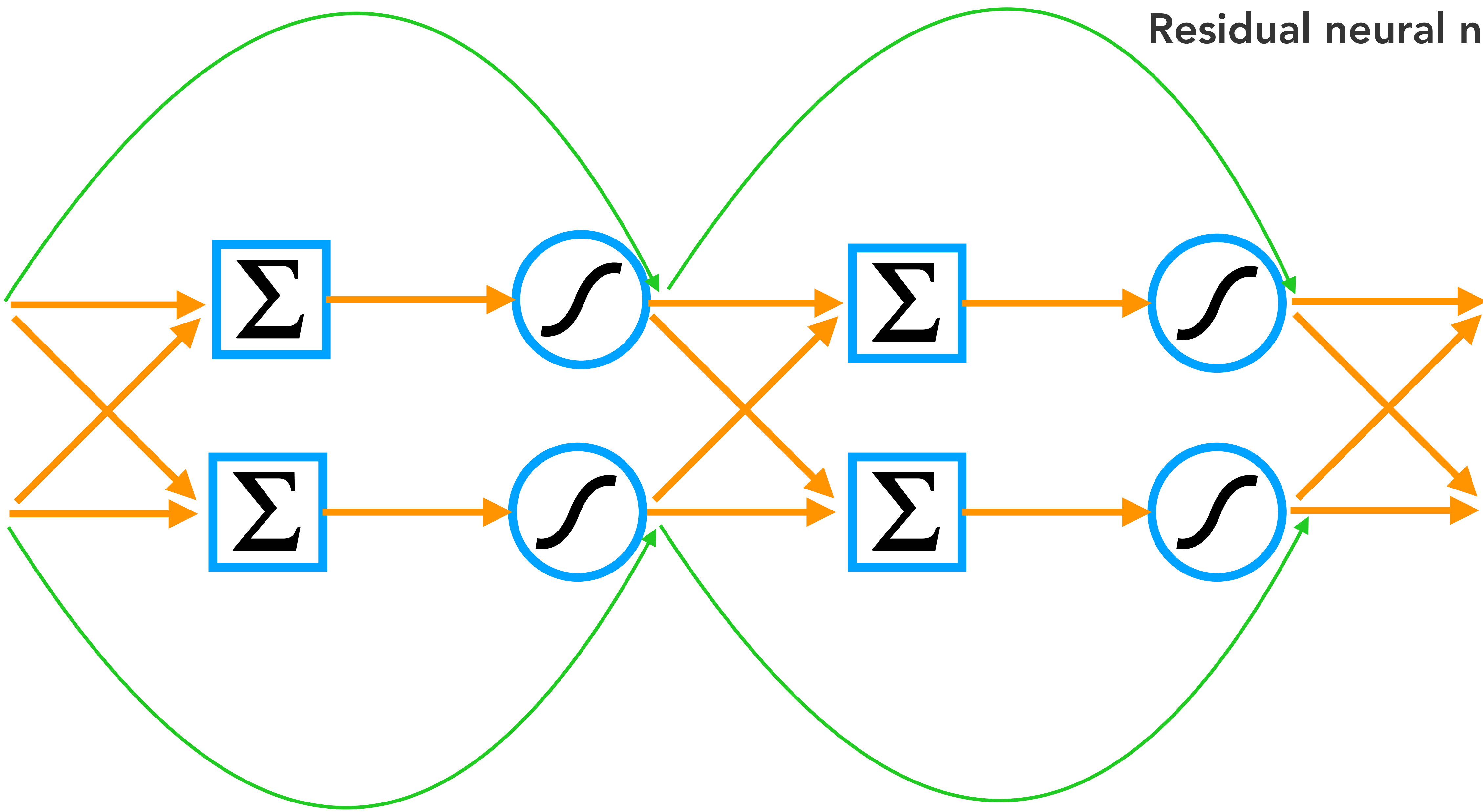
- What we find for ResNets



Signal propagation: $$\mathbf{x}^{\mathbf{l}} = \phi(\mathbf{h}^{\mathbf{l}}) + \mathbf{a}\mathbf{x}^{\mathbf{l-1}}, \quad \mathbf{h}^{\mathbf{l}} = \mathbf{W}^{\mathbf{l}}\mathbf{x}^{\mathbf{l-1}} + \mathbf{b}^{\mathbf{l}}$$

(a slightly more sophisticated version outmatched other models in the 2015 ILSVRC and COCO competitions)

- What we find for ResNets



Signal propagation: $$\mathbf{x}^l = \phi(\mathbf{h}^l) + \boxed{\mathbf{a}\mathbf{x}^{l-1}}, \quad \mathbf{h}^l = \mathbf{W}^l\mathbf{x}^{l-1} + \mathbf{b}^l$$

Question: Can we say sth about how weights (bias) initialization will effect learning?

• What we find for ResNets

For a given activation function and network depth L,
how to initialize the weights?

Use Random Matrix and Free Probability Theories to find the singular values of

$$J = \prod_{l=1}^{L} \left(\boldsymbol{D}^l \boldsymbol{W}^l + \mathbf{1}a \right), \quad \text{with} \quad D^l_{ij} = \phi'(h^l_i)\delta_{ij}$$

Study Signal propagation in the network to find the statistics of

• Rapid introduction to RMT

Random Matrix Theory

$$G_H(z) = \left\langle \frac{1}{N} \mathrm{Tr}\,(z\mathbf{1} - \boldsymbol{H})^{-1} \right\rangle = \int_{-\infty}^{\infty} \frac{\rho_H(\lambda) d\lambda}{z - \lambda}$$

$$\rho_H(x) = -\frac{1}{\pi} \lim_{\epsilon \to 0} G_H(x + i\epsilon)$$

Free Probability Theory

$$G\left(R(z) + \frac{1}{z}\right) = z, \qquad R(G(z)) + \frac{1}{G(z)} = z.$$

$$R_{X+Y}(z) = R_X(z) + R_Y(z)$$

R-transform

$$S(zR(z)) = \frac{1}{R(z)}, \quad R(zS(z)) = \frac{1}{S(z)}.$$

$$S_{AB}(z) = S_A(z) S_B(z)$$

S-transform

- What we find for ResNets

Singular spectrum of $$\boldsymbol{J} = \prod_{l=1}^{L}\left(\boldsymbol{D}^l\boldsymbol{W}^l + \mathbf{1}a\right)$$

Calculate $$S_{JJ^T}(z) = \prod_{l=1}^{L} S_{Y_l Y_l^T}(z)$$ then revert back to the Greens function

$$G(z) = \left\langle \frac{1}{N}\mathrm{Tr}(z\mathbf{1} - \boldsymbol{Y}_l\boldsymbol{Y}_l^T)^{-1}\right\rangle \qquad \boldsymbol{Y}_l = (a\mathbf{1} + \boldsymbol{D}^l\boldsymbol{W}^l) \qquad \boldsymbol{X} = \boldsymbol{D}^l\boldsymbol{W}^l$$

Generalized resolvent $$\mathcal{G} := \begin{pmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \end{pmatrix} = \left\langle \frac{1}{N}\mathrm{bTr}\begin{pmatrix} -a - \boldsymbol{X} & 1 \\ z & -a - \boldsymbol{X}^T \end{pmatrix}^{-1}\right\rangle$$

$$\mathcal{Z} := \begin{pmatrix} -a & 1 \\ z & -a \end{pmatrix}, \qquad \mathcal{X} := \begin{pmatrix} \boldsymbol{X} & 0 \\ 0 & \boldsymbol{X}^T \end{pmatrix}$$

Generalized R-transform $$\mathcal{G}(\mathcal{Z}) = (\mathcal{Z} - \mathcal{R}(\mathcal{G}(\mathcal{Z})))^{-1}$$

- What we find for ResNets

Singular spectrum of $\boldsymbol{J} = \prod_{l=1}^{L} \left(\boldsymbol{D}^l \boldsymbol{W}^l + \mathbf{1} a \right)$

$$c_2^l = \sigma_W^2 \left\langle (\phi'(h))^2 \right\rangle_l = \sigma_W^2 \int \mathcal{D}z \phi'^2 \left(\sqrt{q^l} z \right)$$

define the effective cumulant: $c = \frac{1}{L} \sum_{l=1}^{L} c_2^l$

The large network depth limit (recall the scaling of the variance: $\sigma_w^2/(NL)$)

$$\ln S_{JJ^T}(z) = -2L \ln a + \sum_{l=1}^{L} \ln \left(1 - \frac{c_2^l}{a^2 L}(1+2z) \right) \approx -2L \ln a - \frac{1+2z}{a^2 L} \sum_{l=1}^{L} c_2^l =: -2L \ln a - \frac{(1+2z)}{a^2} c$$

$$S_{JJ^T}(z) = \frac{1}{a^{2L}} e^{-\frac{c}{a^2}(1+2z)}$$

Universal formula for any activation function!

$$a^{2L} G(z) = (zG(z) - 1) e^{\frac{c}{a^2}(1 - 2zG(z))}$$

(solution in terms of the Lambert function)

$$c_2^l = \left\langle \frac{1}{N} \mathrm{Tr} \boldsymbol{W}^l \boldsymbol{D}^l \boldsymbol{D}^l (\boldsymbol{W}^l)^T \right\rangle = \frac{\sigma_w^2}{N} \sum_{i}^{N} \left(\phi'(h_i^l) \right)^2 = \sigma_W^2 \int \mathcal{D}z \phi'^2 \left(\sqrt{q^l} z \right)$$

- What we find for ResNets

Signal propagation: $$\mathbf{x^l} = \phi(\mathbf{h^l}) + \mathbf{a}\mathbf{x^{l-1}}, \quad \mathbf{h^l} = \mathbf{W^l}\mathbf{x^{l-1}} + \mathbf{b^l}$$

elements of weight matrices and bias vectors - i.i.d. Gaussian with mean 0 and variances: $\sigma_w^2/(NL)$ and σ_b^2

Study: $$q^l = \frac{1}{N_l}\sum_{i=1}^{N_l}(\mathbf{h}_i^l)^2$$

The resulting mapping:

$$q^{l+1} = a^2 q^l - \left(a^2 - 1\right)\sigma_b^2 + \frac{(\sigma_W)^2}{L}\int \mathcal{D}z\phi^2\left(\sqrt{q^l}z\right) + 2\frac{(\sigma_W)^2}{L}\left[\sum_{k=1}^{l-1} a^k \int \mathcal{D}z\phi\left(\sqrt{q^{l-k}}z\right)\right]\int \mathcal{D}z\phi\left(\sqrt{q^l}z\right)$$

$$c_2^l = \sigma_W^2\left\langle(\phi'(h))^2\right\rangle_l = \sigma_W^2\int \mathcal{D}z\phi'^2\left(\sqrt{q^l}z\right)$$

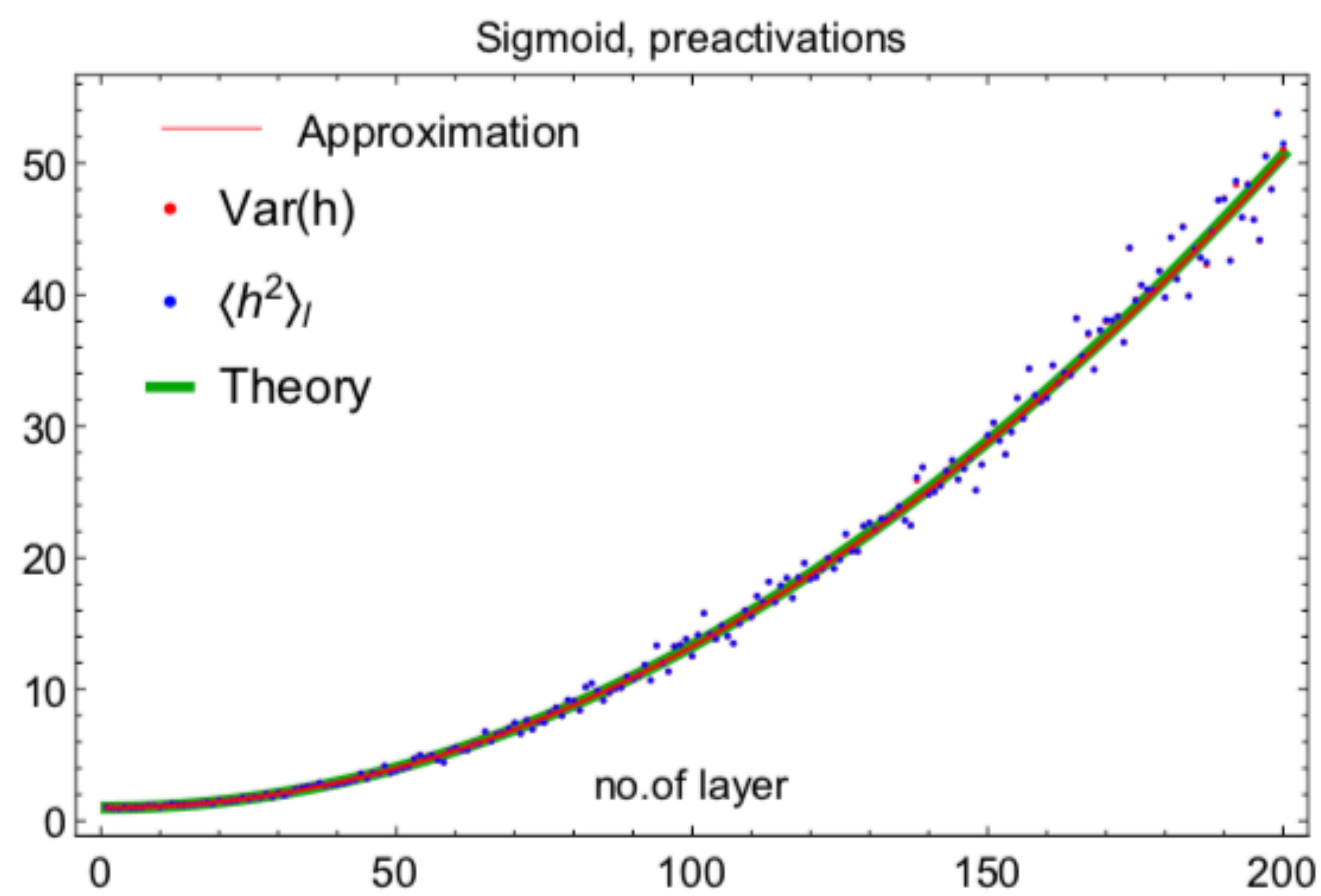
Same result for orthogonal weight matrices.

- **What we find for ResNets**

For example, in the case of the sigmoid activation function: $\phi(x) = \frac{1}{1+e^{-x}}$

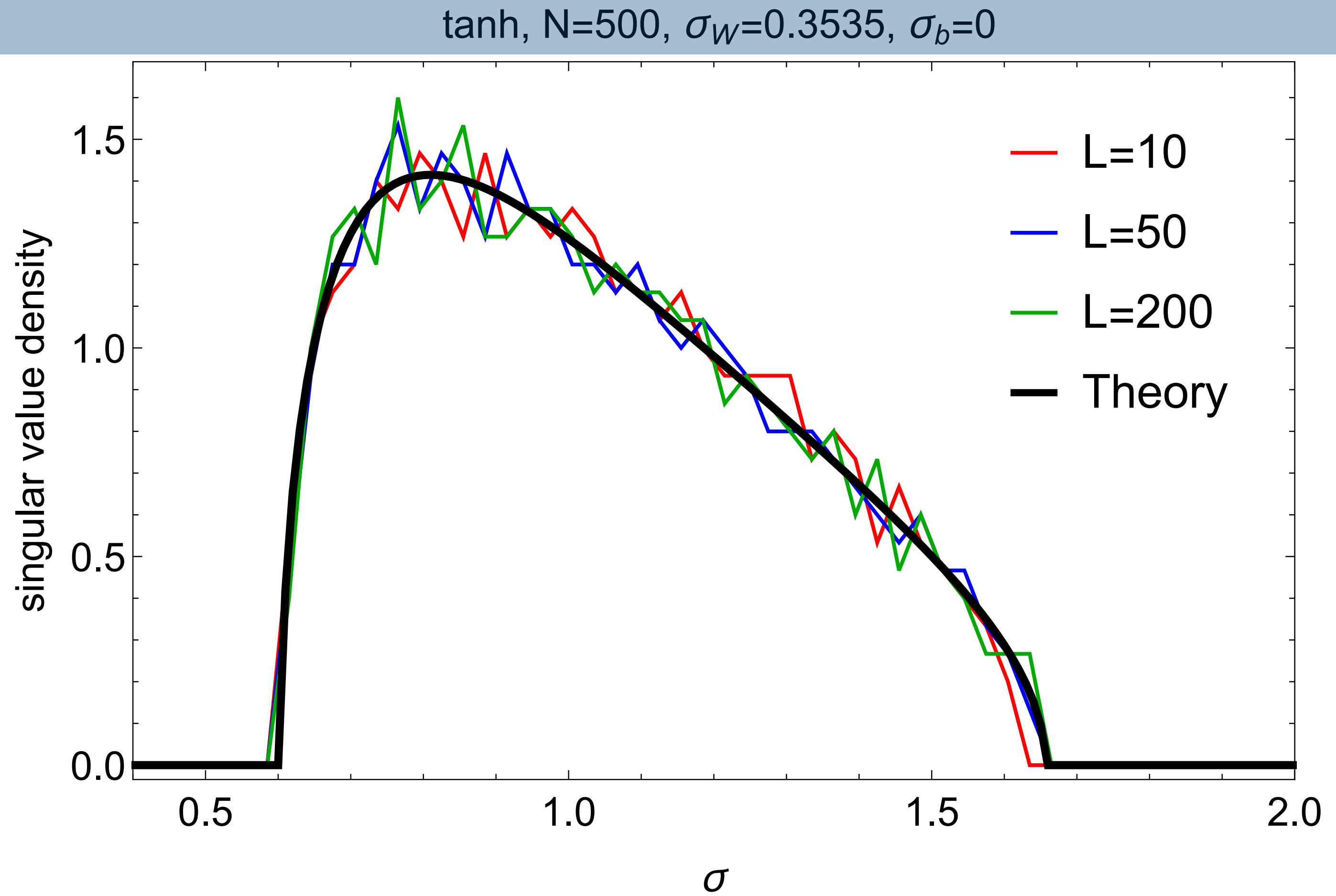
$$q^{l+1} = q^{l} + \frac{(\sigma_W)^2}{L}\int \mathcal{D}z\phi^2\left(\sqrt{q^l}z\right) + \frac{(\sigma_W)^2}{2L}(l-1)$$

$$q^{l+1} \approx q^{1} + \frac{(\sigma_W)^2 l}{L}\int \mathcal{D}z\phi^2\left(\sqrt{q^1}z\right) + \frac{(\sigma_W)^2}{4L}l(l-1)$$

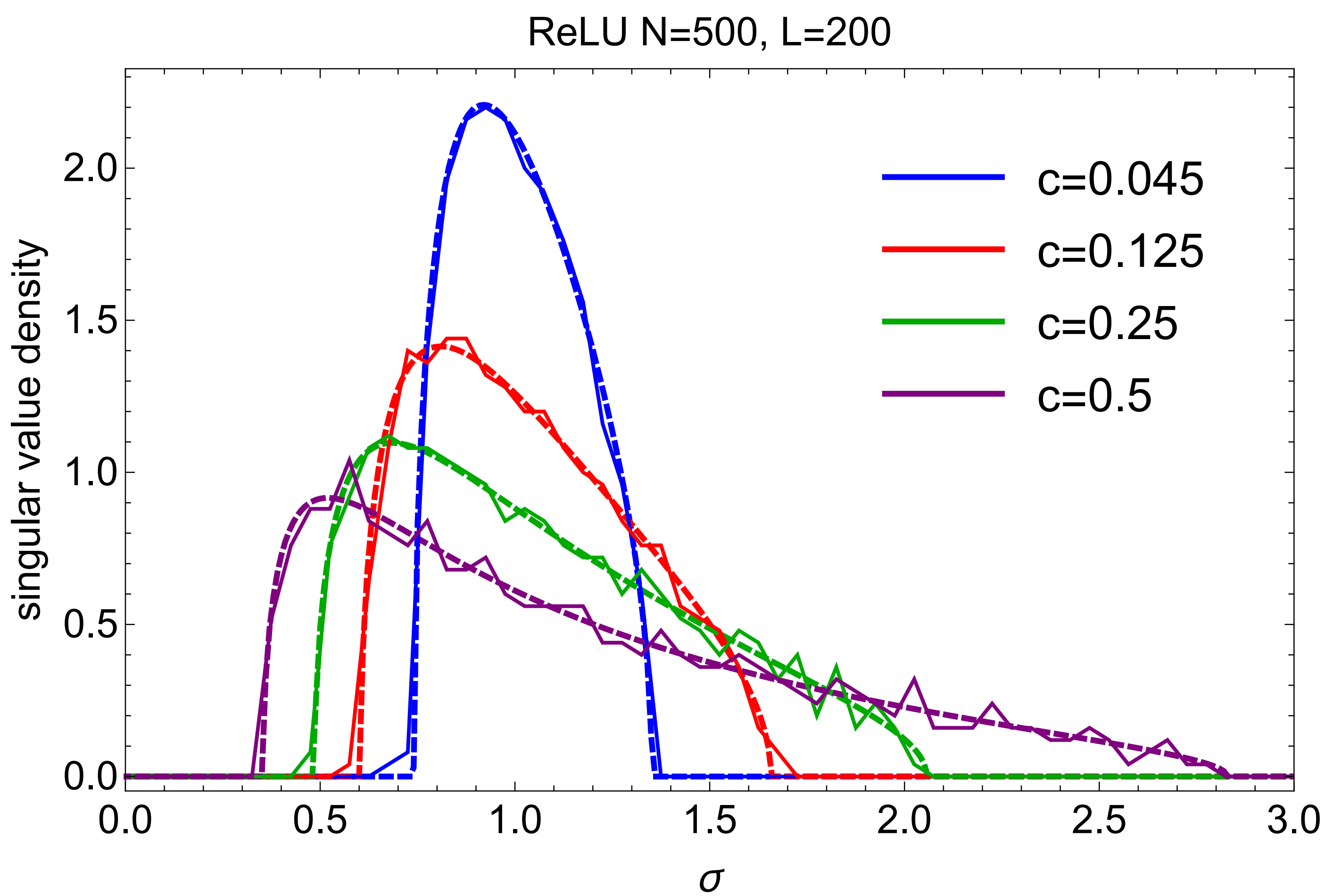


• What we find for ResNets

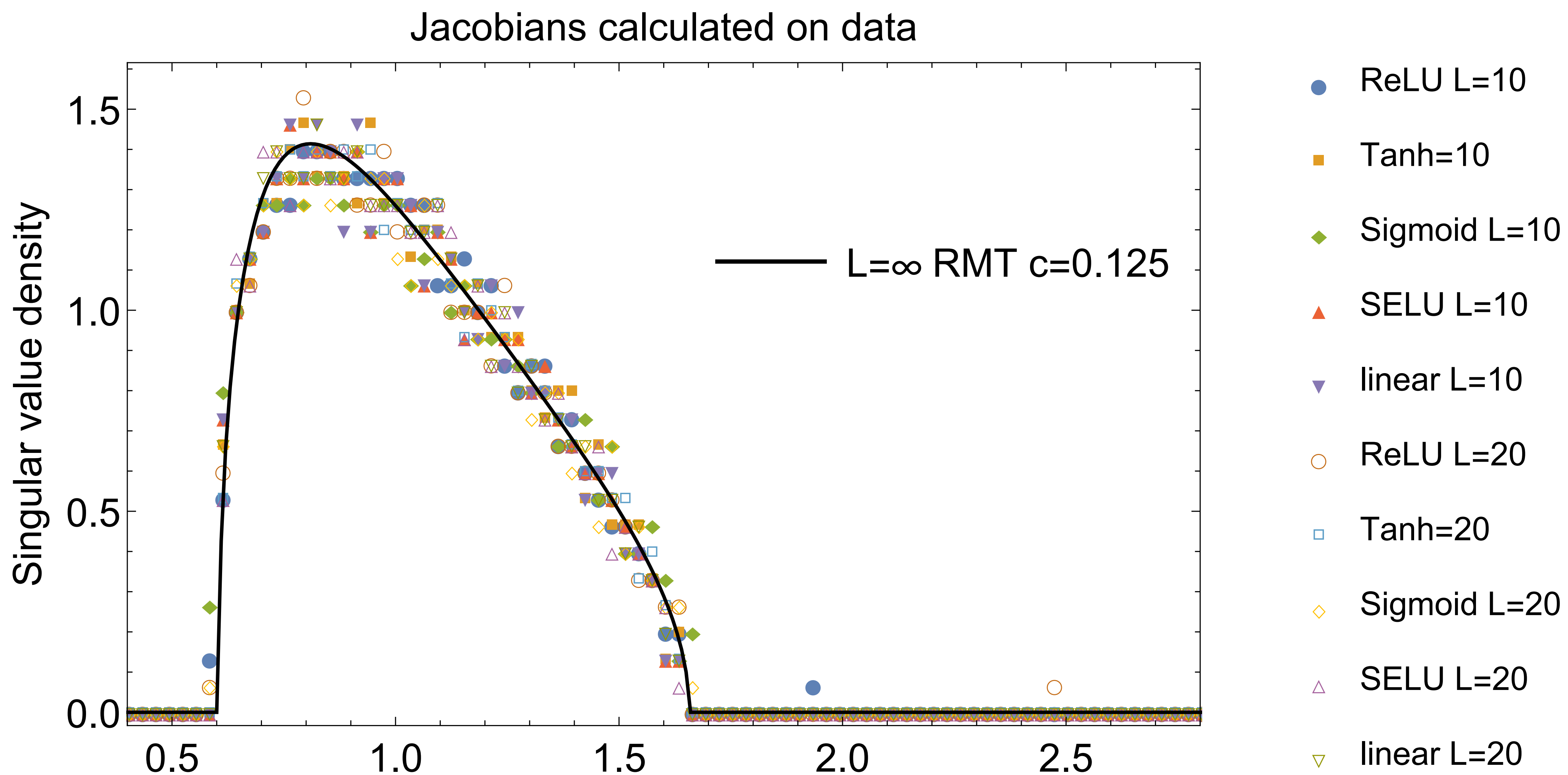
With a proper scaling of the variances of the weights, the result is a universal formula for the probability density of the singular values, depending on a single parameter c.



Corroborated with numerical experiments with neural networks with random inputs

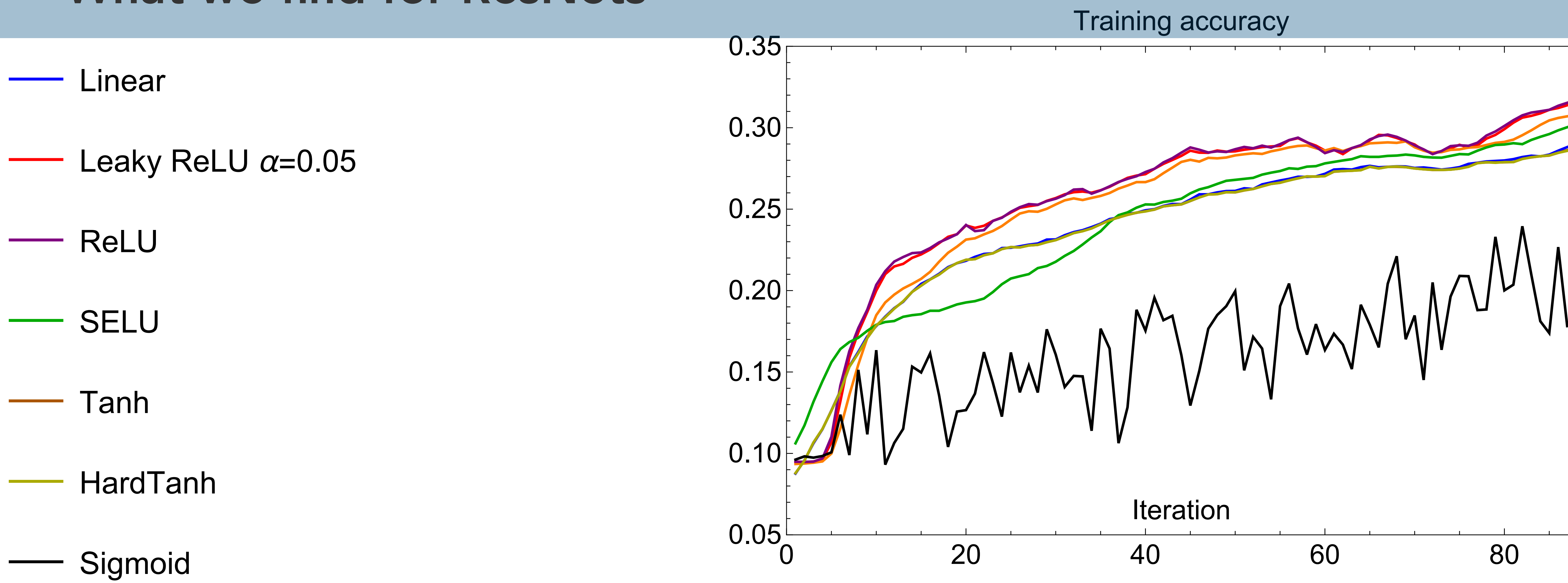


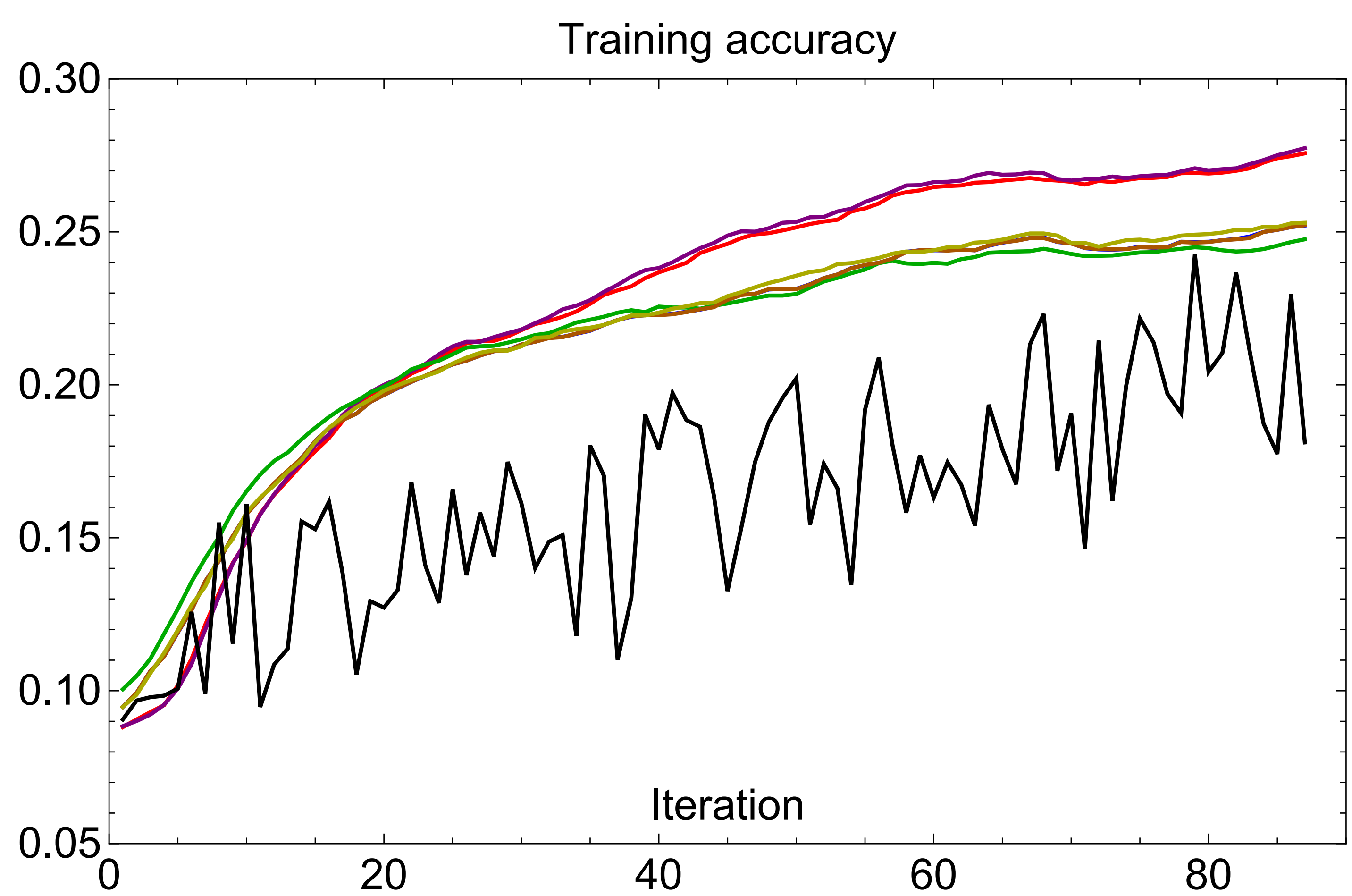
- **What we find for ResNets**



Corroborated with numerical experiments with neural networks initialized on the CIFAR-10 dataset.

• What we find for ResNets





These results allow us to eliminate the singular spectrum of the Jacobian treated as a confounding factor in experiments with the learning process of simple residual neural networks for different activation functions enabling meaningful comparisons.

To remember:

- Deep Neural Networks - powerful but many aspects not understood

- A proper initialization can allow efficient training for VERY deep feedforward NN. This is facilitated by Dynamical Isometry.

- For ResNets - Dynamical Isometry can be achieved for any activation functions, but remember about proper order of magnitude for variances.

arXiv:1809.08848

accepted for AISTATS 2019

NATIONAL SCIENCE CENTRE
POLAND

Random Matrices and Information Measures -
SONATA project number 2016/21/D/ST2/01142.

Thank you for your attention.

JAGIELLONIAN UNIVERSITY
IN KRAKÓW