

Landscape Connectivity and Dropout Stability of SGD Solutions for Over-parameterized Neural Networks



Alexander Shevchenko



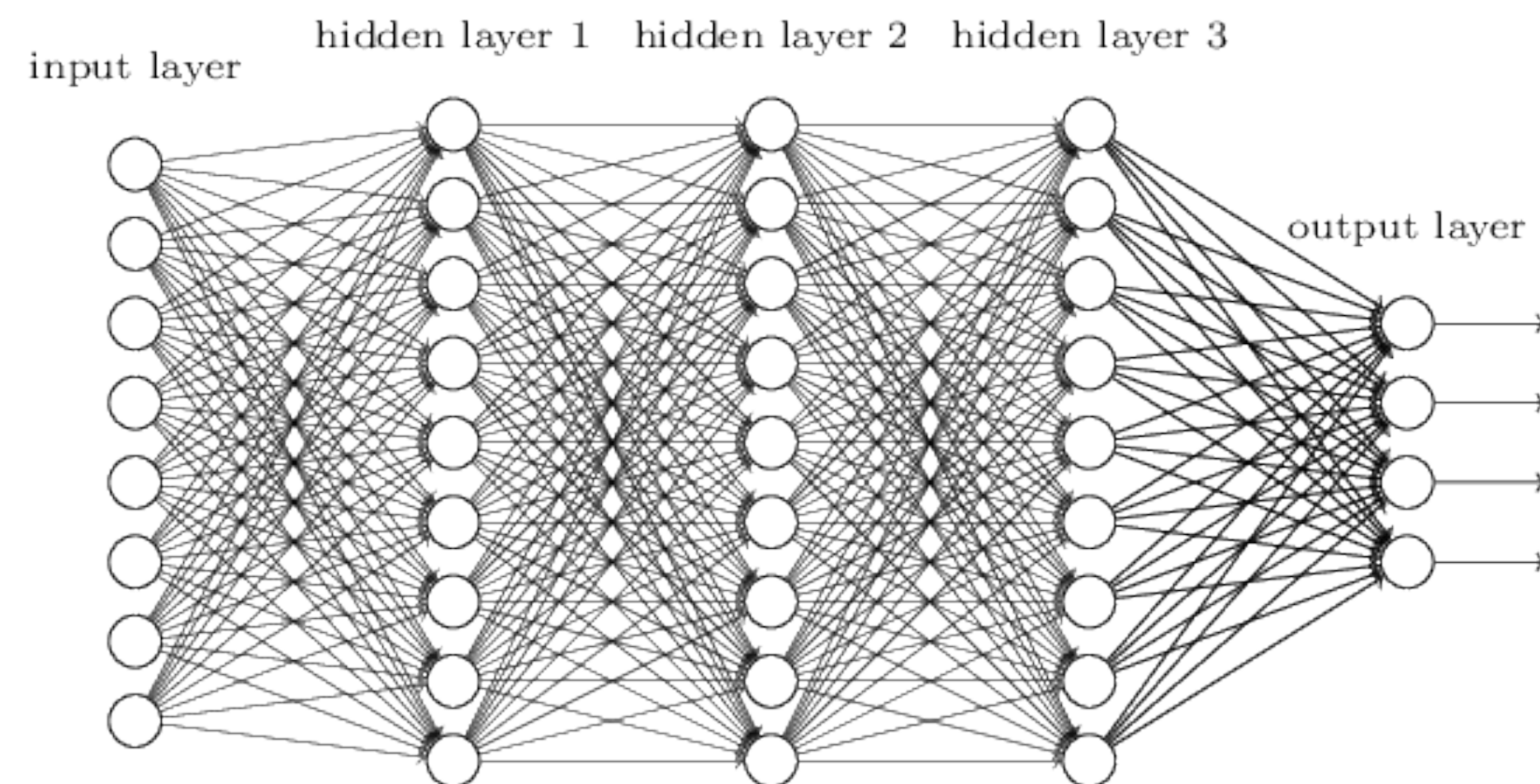
Marco Mondelli

Neural Network Training

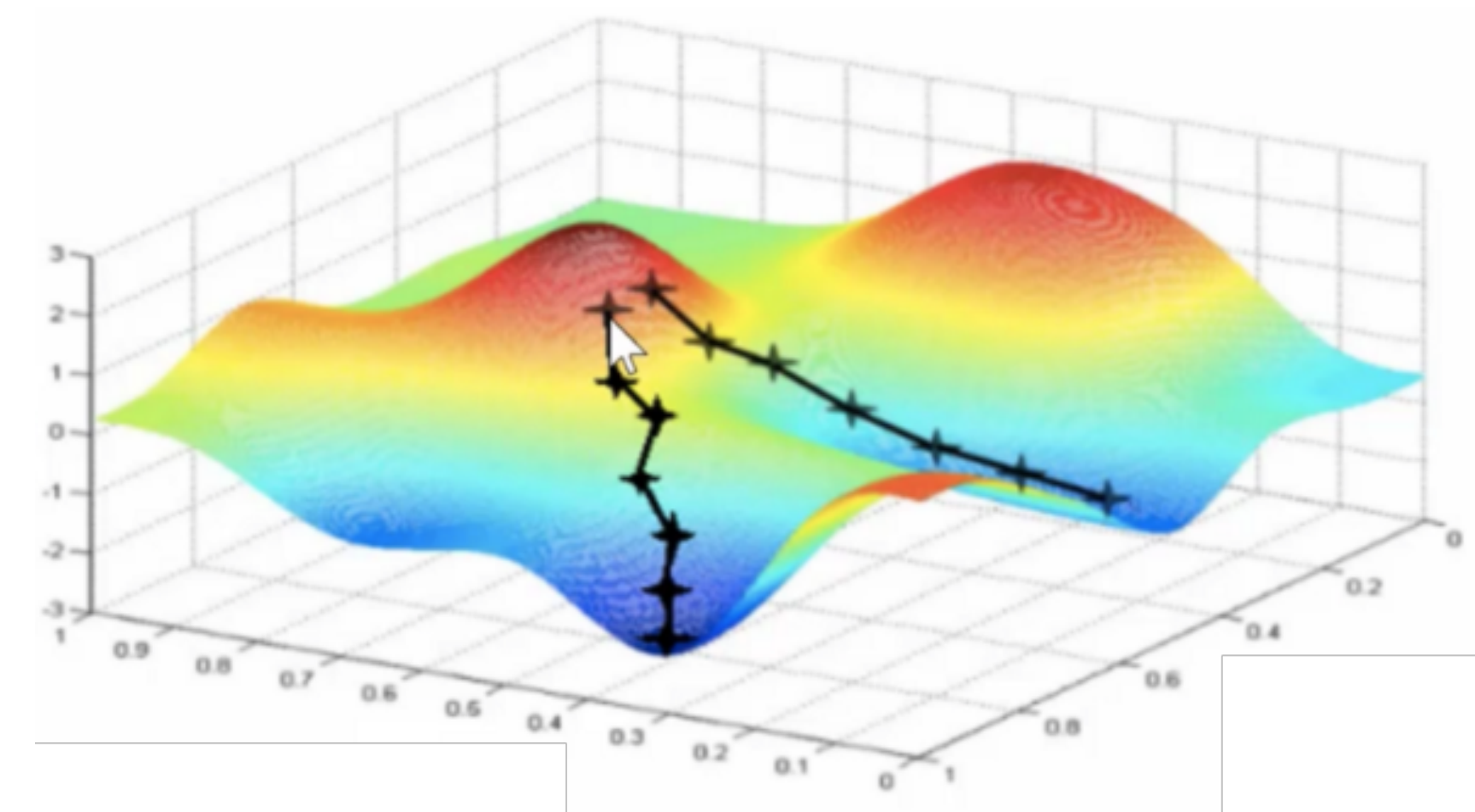
From theoretical perspective training of neural networks is difficult (NP-hardness, local/disconnected minima ...), but in practice works remarkably well!

Two key ingredients of success:

Over-parameterization

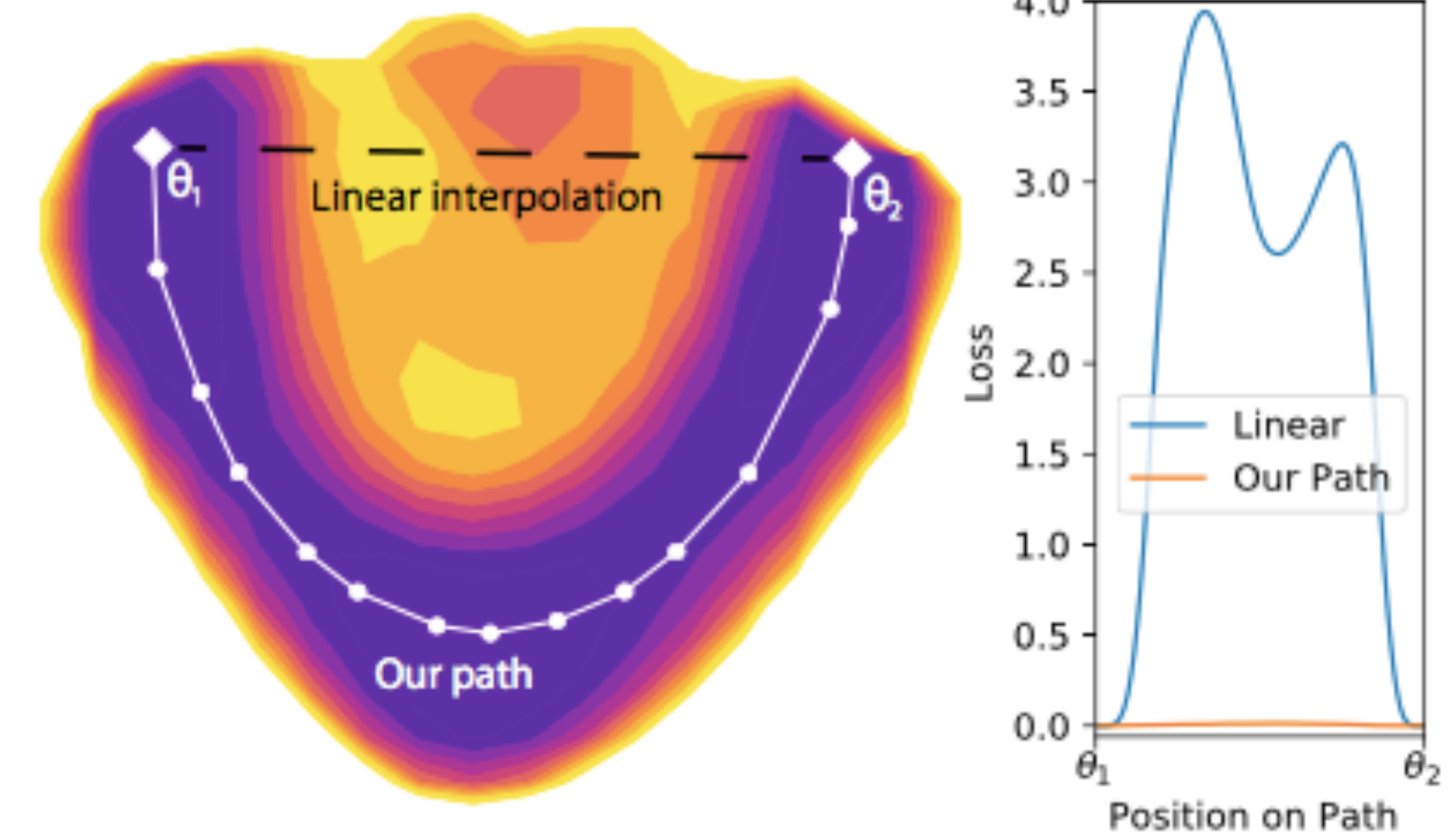


(Stochastic) gradient descent



Training Landscape is indeed NICE

- SGD minima connected via piecewise linear path with constant loss [Garipov et al., 2018; Draxler et al., 2018]
- Mode connectivity proved assuming properties of well-trained networks (dropout/noise stability) [Kuditipudi et al., 2019]



What do we show?

Theorem. (Informal) *As neural network grows wider the solutions obtained via SGD become increasingly more dropout stable and barriers between local optima disappear.*

Mean-field view: Two layers [Mei et al., 2019] Multiple layers [Araujo et al., 2019]

Quantitative bounds:

- **independent** of input dimension for two-layer networks, scale **linearly** for multiple layers
- change in loss scales with network width as $\sqrt{\frac{1}{\text{width}}}$
- number of training samples **is just required** to scale faster than the $\sqrt{\log(\text{width})}$

Related Work

- Local minima are globally optimal for deep linear networks and networks with more neurons than training samples
- Connected landscape if the number of neurons grows large (two-layer networks, energy gap exponential in input dimension)

The Loss Surface of Deep and Wide Neural Networks
Quynh Nguyen¹ Matthias Hein¹

Deep Learning without Poor Local Minima
Kenji Kawaguchi
Massachusetts Institute of Technology
kawaguch@mit.edu

Abstract
In this paper, we prove a conjecture published in 1989 and an open problem announced at the Conference on Learning Theory. With no unrealistic assumption, we first prove the following: squared loss function of deep linear neural networks with widths: 1) the function is non-convex and non-concave, 2) does not encounter problems with suboptimal local minima. However, as the authors admit themselves in (Goodfellow et al., 2015), the reason for this might be that there is a connection between the fact that these networks have

TOPOLOGY AND GEOMETRY OF HALF-RECTIFIED NETWORK OPTIMIZATION
C. Daniel Freeman
Department of Physics,
UC Berkeley, Berkeley, CA 94720, USA
daniel.freeman@berkeley.edu
Joan Bruna^{*}
Courant Institute of Mathematical Sciences
New York University, 251 Mercer St, New York, NY 10011, USA

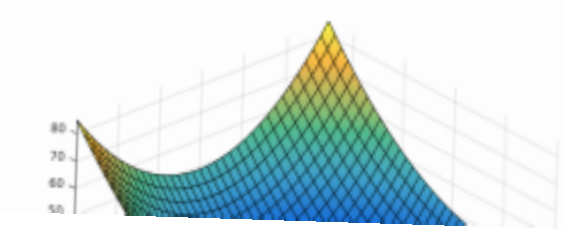
Optimization Landscape and Expressivity of Deep CNNs
Quynh Nguyen¹ Matthias Hein²

Spurious Valleys in Two-layer Neural Network Optimization Landscapes
Luca Venturi^{*1}, Afonso S. Bandeira^{†1,2}, and Joan Bruna^{†1,2}
¹Courant Institute of Mathematics
²Center for Data Science

On Connected Sublevel Sets in Deep Learning
Quynh Nguyen¹

Abstract
This paper shows that every sublevel set of the loss function of a class of deep over-parameterized neural networks with sigmoidal layers

Table 1. The maximum width of all layers in several state-of-the-art CNN architectures compared with the size of ImageNet dataset ($N \approx 1200K$). All numbers are lower bounds on the true width.



Strong assumptions on the model and poor scaling of parameters 😞

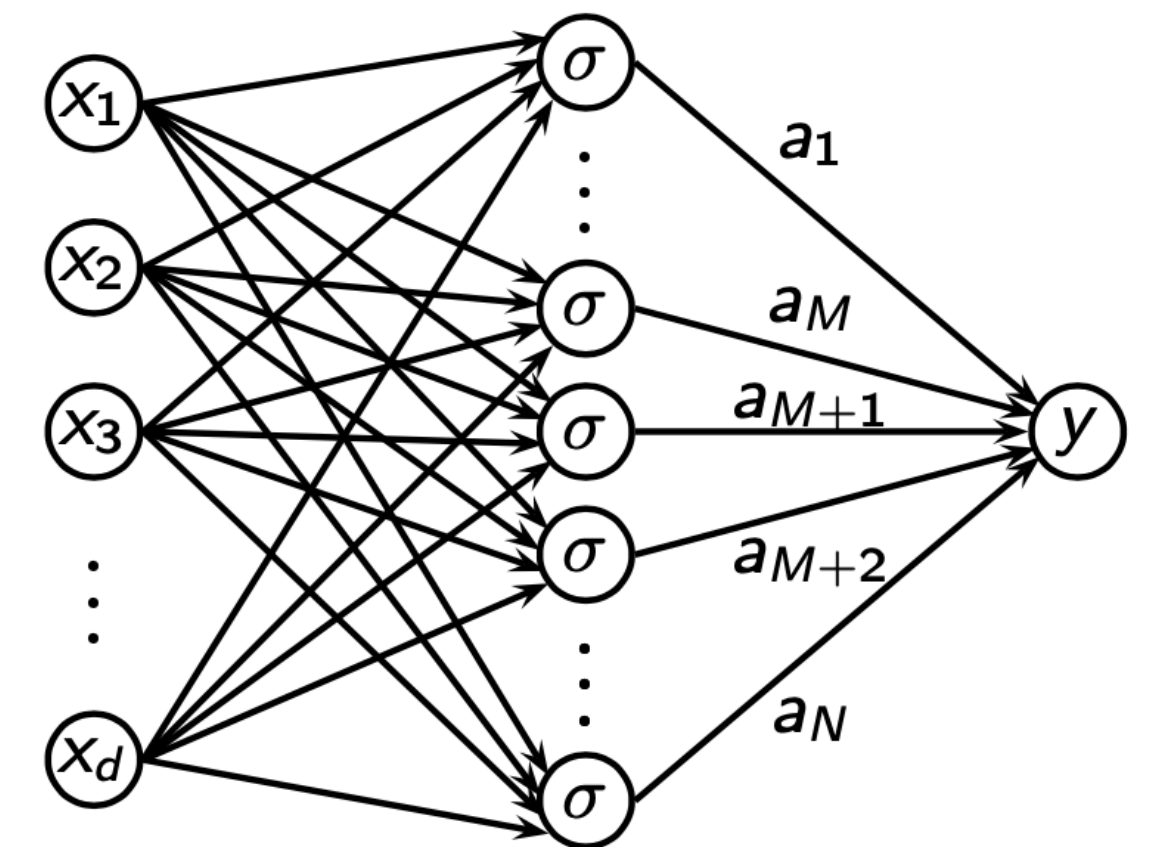
Warm-up: Two Layer Networks

Data: $\{ (\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n) \} \sim_{\text{i.i.d.}} \mathbb{P}(\mathbb{R}^d \times \mathbb{R})$

Model: $\hat{y}_N(\mathbf{x}, \boldsymbol{\theta}) = \frac{1}{N} \sum_{i=1}^N a_i \sigma(\mathbf{x}; \mathbf{w}_i)$

Goal: Minimize loss $L_N(\boldsymbol{\theta}) = \mathbb{E} \left\{ \left(y - \frac{1}{N} \sum_{i=1}^N a_i \sigma(\mathbf{x}; \mathbf{w}_i) \right)^2 \right\}, \boldsymbol{\theta} = (\mathbf{w}, a)$

Online SGD: $\boldsymbol{\theta}^{k+1} = \boldsymbol{\theta}^k + \alpha N \nabla_{\boldsymbol{\theta}^k} \left(\left(y_k - \frac{1}{N} \sum_{i=1}^N a_i^k \sigma(\mathbf{x}_k; \mathbf{w}_i^k) \right)^2 \right)$

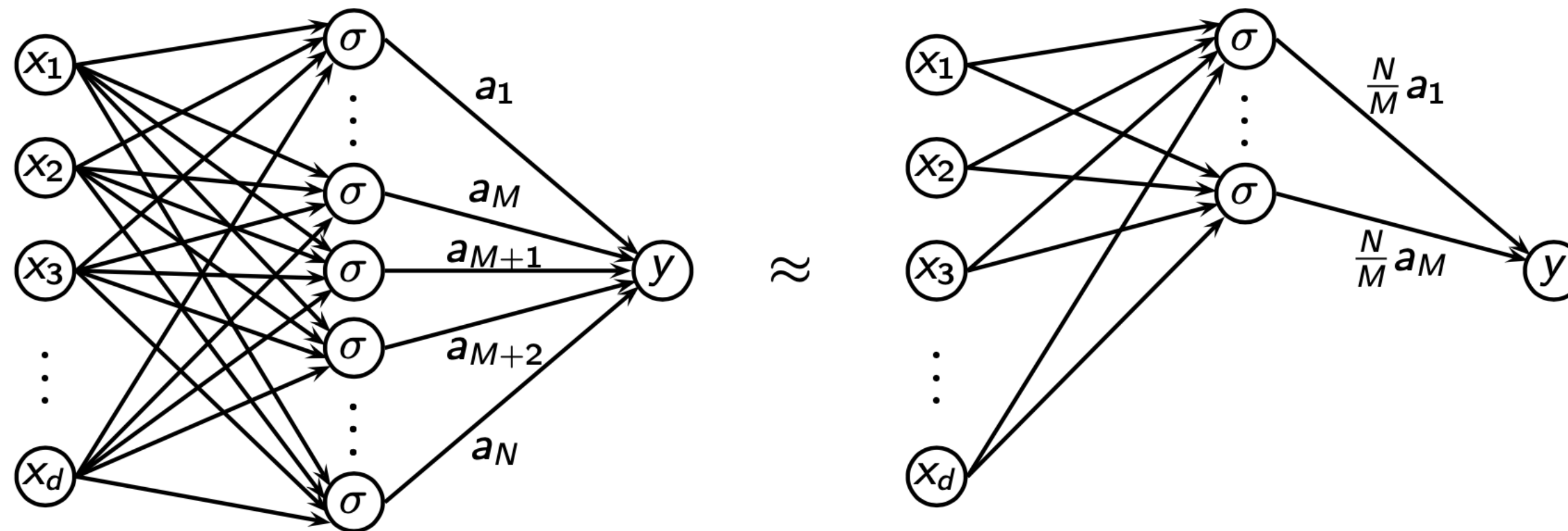


- y bounded, $\nabla_{\mathbf{w}} \sigma(\mathbf{x}, \mathbf{w})$ sub-gaussian
- σ bounded and differentiable, $\nabla \sigma$ bounded and Lipschitz
- initialization of a_i with bounded support

Recap: Dropout Stability

$$L_M(\boldsymbol{\theta}) = \mathbb{E} \left\{ \left(y - \frac{1}{M} \sum_{i=1}^M a_i \sigma(\mathbf{x}; \mathbf{w}_i) \right)^2 \right\}$$

$\boldsymbol{\theta}$ is ε_D - dropout stable if $|L_N(\boldsymbol{\theta}) - L_M(\boldsymbol{\theta})| \leq \varepsilon_D$

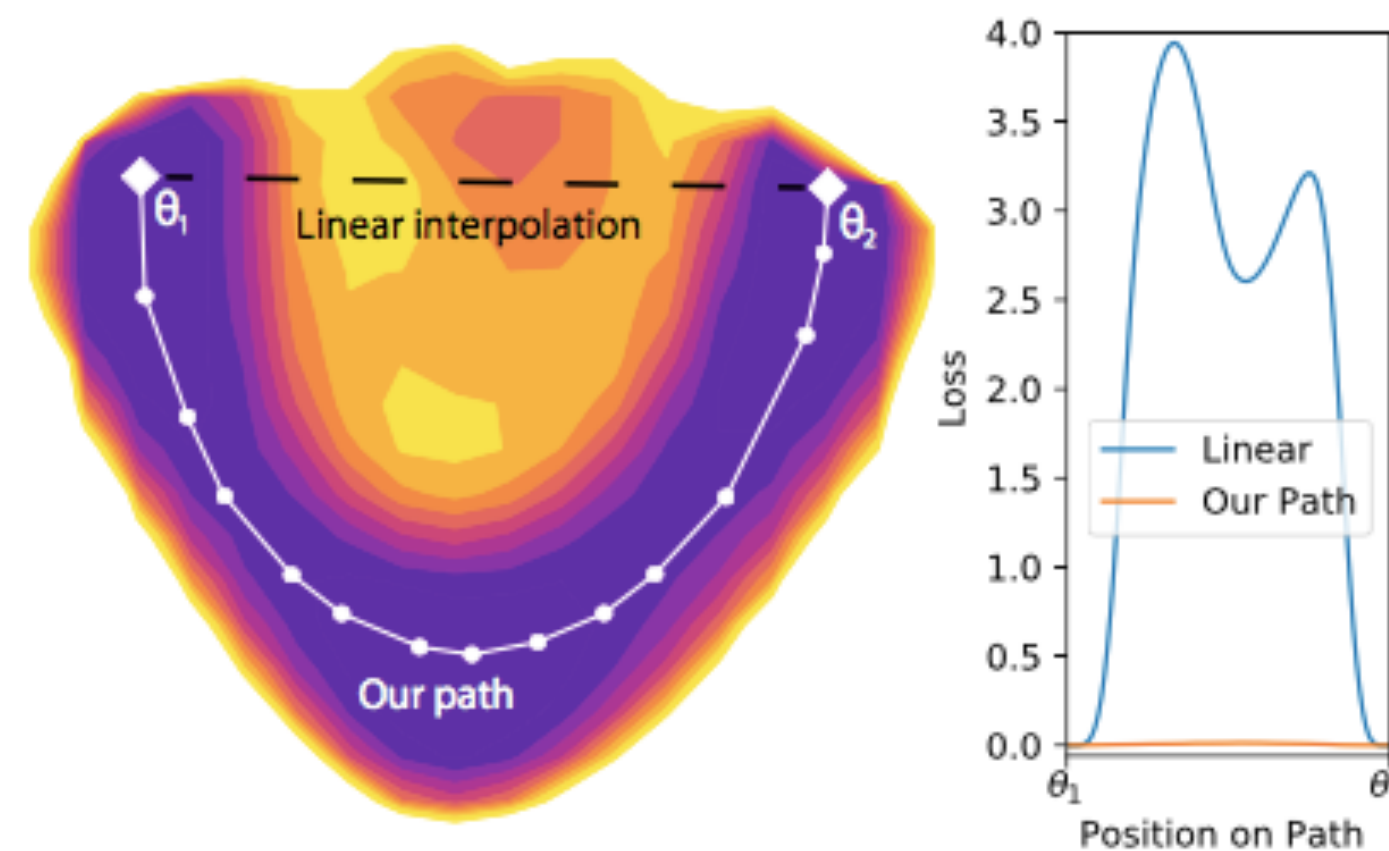


Recap: Dropout Stability and Connectivity

$$L_M(\boldsymbol{\theta}) = \mathbb{E} \left\{ \left(y - \frac{1}{M} \sum_{i=1}^M a_i \sigma(\mathbf{x}; \mathbf{w}_i) \right)^2 \right\}$$

$\boldsymbol{\theta}$ is ε_D - dropout stable if $|L_N(\boldsymbol{\theta}) - L_M(\boldsymbol{\theta})| \leq \varepsilon_D$

$\boldsymbol{\theta}$ and $\boldsymbol{\theta}'$ are ε_C - connected if there exists a continuous path connecting them where the loss does not increase more than ε_C



Main Results: Dropout Stability

- $N = \#$ neurons of full network
- $M = \#$ neurons after dropout
- $\alpha =$ step size of SGD
- $D =$ dimension of weights

Theorem

Let θ^k be obtained after k SGD iterations. Then, with probability $1 - e^{-z^2}$, for all $k \in [T/\alpha]$, θ^k is ε_D -dropout stable with

$$\varepsilon_D = Ke^{KT^3} \left(\frac{\sqrt{\log M} + z}{\sqrt{M}} + \sqrt{\alpha}(\sqrt{D + \log N} + z) \right).$$

Main Results: Dropout Stability

- $N = \#$ neurons of full network
- $M = \#$ neurons after dropout
- $\alpha =$ step size of SGD
- $D =$ dimension of weights

Theorem

Let θ^k be obtained after k SGD iterations. Then, with probability $1 - e^{-z^2}$, for all $k \in [T/\alpha]$, θ^k is ε_D -dropout stable with

$$\varepsilon_D = Ke^{KT^3} \left(\frac{\sqrt{\log M} + z}{\sqrt{M}} + \sqrt{\alpha}(\sqrt{D + \log N} + z) \right).$$

Change in loss scales as $\sqrt{\frac{\log M}{M}} + \sqrt{\alpha(D + \log N)}$

Main Results: Dropout Stability

- $N = \#$ neurons of full network
- $M = \#$ neurons after dropout
- $\alpha =$ step size of SGD
- $D =$ dimension of weights

Theorem

Let θ^k be obtained after k SGD iterations. Then, with probability $1 - e^{-z^2}$, for all $k \in [T/\alpha]$, θ^k is ε_D -dropout stable with

$$\varepsilon_D = Ke^{KT^3} \left(\frac{\sqrt{\log M} + z}{\sqrt{M}} + \sqrt{\alpha}(\sqrt{D + \log N} + z) \right).$$

- Loss change vanishes as $\alpha \ll \left(\sqrt{D + \log N}\right)^{-1}$ and $M \gg 1$
- M does not need to scale with N or D

Main Results: Connectivity

Theorem

Let θ^k be obtained after k SGD iterations using $\{(\mathbf{x}_j, y_j)\}_{j=0}^k \sim \mathbb{P}$, and $(\theta')^{k'}$ after k' SGD iterations using $\{(\mathbf{x}'_j, y'_j)\}_{j=0}^{k'} \sim \mathbb{P}$. Then, with probability $1 - e^{-z^2}$, for all $k \in [T/\alpha]$ and $k' \in [T'/\alpha]$, θ^k and $(\theta')^{k'}$ are ε_C -connected with

$$\varepsilon_C = Ke^{K \max(T, T')^3} \left(\frac{\sqrt{\log N} + z}{\sqrt{N}} + \sqrt{\alpha}(\sqrt{D + \log N} + z) \right).$$

Main Results: Connectivity

Theorem

Let θ^k be obtained after k SGD iterations using $\{(\mathbf{x}_j, y_j)\}_{j=0}^k \sim \mathbb{P}$, and $(\theta')^{k'}$ after k' SGD iterations using $\{(\mathbf{x}'_j, y'_j)\}_{j=0}^{k'} \sim \mathbb{P}$. Then, with probability $1 - e^{-z^2}$, for all $k \in [T/\alpha]$ and $k' \in [T'/\alpha]$, θ^k and $(\theta')^{k'}$ are ε_C -connected with

$$\varepsilon_C = Ke^{K \max(T, T')^3} \left(\frac{\sqrt{\log N} + z}{\sqrt{N}} + \sqrt{\alpha}(\sqrt{D + \log N} + z) \right).$$

- Change in loss scales as $\sqrt{\frac{\log N}{N}} + \sqrt{\alpha(D + \log N)}$

Main Results: Connectivity

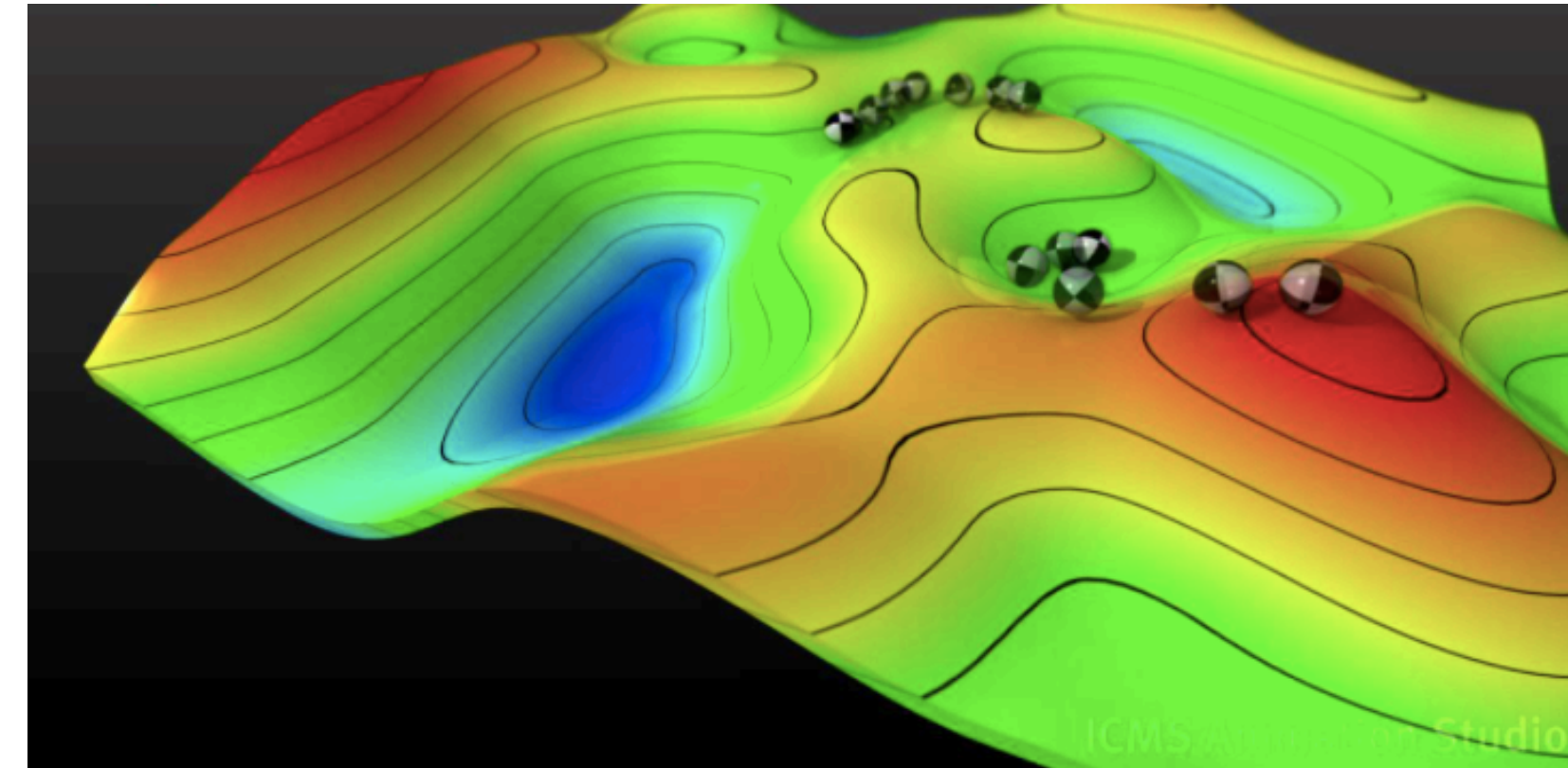
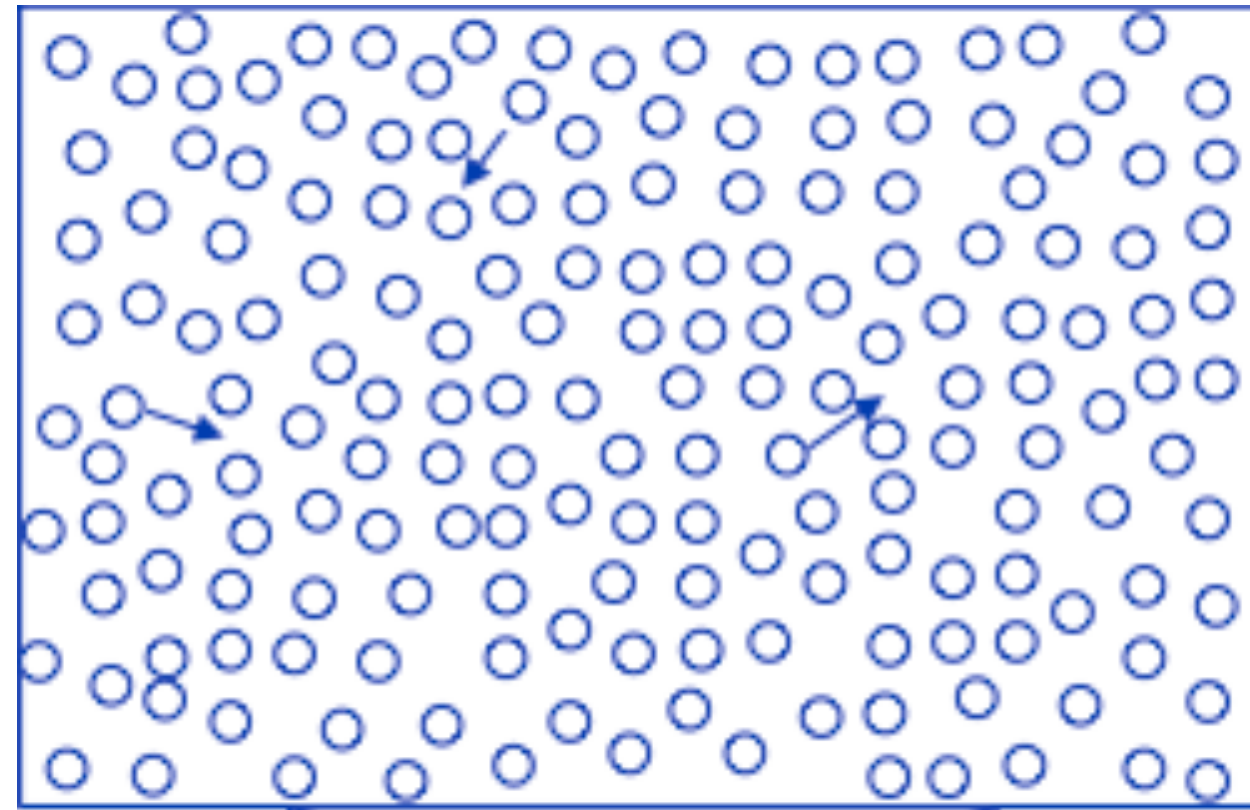
Theorem

Let θ^k be obtained after k SGD iterations using $\{(\mathbf{x}_j, y_j)\}_{j=0}^k \sim \mathbb{P}$, and $(\theta')^{k'}$ after k' SGD iterations using $\{(\mathbf{x}'_j, y'_j)\}_{j=0}^{k'} \sim \mathbb{P}$. Then, with probability $1 - e^{-z^2}$, for all $k \in [T/\alpha]$ and $k' \in [T'/\alpha]$, θ^k and $(\theta')^{k'}$ are ε_C -connected with

$$\varepsilon_C = Ke^{K \max(T, T')^3} \left(\frac{\sqrt{\log N} + z}{\sqrt{N}} + \sqrt{\alpha}(\sqrt{D + \log N} + z) \right).$$

- Change in loss scales as $\sqrt{\frac{\log N}{N}} + \sqrt{\alpha(D + \log N)}$
- Can connect SGD solutions obtained from different training data (but same data distribution) and different initialization

Proof Idea



Discrete dynamics of SGD

Continuous dynamics of gradient flow

- θ^k close to N i.i.d. particles that evolve with gradient flow
- $L_N(\theta)$ and $L_M(\theta)$ concentrate to the same limit
- Dropout stability with $M = N/2 \Rightarrow$ connectivity

Multilayer Case: Setup

Data: $\left\{ (\mathbf{x}_1, \mathbf{y}_1), \dots, (\mathbf{x}_n, \mathbf{y}_n) \right\} \sim_{\text{i.i.d.}} \mathbb{P} \left(\mathbb{R}^{d_x} \times \mathbb{R}^{d_y} \right)$

Model: $\hat{\mathbf{y}}_N(\mathbf{x}, \boldsymbol{\theta}) = \frac{1}{N} \mathbf{W}_{L+1} \sigma_L \left(\dots \left(\frac{1}{N} \mathbf{W}_2 \sigma_1 \left(\mathbf{W}_1 \mathbf{x} \right) \right) \dots \right)$

Goal: Minimize loss $L_N(\boldsymbol{\theta}) = \mathbb{E} \left\{ \left\| \mathbf{y} - \hat{\mathbf{y}}_N(\mathbf{x}, \boldsymbol{\theta}) \right\|^2 \right\}$

Online SGD: $\boldsymbol{\theta}^{k+1} = \boldsymbol{\theta}^k + \alpha N^2 \nabla_{\boldsymbol{\theta}^k} \left\| \mathbf{y}_k - \hat{\mathbf{y}}_N(\mathbf{x}_k, \boldsymbol{\theta}^k) \right\|^2$

- \mathbf{y} bounded
- σ_ℓ bounded and differentiable, $\nabla \sigma_\ell$ bounded and Lipschitz
- initialization with bounded support
- \mathbf{W}_1 and \mathbf{W}_{L+1} stay fixed (random features)

Multilayer Case: Dropout Stability

Dropout stability: loss does not change much if we remove part of neurons from each layer (and suitably rescale remaining neurons).

Multilayer Case: Dropout Stability

$L_M(\boldsymbol{\theta})$:= loss when we keep at most M neurons per layer

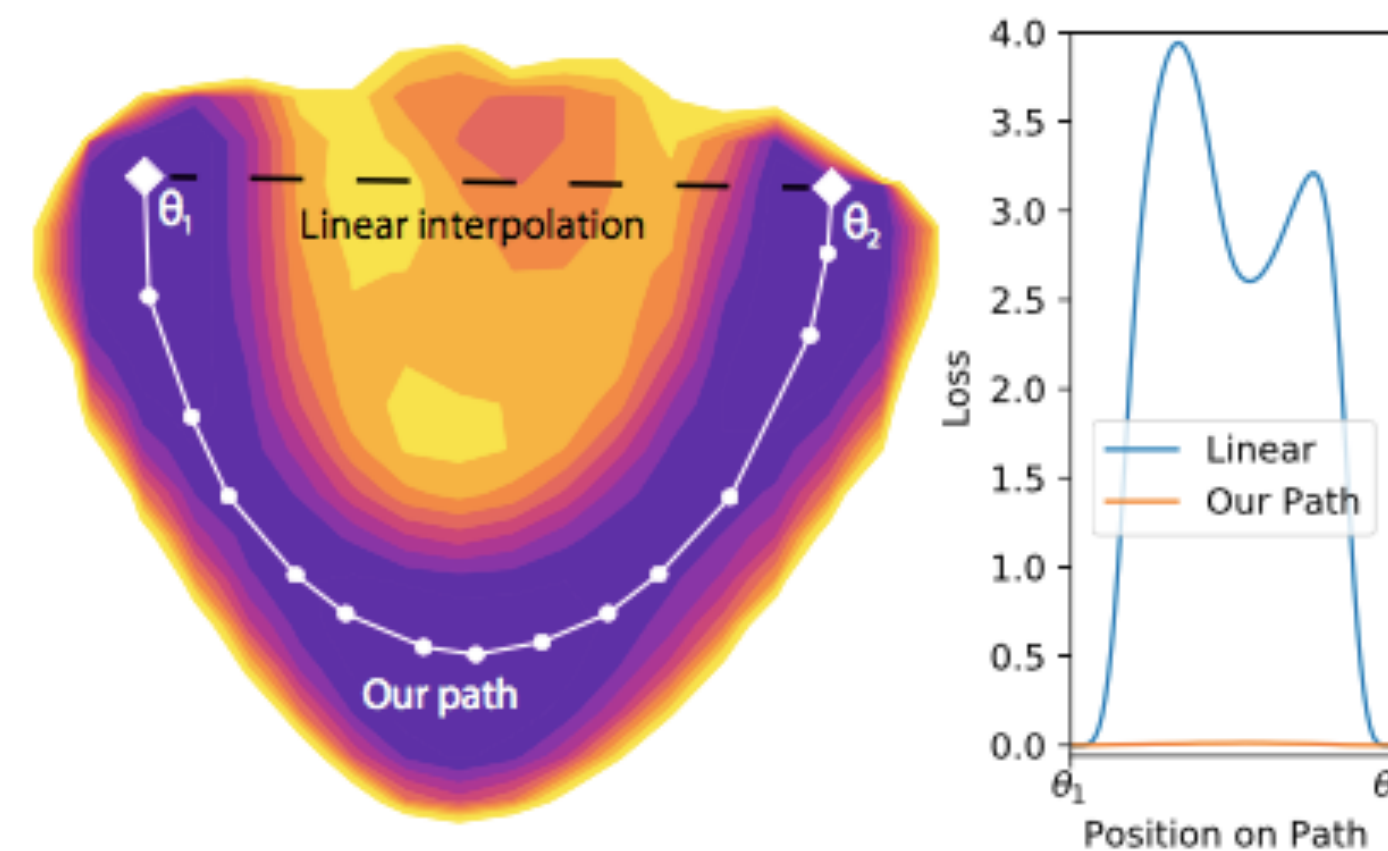
$\boldsymbol{\theta}$ is ε_D - dropout stable if $|L_N(\boldsymbol{\theta}) - L_M(\boldsymbol{\theta})| \leq \varepsilon_D$

Multilayer Case: Dropout Stability and Connectivity

$L_M(\theta)$:= loss when we keep at most M neurons per layer

θ is ε_D - dropout stable if $|L_N(\theta) - L_M(\theta)| \leq \varepsilon_D$

θ and θ' are ε_C - connected if there exists a continuous path connecting them where the loss does not increase more than ε_C



Multilayer Case: Results

- $N = \#$ neurons per layer of full network
- $M = \max. \#$ neurons per layer after dropout
- $\alpha =$ step size of SGD
- $D = \max(d_x, d_y)$

Theorem

Let θ^k be obtained after k SGD iterations, with $k = T/\alpha$. Then, w. p. $1 - e^{-z^2}$, θ^k is ε_D -dropout stable with

$$\varepsilon_D = K(T, L) \left(\frac{\sqrt{D} + z}{\sqrt{M}} + \frac{\sqrt{\log N}}{\sqrt{N}} + \sqrt{\alpha}(\sqrt{D + \log N} + z) \right).$$

Multilayer Case: Results

- $N = \#$ neurons per layer of full network
- $M = \max. \#$ neurons per layer after dropout
- $\alpha = \text{step size of SGD}$
- $D = \max(d_x, d_y)$

Theorem

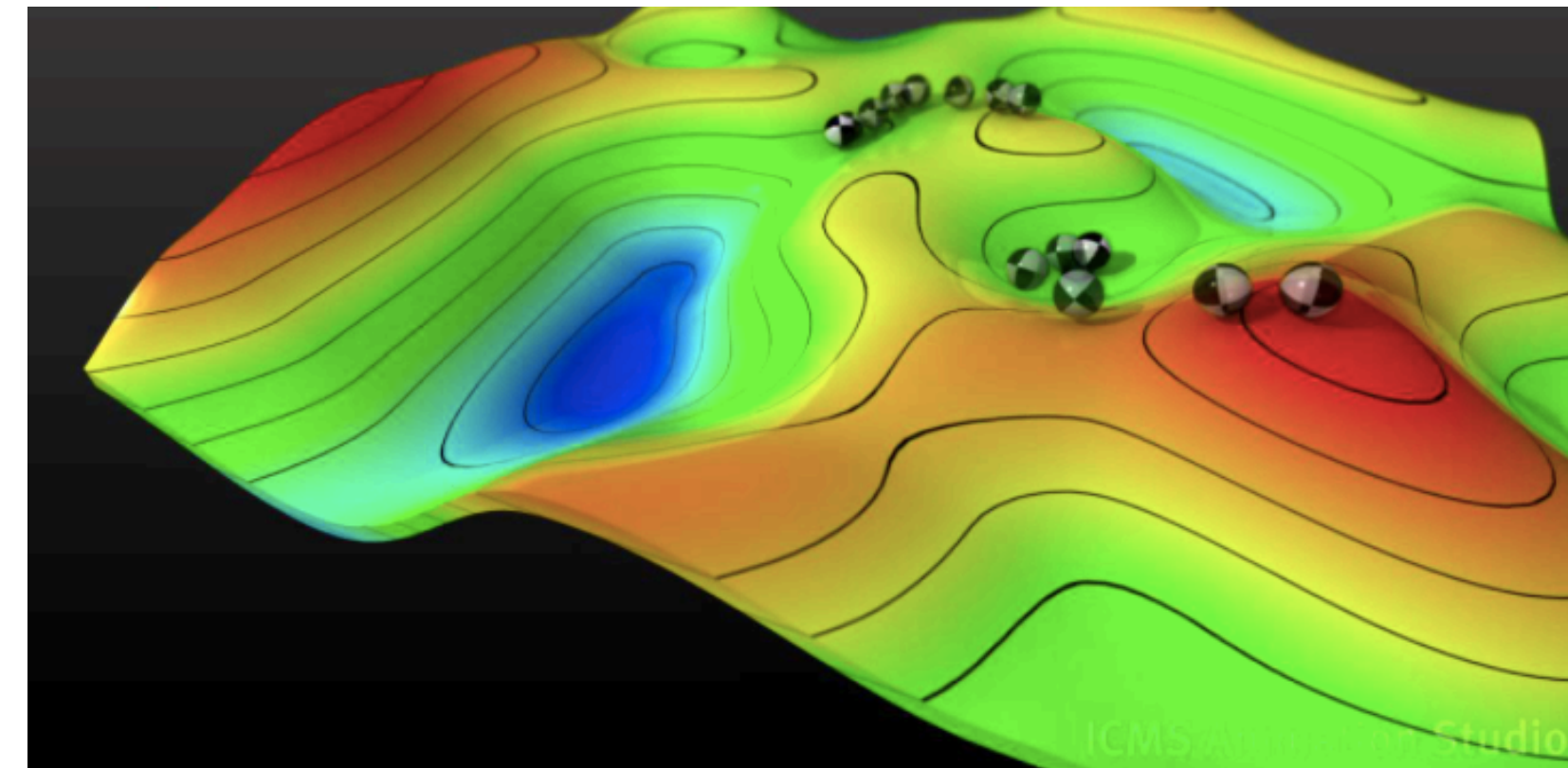
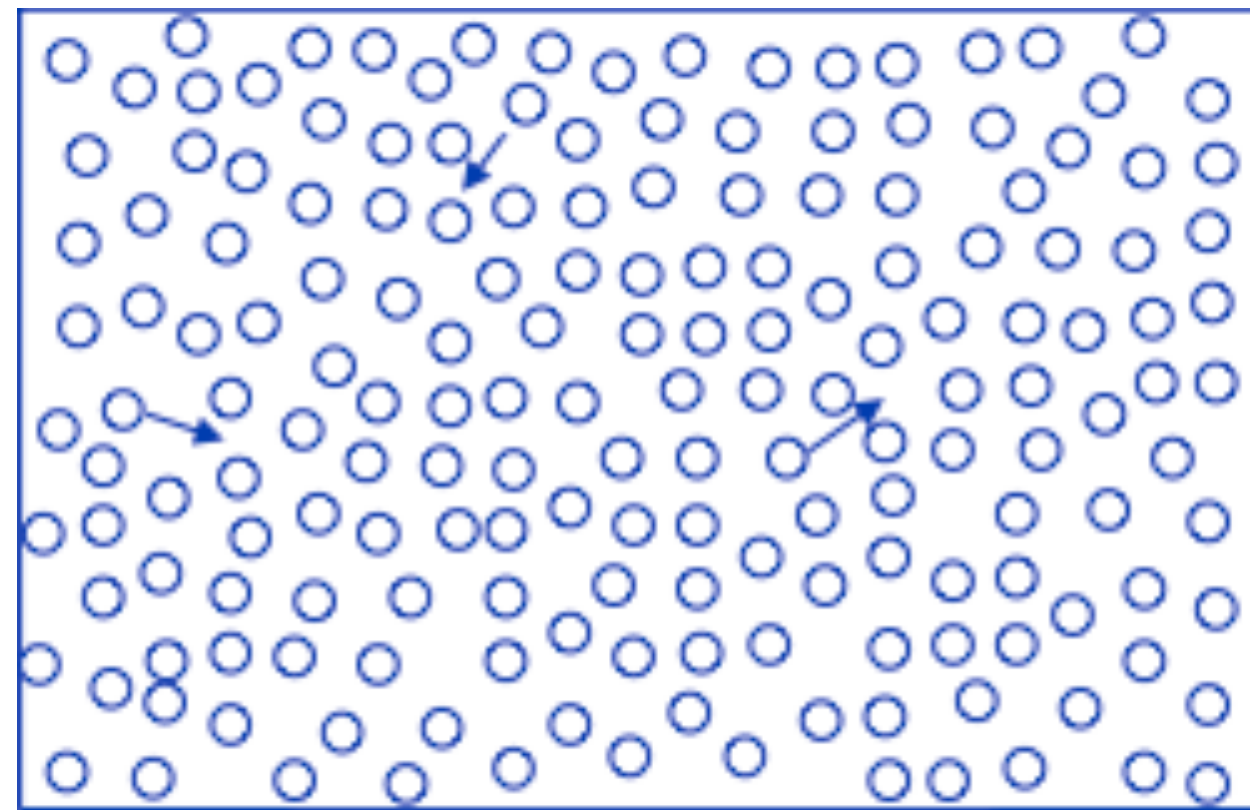
Let θ^k be obtained after k SGD iterations, with $k = T/\alpha$. Then, w. p. $1 - e^{-z^2}$, θ^k is ε_D -dropout stable with

$$\varepsilon_D = K(T, L) \left(\frac{\sqrt{D} + z}{\sqrt{M}} + \frac{\sqrt{\log N}}{\sqrt{N}} + \sqrt{\alpha}(\sqrt{D + \log N} + z) \right).$$

Let $(\theta')^{k'}$ be obtained after k' SGD iterations, with $k' = T'/\alpha$. Then, w. p. $1 - e^{-z^2}$, θ^k and $(\theta')^{k'}$ are ε_C -connected with

$$\varepsilon_C = K(T, T', L) \left(\frac{\sqrt{D + \log N} + z}{\sqrt{N}} + \sqrt{\alpha}(\sqrt{D + \log N} + z) \right).$$

Proof Challenges



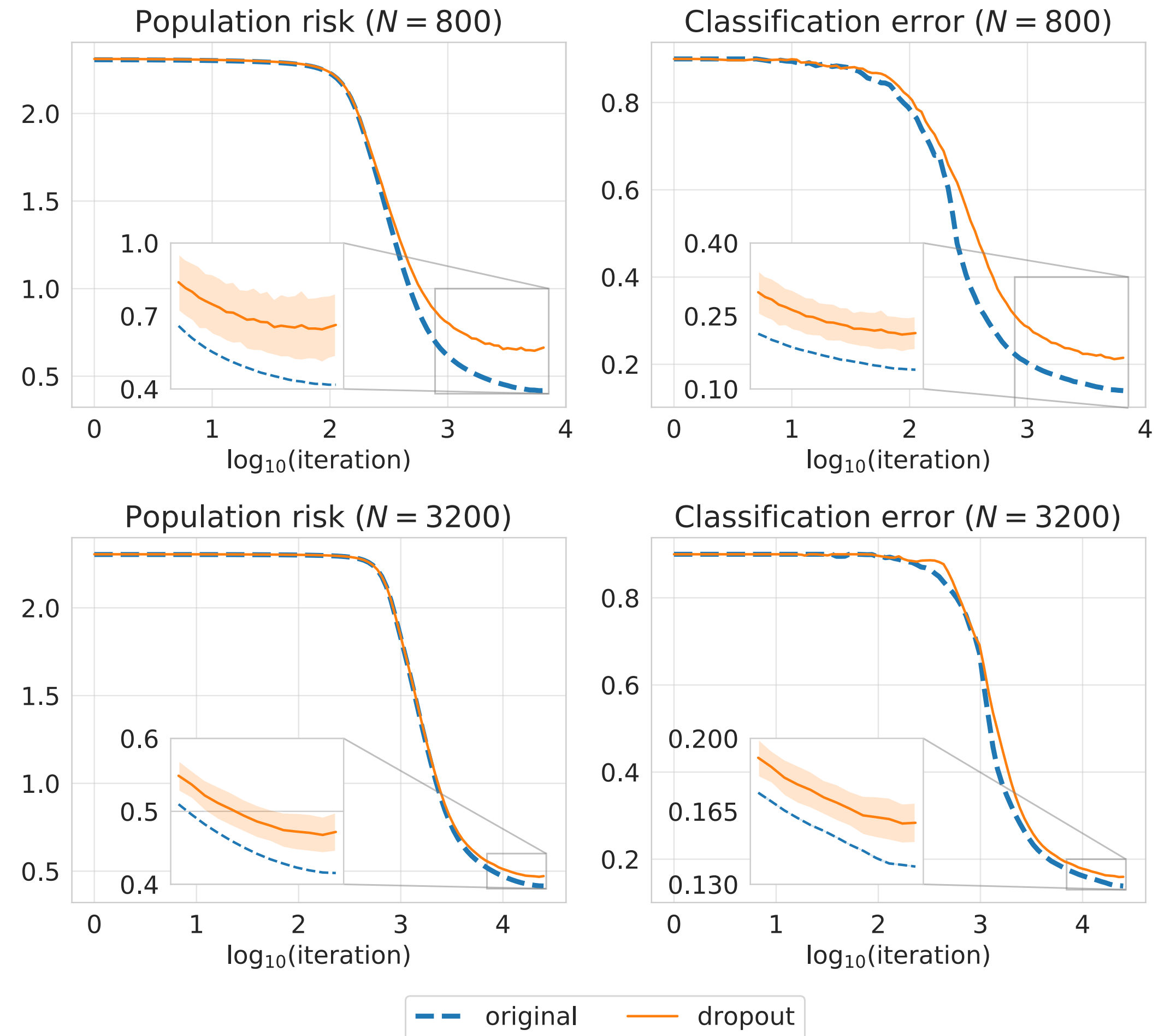
Discrete dynamics of SGD

Continuous dynamics of gradient flow

- Ideal particles are no longer independent (weights in different layers are correlated)
- Bound on norm of weights during the training
- Bound **maximum** distance between SGD and ideal particles ([Araujo et al., 2019] bounds the **average** distance)

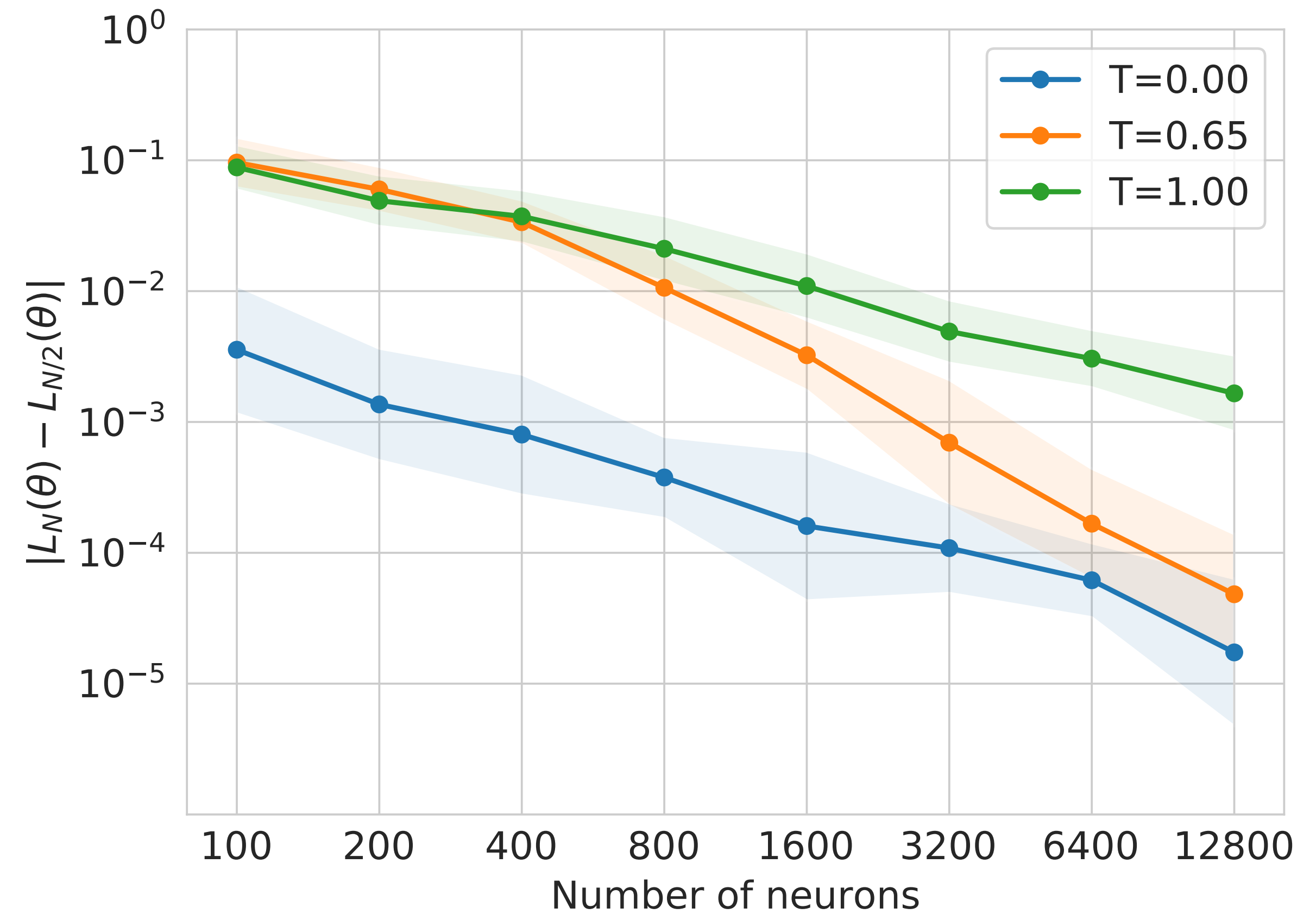
Numerical Results

- CIFAR-10 dataset
- Pretrained VGG-16 features
- # layers = 3
- Keep half of neurons



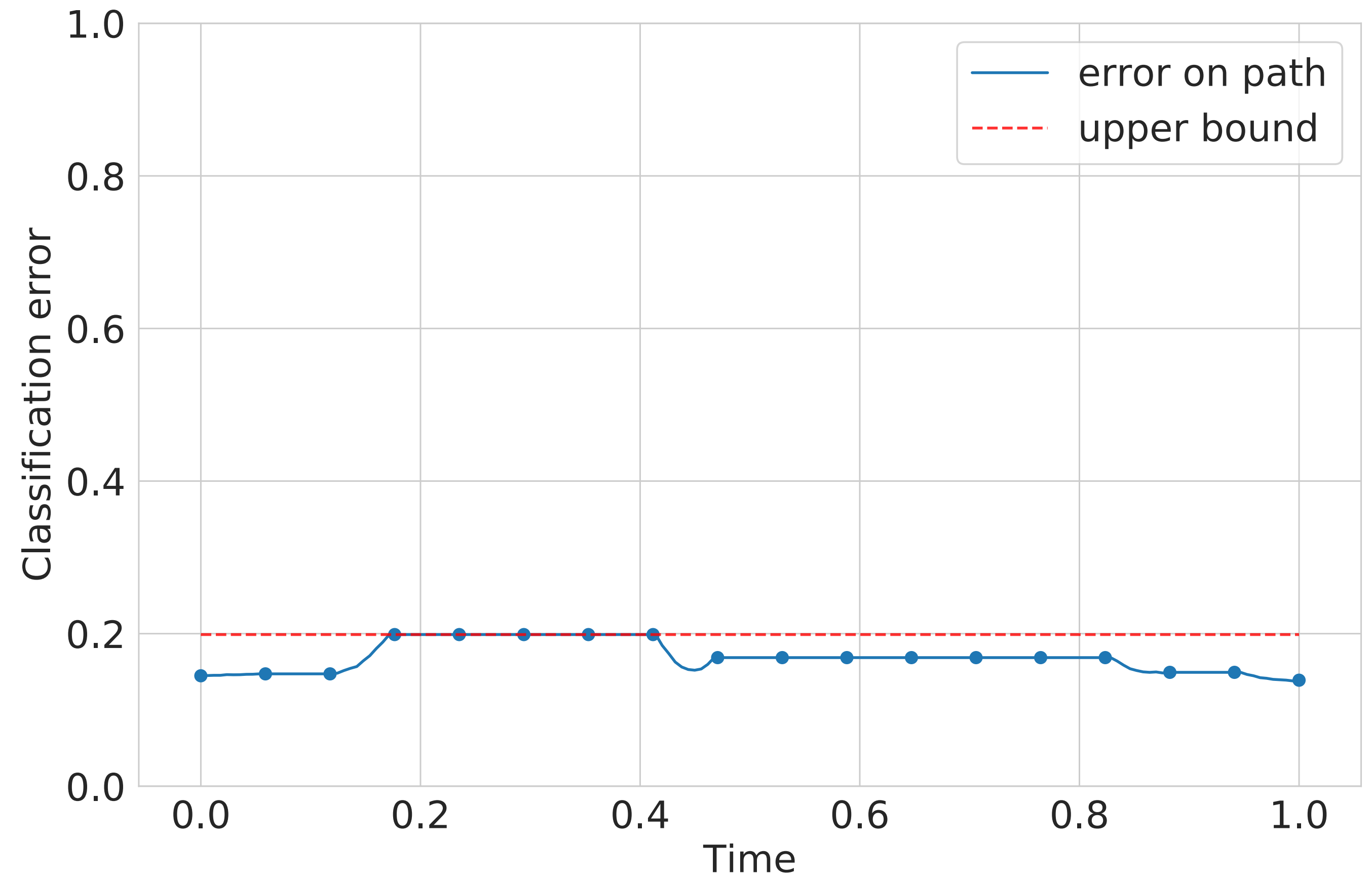
Numerical Results

- CIFAR-10 dataset
- Pretrained VGG-16 features
- # layers = 3
- Keep half of neurons

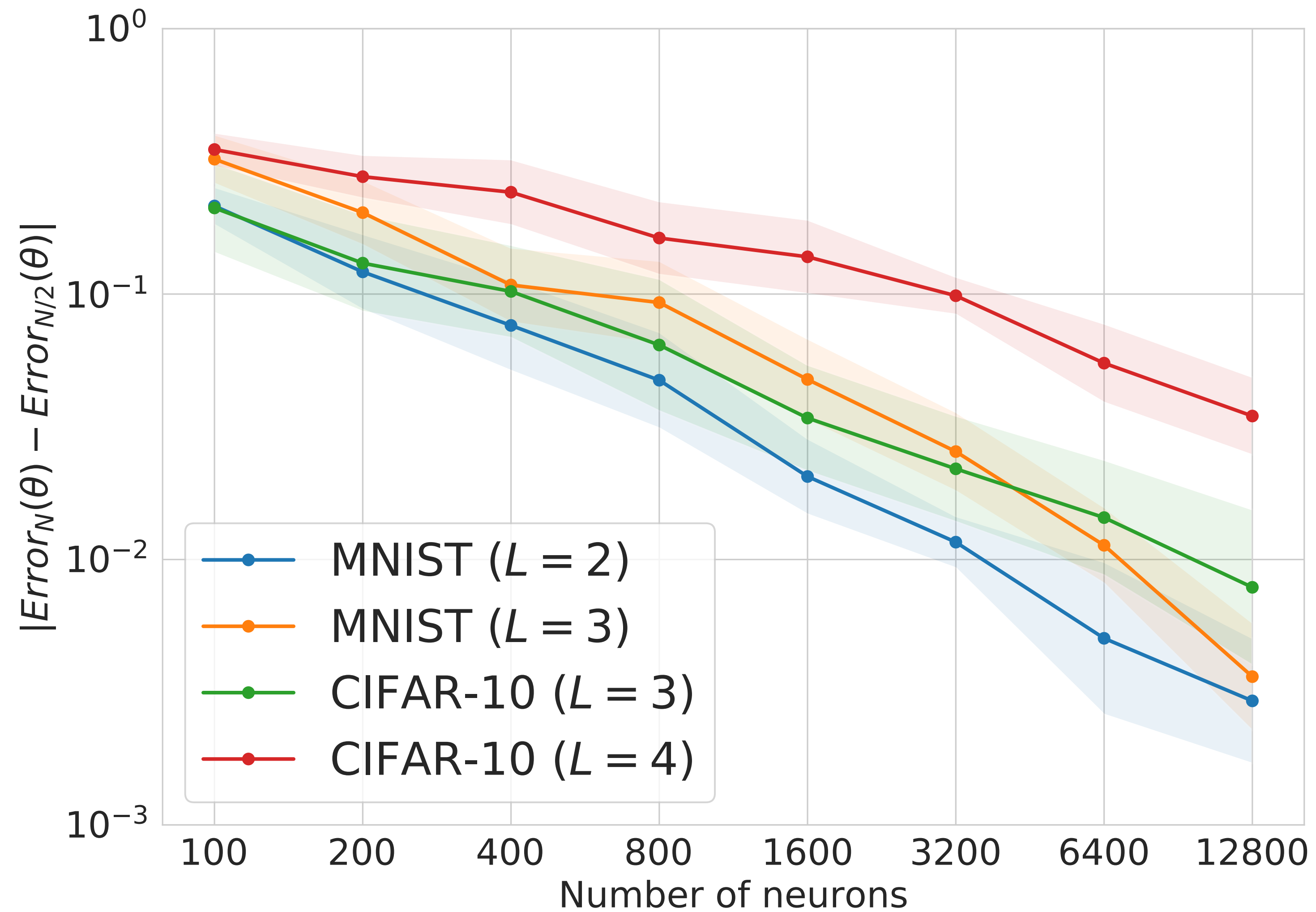


Numerical Results

- CIFAR-10 dataset
- Pretrained VGG-16 features
- # layers = 3
- Keep half of neurons

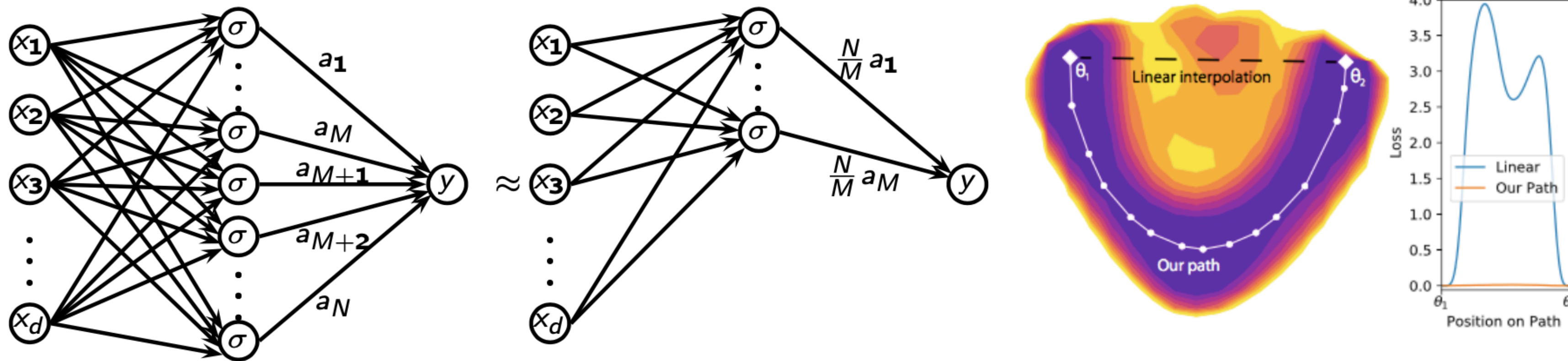


Numerical Results



Conclusion

Over-parameterization + SGD \Rightarrow dropout stability & connectivity



Thank You for Your Attention

Over-parameterization + SGD \Rightarrow dropout stability & connectivity

