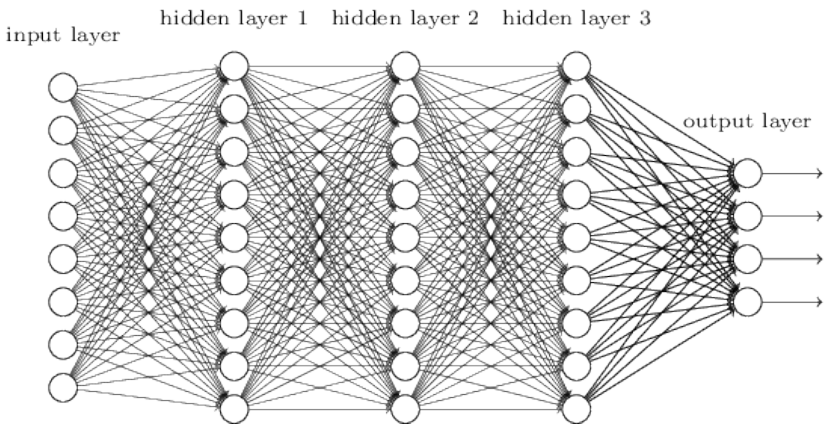
Understanding Gradient Descent for Over-parameterized Deep Neural Networks

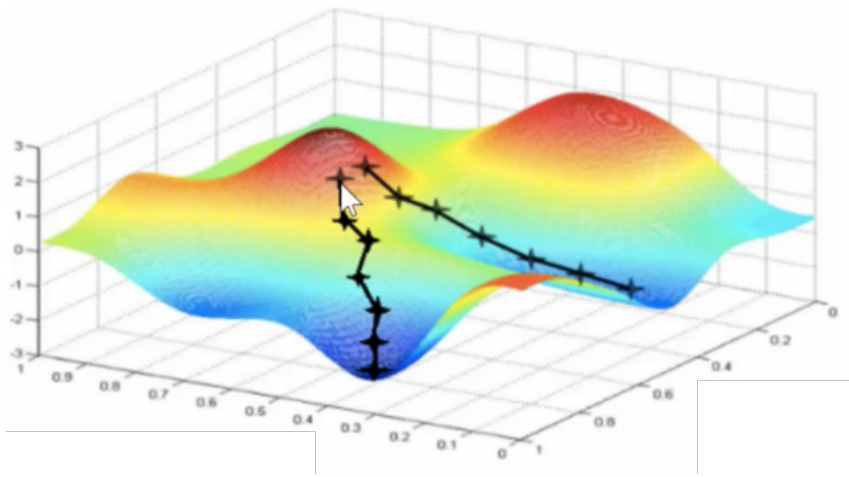
Marco Mondelli

Institute of Science and Technology Austria (IST Austria)

Youth in High-dimensions, 2 July 2020

Training a neural network is difficult (NP-hardness, local/disconnected minima...), but it works remarkably well!

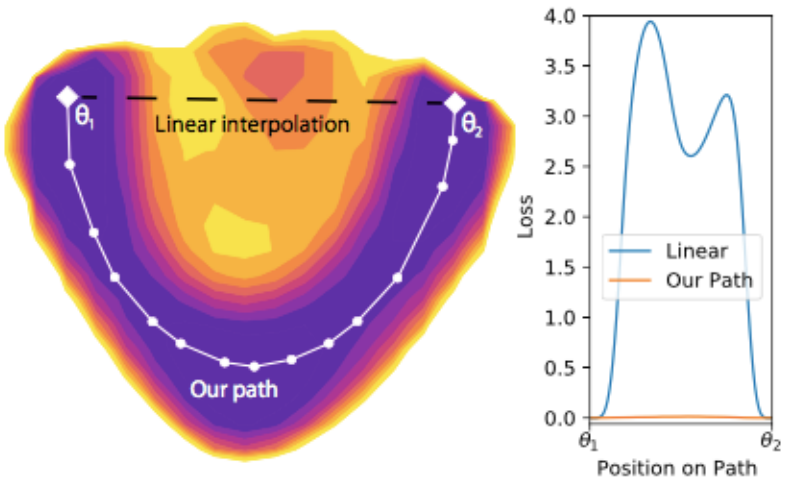




Over-parameterization

(Stochastic) gradient descent

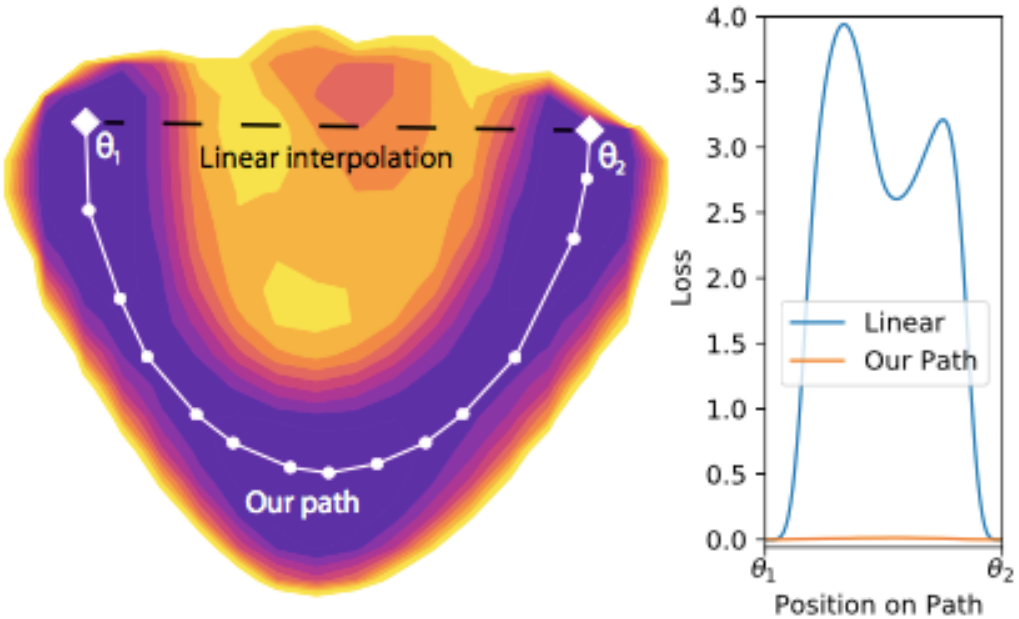
Landscape Connectivity and Dropout Stability



[SM20] A. Shevchenko and M. Mondelli, "Landscape Connectivity and Dropout Stability of SGD Solutions for Over-parameterized Neural Networks", *ICML*, 2020.

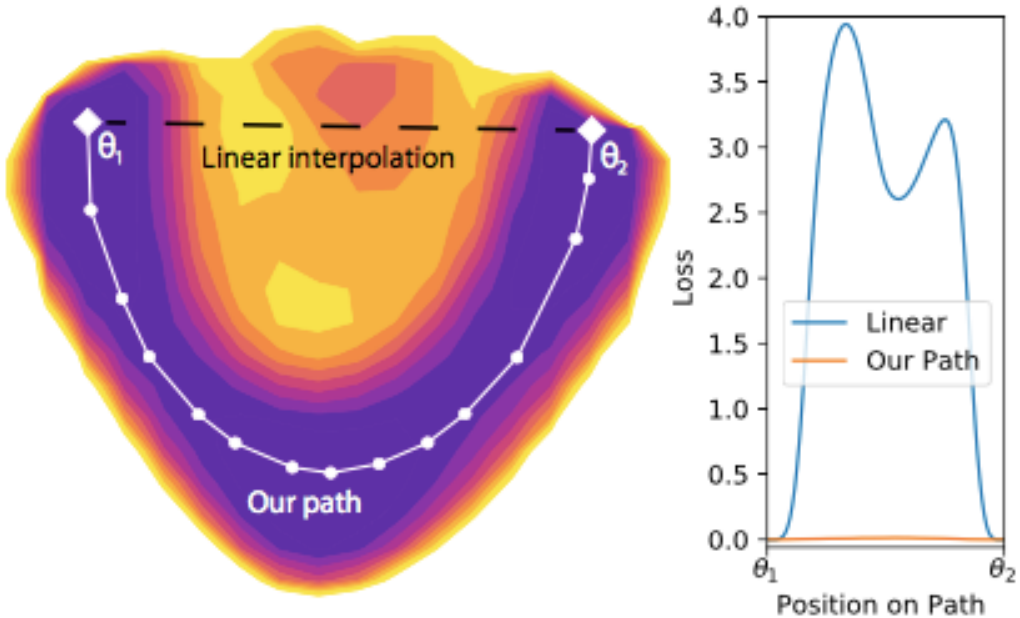
Landscape Connectivity

- SGD minima connected via piecewise linear path with constant loss [Garipov et al., 2018; Draxler et al., 2018]



Landscape Connectivity

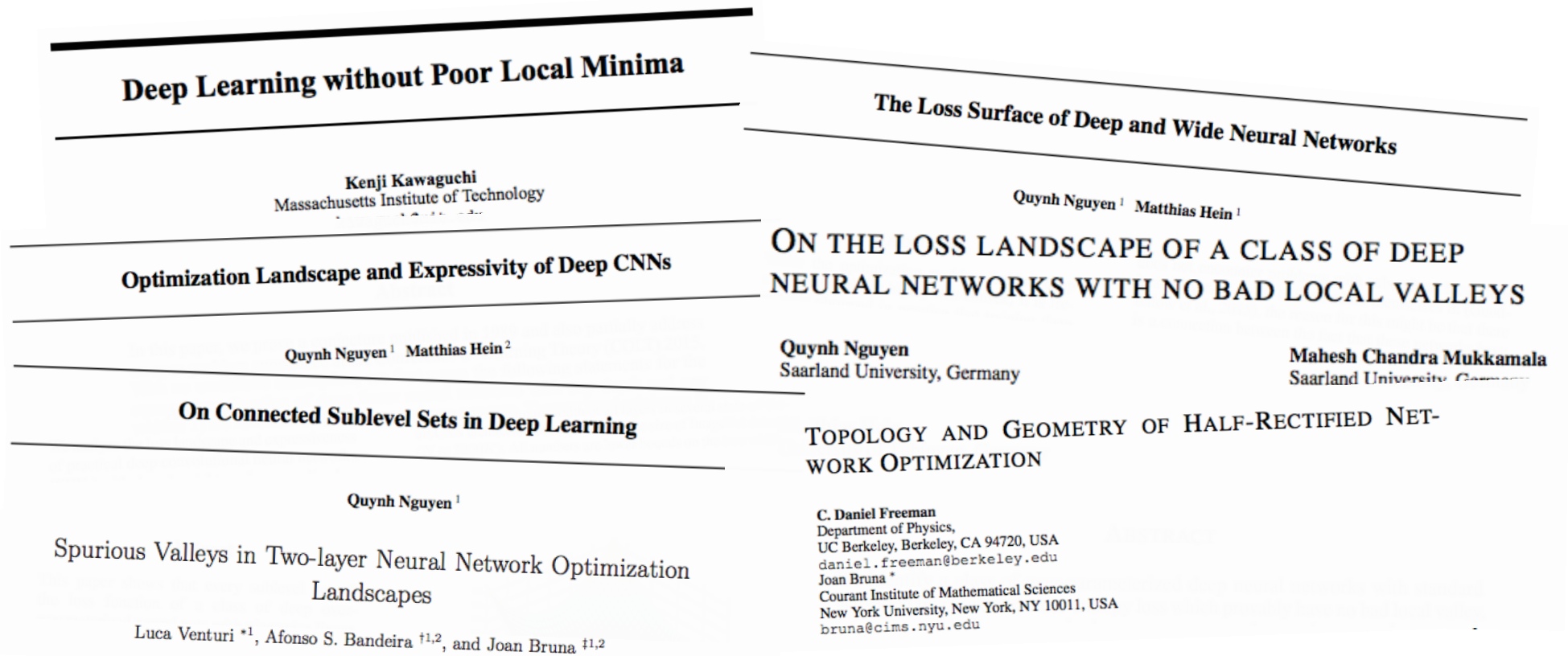
- SGD minima connected via piecewise linear path with constant loss [Garipov et al., 2018; Draxler et al., 2018]



- Mode connectivity proved assuming properties of well-trained networks (dropout/noise stability) [Kuditipudi et al., 2019]

Landscape of Neural Networks

- Local minima are globally optimal for deep linear networks and networks with more neurons than training samples
- Connected sublevel sets for one wide layer and pyramidal topology
- Energy gap scaling exponentially with input dimension

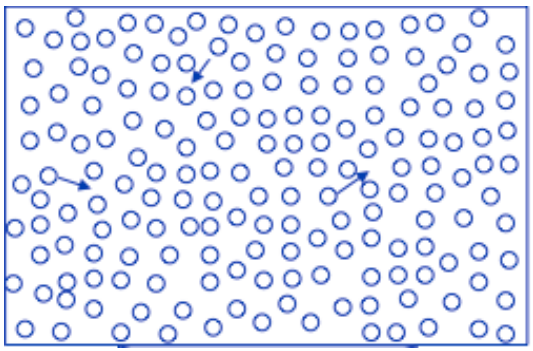


Landscape of Neural Networks

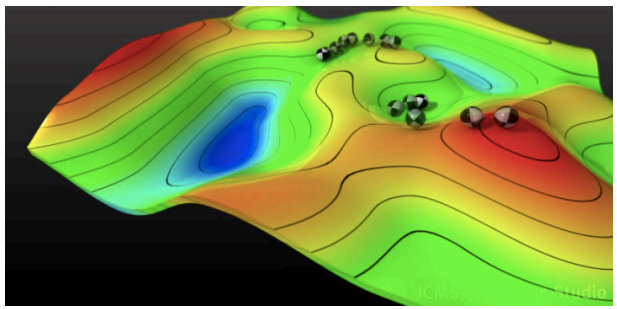
- Local minima are globally optimal for deep linear networks and networks with more neurons than training samples
- Connected sublevel sets for one wide layer and pyramidal topology
- Energy gap scaling exponentially with input dimension

Strong assumptions on the model and **poor scaling** of parameters ☹

Mean Field View of Neural Networks



$\longrightarrow$



Discrete dynamics of SGD

Continuous dynamics of gradient flow

- Two layers [Mei et al., 2018; Rotskoff et al., 2018; Chizat et al., 2018; Sirignano et al., 2018; . . .]
- Multiple layers [Nguyen, 2019; Sirignano et al., 2019; Araujo et al., 2019]

Warm-Up: Two-Layer Networks

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

Warm-Up: Two-Layer Networks

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

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

Warm-Up: Two-Layer Networks

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

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

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

Warm-Up: Two-Layer Networks

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

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

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

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

Dropout Stability

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

Dropout Stability

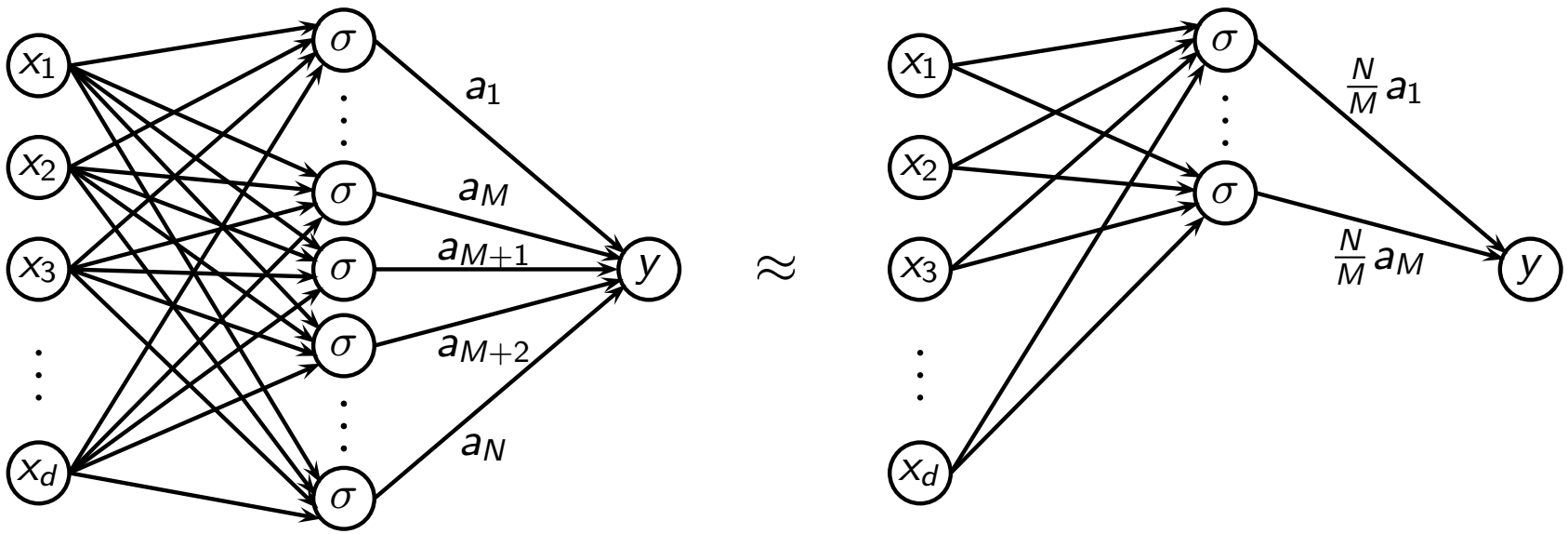
$$L_M(\boldsymbol{\theta}) = \mathbb{E}\left\{ \left(y - \frac{1}{M} \sum_{i=1}^{M} a_i \sigma(\boldsymbol{x}, \boldsymbol{w}_i) \right)^2 \right\}$$

$\boldsymbol{\theta}$ is ε_{D}-**dropout stable** if $|L_N(\boldsymbol{\theta}) - L_M(\boldsymbol{\theta})| \leq \varepsilon_{\mathrm{D}}$

Dropout Stability

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

$\boldsymbol{\theta}$ is ε_{D}-**dropout stable** if $|L_N(\boldsymbol{\theta}) - L_M(\boldsymbol{\theta})| \leq \varepsilon_{\mathrm{D}}$



Dropout Stability and Connectivity

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

$\boldsymbol{\theta}$ is ε_{D}-**dropout stable** if $|L_N(\boldsymbol{\theta}) - L_M(\boldsymbol{\theta})| \leq \varepsilon_{\mathrm{D}}$

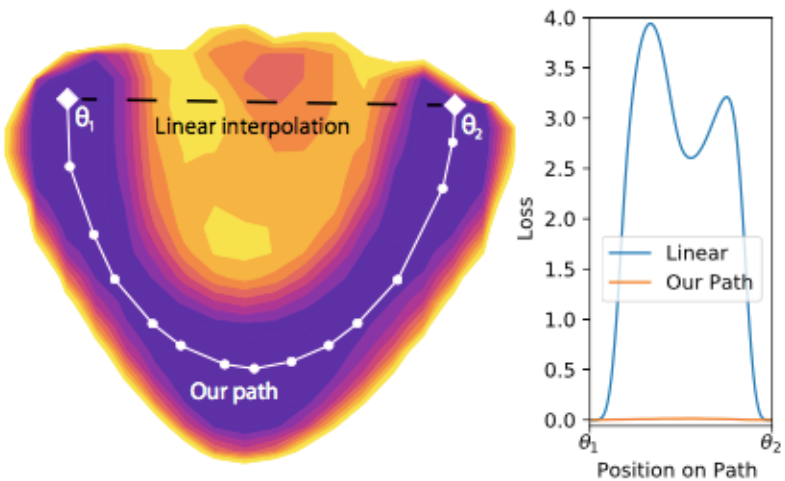
$\boldsymbol{\theta}$ and $\boldsymbol{\theta}'$ are ε_{C}-**connected** if there exists a continuous path connecting them where the loss does not increase more than ε_{C}

Dropout Stability and Connectivity

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

$\boldsymbol{\theta}$ is ε_{D}-**dropout stable** if $|L_N(\boldsymbol{\theta}) - L_M(\boldsymbol{\theta})| \leq \varepsilon_{\mathrm{D}}$

$\boldsymbol{\theta}$ 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}



Dropout Stability, Provably

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

Theorem [SM20]

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

$$\varepsilon_{\mathrm{D}} = K e^{KT^3} \left(\frac{\sqrt{\log M} + z}{\sqrt{M}} + \sqrt{\alpha} \left(\sqrt{D + \log N} + z \right) \right).$$

Dropout Stability, Provably

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

Theorem [SM20]

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

$$\varepsilon_{\mathrm{D}} = K e^{KT^3} \left(\frac{\sqrt{\log M} + z}{\sqrt{M}} + \sqrt{\alpha}\left(\sqrt{D + \log N} + z\right) \right).$$

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

Connectivity, Provably

Theorem [SM20]

Let $\boldsymbol{\theta}^k$ be obtained after k SGD iterations using $\{(\boldsymbol{x}_j, y_j)\}_{j=0}^{k} \sim \mathbb{P}$, and $(\boldsymbol{\theta}')^{k'}$ after k' SGD iterations using $\{(\boldsymbol{x}'_j, y'_j)\}_{j=0}^{k'} \sim \mathbb{P}$.

Connectivity, Provably

Theorem [SM20]

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

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

Connectivity, Provably

Theorem [SM20]

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

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

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

Connectivity, Provably

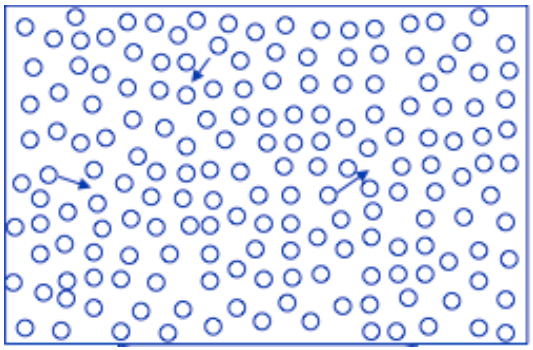
Theorem [SM20]

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

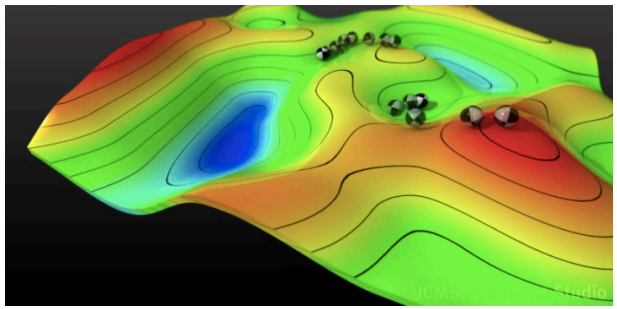
$$\varepsilon_{\mathrm{C}} = K e^{K \max(T,T')^3} \left(\frac{\sqrt{\log N} + z}{\sqrt{N}} + \sqrt{\alpha}\left(\sqrt{D + \log N} + z\right) \right).$$

- 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



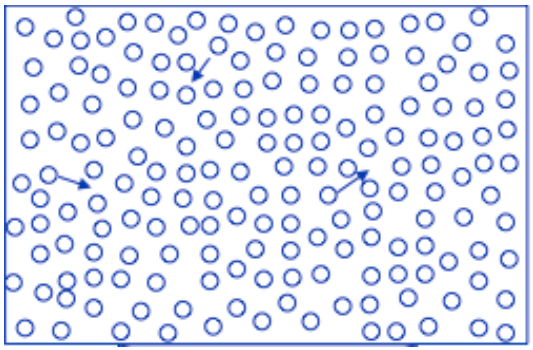
$\longrightarrow$



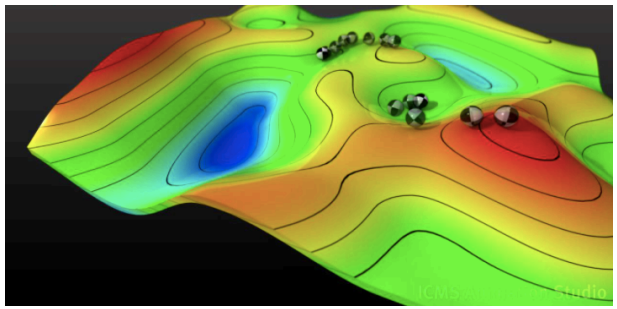
Discrete dynamics of SGD

Continuous dynamics of gradient flow

Proof Idea



$\longrightarrow$

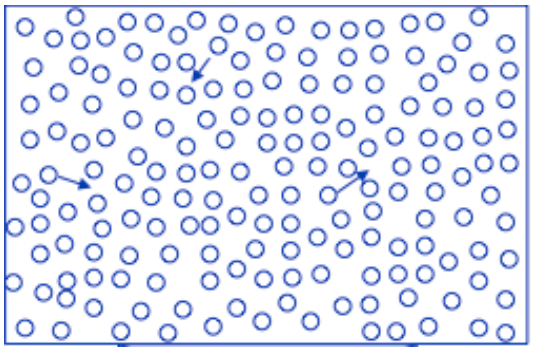


Discrete dynamics of SGD

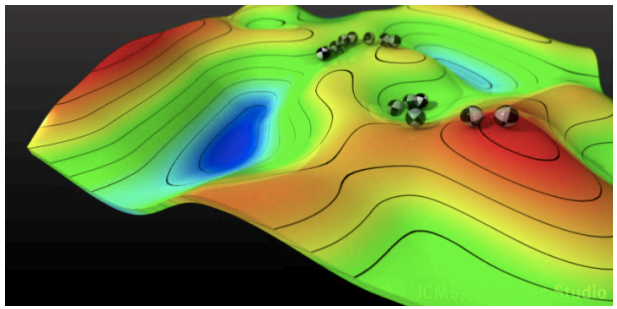
Continuous dynamics of gradient flow

- θ^k close to N i.i.d. particles that evolve with gradient flow

Proof Idea



$\longrightarrow$

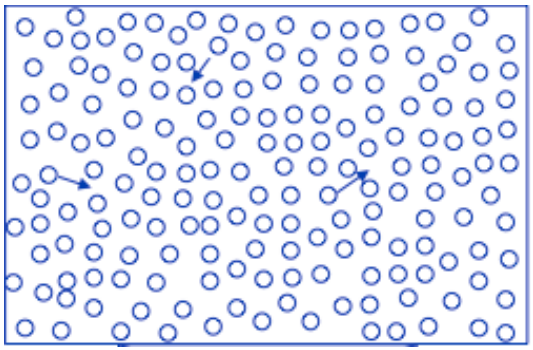


Discrete dynamics of SGD

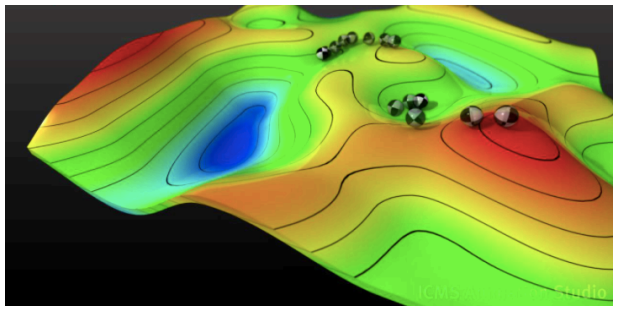
Continuous dynamics of gradient flow

- θ^k close to N i.i.d. particles that evolve with gradient flow
- $L_N(\theta^k)$ and $L_M(\theta^k)$ concentrate to the same limit

Proof Idea



$\longrightarrow$



Discrete dynamics of SGD

Continuous dynamics of gradient flow

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

Multilayer Networks

- $N = \#$ 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 [SM20]

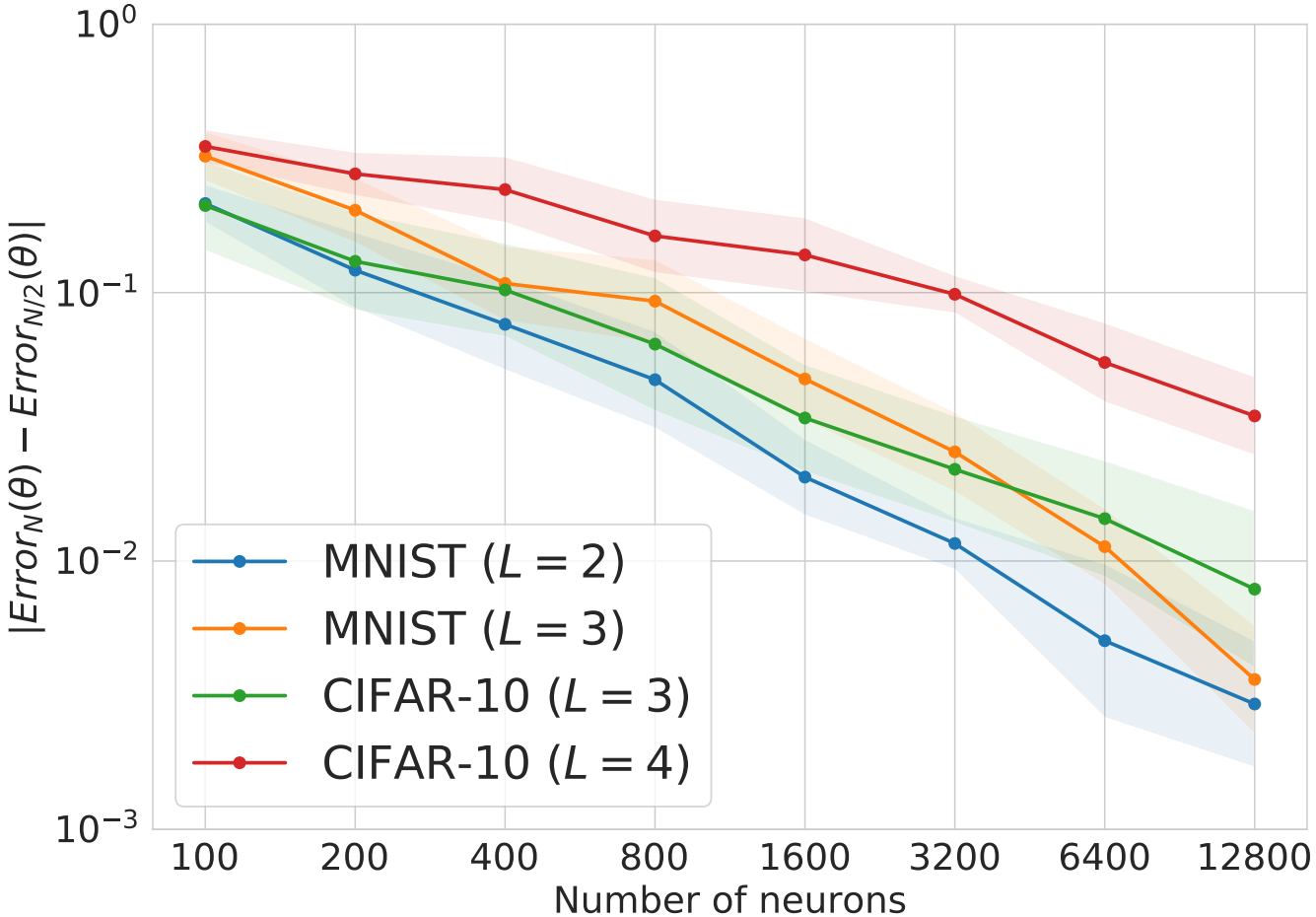
Let $\boldsymbol{\theta}^k$ be obtained after k SGD iterations, with $k = T/\alpha$. Then, w. p. $1 - e^{-z^2}$, $\boldsymbol{\theta}^k$ is ε_{D}-dropout stable with

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

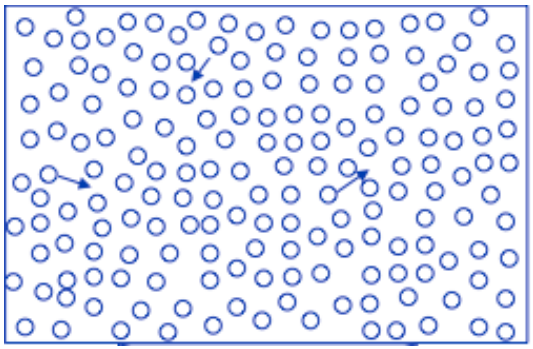
Let $(\boldsymbol{\theta}')^{k'}$ be obtained after k' SGD iterations, with $k' = T'/\alpha$. Then, w. p. $1 - e^{-z^2}$, $\boldsymbol{\theta}^k$ and $(\boldsymbol{\theta}')^{k'}$ are ε_{C}-connected with

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

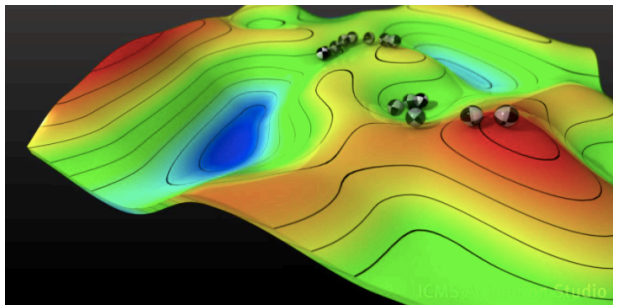
Numerical Results



Global Convergence Without Rates



$\longrightarrow$



Discrete dynamics of SGD

Continuous dynamics of gradient flow

A mean field view of the landscape of two-layer neural networks

Song Mei[a], Andrea Montanari[b,c,1], and Phan-Minh Nguyen[b]

[a]Institute for Computational and Mathematical Engineering, Stanford University, Stanford, CA 94305; [b]Department of Electrical Engineering, Stanford University, Stanford, CA 94305; and [c]Department of Statistics, Stanford University, Stanford, CA 94305

Edited by Peter J. Bickel, University of California, Berkeley, CA, and approved June 21, 2018 (received for review April 16, 2018)

Multilayer neural networks are among the most powerful models in machine learning, yet the fundamental reasons for this success defy mathematical understanding. Learning a neural network requires optimizing a nonconvex high-dimensional objective (risk function), a problem that is usually attacked using stochastic gra-

present case, this amounts to the iteration

$$\theta_i^{k+1} = \theta_i^k + 2\varepsilon_k \left(y_k - \hat{y}(\boldsymbol{x}_k; \boldsymbol{\theta}^k)\right) \nabla_{\theta_i} \sigma_*(\boldsymbol{x}_k; \theta_i^k). \quad [3]$$

On the Global Convergence of Gradient Descent for Over-parameterized Models using Optimal Transport

Lénaïc Chizat
INRIA, ENS, PSL Research University
Paris, France
lenaic.chizat@inria.fr

Francis Bach
INRIA, ENS, PSL Research University
Paris, France
francis.bach@inria.fr

Abstract

Many tasks in machine learning and signal processing can be solved by minimizing a convex function of a measure. This includes sparse spikes deconvolution or training a neural network with a single hidden layer. For these problems, we study a simple minimization method: the unknown measure is discretized into a mixture of particles and a continuous-time gradient descent is performed on their weights

Data Model

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

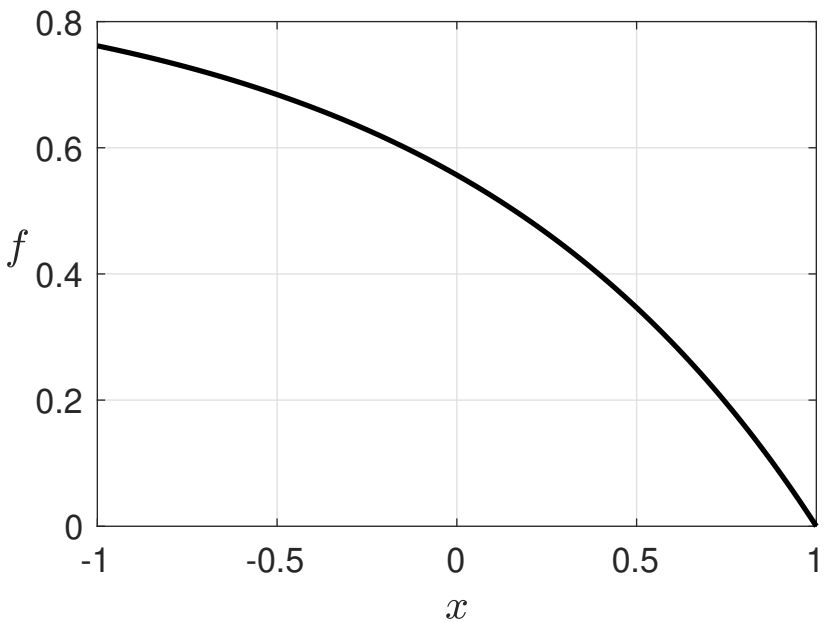
$$\boldsymbol{x}_i \sim \text{Unif}(\Omega) \text{ and } y_i = f(\boldsymbol{x}_i)$$

Data Model

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

$$\boldsymbol{x}_i \sim \text{Unif}(\Omega) \text{ and } y_i = f(\boldsymbol{x}_i)$$

- Ω bounded and convex
- f positive, smooth and γ-strongly concave



Problem Statement

Goal: Minimize loss $L_N(\boldsymbol{w}) = \mathbb{E}\Big\{\Big(y - \frac{1}{N}\sum_{i=1}^{N}\sigma(\boldsymbol{x};\boldsymbol{w}_i)\Big)^2\Big\}$

Problem Statement

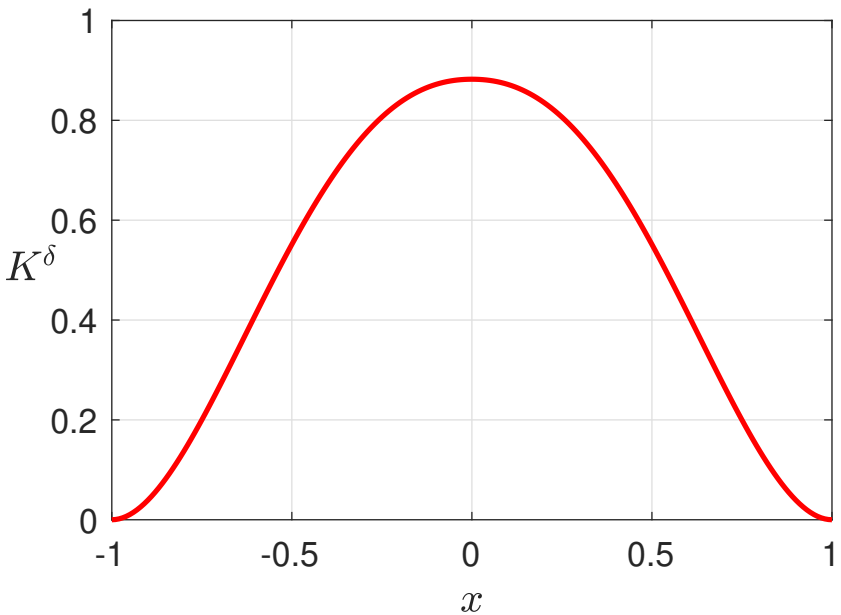
Goal: Minimize loss $L_N(\boldsymbol{w}) = \mathbb{E}\left\{\left(y - \frac{1}{N}\sum_{i=1}^{N}\sigma(\boldsymbol{x};\boldsymbol{w}_i)\right)^2\right\}$

- σ 'bump'-like

$\sigma(\boldsymbol{x};\boldsymbol{w}_i) = K^\delta(\boldsymbol{x} - \boldsymbol{w}_i)$

- $\boldsymbol{w}_i$ = center of bump
- δ = width of the bump

$K^\delta(\boldsymbol{x}) = \delta^{-d}K(\boldsymbol{x}/\delta)$



Exponential and Dimension-free Convergence Rate

Theorem [JMM20]

Let f be γ-strongly concave, σ a bump of width δ, and $\boldsymbol{w}^k$ obtained after k SGD iterations with step size α. Then, with high probability,

$$\underbrace{L_N(\boldsymbol{w}^k)}_{\text{loss at step } k} \leq L_N(\boldsymbol{w}^0) \underbrace{e^{-2\gamma k\alpha}}_{\text{exp. rate}} + \underbrace{\Delta(N,\alpha,\delta)}_{\text{distance to opt.}},$$

where $\Delta(N,\alpha,\delta) \to 0$ as $N \to \infty$, $\alpha \to 0$, and $\delta \to 0$.

- As $\delta \to 0$, gradient flow optimizes a **displacement convex** loss.

[JMM20] A. Javanmard, M. Mondelli, and A. Montanari, "Analysis of a Two-Layer Neural Network via Displacement Convexity", *Annals of Statistics*, 2020+.

Thank You for Your Attention