Collapse of Deep and Narrow ReLU Neural Nets

Lu Lu, Yeonjong Shin, Yanhui Su, George Karniadakis

Division of Applied Mathematics, Brown University

Scientific Machine Learning, ICERM
January 28, 2019

Overview

Introduction

- Shallow NNs (single hidden layer)
  - universal approximation theorem

Introduction

- Shallow NNs (single hidden layer)
  - universal approximation theorem
- Deep (& narrow) NNs
  - Better than shallow NNs (of comparable size)
  - $\frac{\text{size}_{\text{deep}}}{\text{size}_{\text{shallow}}} \approx \epsilon^d$ [Mhaskar & Poggio, 2016]

Introduction

- Shallow NNs (single hidden layer)
  - universal approximation theorem
- Deep (& narrow) NNs
  - Better than shallow NNs (of comparable size)
  - $\frac{\text{size}_{\text{deep}}}{\text{size}_{\text{shallow}}} \approx \epsilon^d$ [Mhaskar & Poggio, 2016]

$\Rightarrow$ Deep & narrow

Introduction

- Shallow NNs (single hidden layer)
  - universal approximation theorem
- Deep (& narrow) NNs
  - Better than shallow NNs (of comparable size)
  - $\frac{\text{size}_{\text{deep}}}{\text{size}_{\text{shallow}}} \approx \epsilon^d$ [Mhaskar & Poggio, 2016]

$\Rightarrow$ Deep & narrow

- ReLU $:= \max(x, 0)$

Introduction

- Shallow NNs (single hidden layer)
  - universal approximation theorem
- Deep (& narrow) NNs
  - Better than shallow NNs (of comparable size)
  - $\frac{\text{size}_{\text{deep}}}{\text{size}_{\text{shallow}}} \approx \epsilon^d$ [Mhaskar & Poggio, 2016]

$\Rightarrow$ Deep & narrow

- ReLU $:= \max(x, 0)$
  - Width limit?
    For continuous functions $[0,1]^{d_{in}} \to \mathbb{R}^{d_{out}}$ [Hanin & Sellke, 2017]:

$$d_{in} + 1 \leq \text{ minimal width } \leq d_{in} + d_{out}$$

Introduction

- Shallow NNs (single hidden layer)
  - universal approximation theorem
- Deep (& narrow) NNs
  - Better than shallow NNs (of comparable size)
  - $\frac{\text{size}_{\text{deep}}}{\text{size}_{\text{shallow}}} \approx \epsilon^d$ [Mhaskar & Poggio, 2016]

$\Rightarrow$ Deep & narrow

- ReLU $:= \max(x, 0)$
  - Width limit?
    For continuous functions $[0, 1]^{d_{in}} \to \mathbb{R}^{d_{out}}$ [Hanin & Sellke, 2017]:

$$d_{in} + 1 \leq \text{ minimal width } \leq d_{in} + d_{out}$$

  - Depth limit?

Introduction

Training of NNs

- NP-hard [Sima, 2002]
- Local minima [Fukumizu & Amari, 2002]
- Bad saddle points [Kawaguchi, 2016]

Introduction

Training of NNs

- NP-hard [Sima, 2002]
- Local minima [Fukumizu & Amari, 2002]
- Bad saddle points [Kawaguchi, 2016]

ReLU

- Dying ReLU neuron: stuck in the negative side

Introduction

Training of NNs

- NP-hard [Sima, 2002]
- Local minima [Fukumizu & Amari, 2002]
- Bad saddle points [Kawaguchi, 2016]

ReLU

- Dying ReLU neuron: stuck in the negative side

Deep ReLU nets?

Dying ReLU network

NN is a constant function after initialization

Introduction

Training of NNs

- NP-hard [Sima, 2002]
- Local minima [Fukumizu & Amari, 2002]
- Bad saddle points [Kawaguchi, 2016]

ReLU

- Dying ReLU neuron: stuck in the negative side

Deep ReLU nets?

Dying ReLU network

NN is a constant function after initialization

Collapse

NN converges to the "mean" state of the target function during training

Overview

1D Examples

$f(x) = |x|$

- $|x| = \text{ReLU}(x) + \text{ReLU}(-x) = \begin{bmatrix} 1 & 1 \end{bmatrix} \text{ReLU}(\begin{bmatrix} 1 \\ -1 \end{bmatrix} x)$
- 2-layer with width 2

Train a 10-layer ReLU NN with width 2 (MSE loss, whatever optimizer)

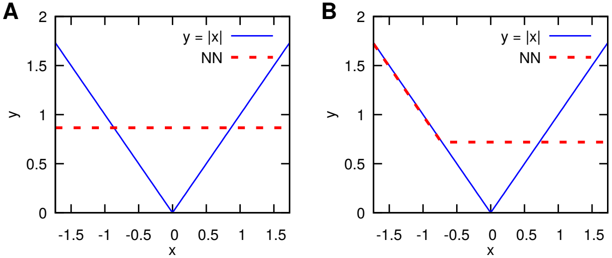
1D Examples

$f(x) = |x|$

- $|x| = \text{ReLU}(x) + \text{ReLU}(-x) = \begin{bmatrix} 1 & 1 \end{bmatrix} \text{ReLU}(\begin{bmatrix} 1 \\ -1 \end{bmatrix} x)$
- 2-layer with width 2

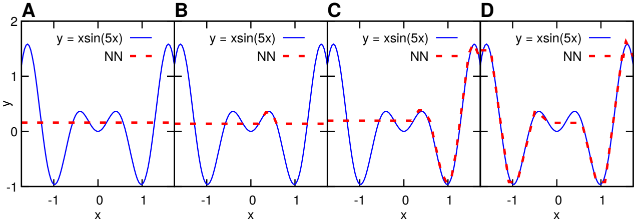
Train a 10-layer ReLU NN with width 2 (MSE loss, whatever optimizer)

- Collapse to the mean value (A): ∼93%
- Collapse partially (B)

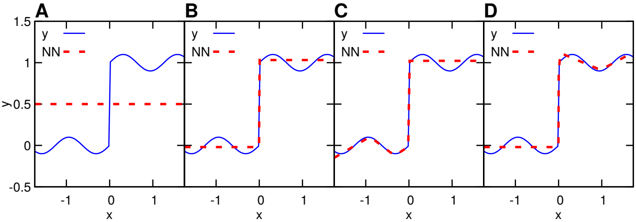


1D Examples

$f(x) = x\sin(5x)$

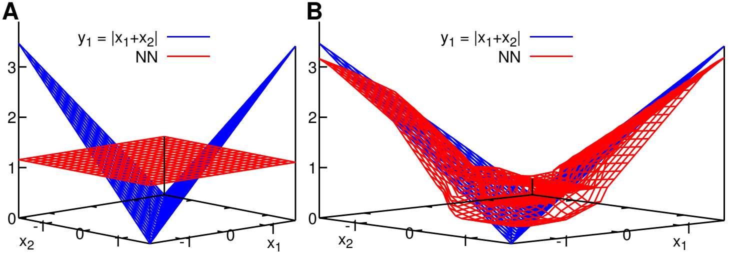


$f(x) = 1_{\{x>0\}} + 0.2\sin(5x)$



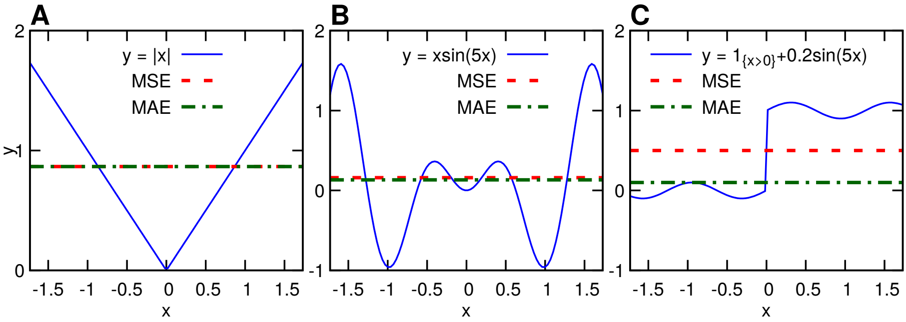
2D Examples

$$f(\mathbf{x}) = \begin{bmatrix} |\mathbf{x}_1 + \mathbf{x}_2| \\ |\mathbf{x}_1 - \mathbf{x}_2| \end{bmatrix} = \begin{bmatrix} 1 & 1 & & \\ & & 1 & 1 \end{bmatrix} \text{ReLU}\left(\begin{bmatrix} 1 & 1 \\ -1 & -1 \\ 1 & -1 \\ -1 & 1 \end{bmatrix} \mathbf{x} \right)$$



Loss

- Mean squared error (MSE) $\Rightarrow$ mean
- Mean absolute error (MAE) $\Rightarrow$ median



Overview

Setup

- Feed-forward ReLU neural network $\mathcal{N}^L : \mathbb{R}^{d_{in}} \to \mathbb{R}^{d_{out}}$
- L layers
- In the layer ℓ
  - N_ℓ neurons ($N_0 = d_{in}$, $N_L = d_{out}$)
  - Weight $\mathbf{W}^\ell$: $N_\ell \times N_{\ell-1}$ matrix
  - Bias $\mathbf{b}^\ell \in \mathbb{R}^{N_\ell}$
- Input: $\mathbf{x} \in \mathbb{R}^{d_{in}}$
- Neural activity in the layer ℓ: $\mathcal{N}^\ell(\mathbf{x}) \in \mathbb{R}^{N_\ell}$

$$\mathcal{N}^\ell(\mathbf{x}) = \boldsymbol{W}^\ell \phi(\mathcal{N}^{\ell-1}(\mathbf{x})) + \boldsymbol{b}^\ell \in \mathbb{R}^{N_\ell}, \qquad \text{for} \quad 2 \le \ell \le L$$

$$\mathcal{N}^1(\mathbf{x}) = \boldsymbol{W}^1 \mathbf{x} + \boldsymbol{b}^1$$

Setup

- Training data

$$\mathcal{T} = \{\mathbf{x}_i, f(\mathbf{x}_i)\}_{1 \le i \le M} \subset \mathcal{D} \equiv B_r(0) = \{x \in \mathbb{R}^{d_{\text{in}}} | \|x\|_2 \le r\}$$

- Loss function

$$\mathcal{L}(\theta, \mathcal{T}) = \sum_{i=1}^{M} \ell(\mathcal{N}^L(\mathbf{x}_i; \theta), f(\mathbf{x}_i)),$$

where $\boldsymbol{\theta} = \{\boldsymbol{W}^\ell, \boldsymbol{b}^\ell\}_{1 \le \ell \le L}$

$\mathcal{N}^L$ will eventually Die in probability as $L \to \infty$

Theorem 1

Let $\mathcal{N}^L(\mathbf{x})$ be a ReLU NN with L layers, each having $N_1, \cdots, N_L$ neurons. Suppose

1. Weights are independently initialized from a symmetric distribution around 0,
2. Biases are either from a symmetric distribution or set to be zero.

Then

$$P(\mathcal{N}^L(\mathbf{x}) \text{ dies}) \leq 1 - \prod_{\ell=1}^{L-1} (1 - (1/2)^{N_\ell}).$$

Furthermore, assuming $N_\ell = N$ for all ℓ,

$$\lim_{L\to\infty} P(\mathcal{N}^L(\mathbf{x}) \text{ dies}) = 1, \qquad \lim_{N\to\infty} P(\mathcal{N}^L(\mathbf{x}) \text{ dies}) = 0.$$

Proof

Lemma 1

Let $\mathcal{N}^L(\mathbf{x})$ be a ReLU NN of L-layers. Suppose weights are independently from distributions satisfying $P(\boldsymbol{W}_j^\ell \boldsymbol{z} = \mathbf{0}) = 0$ for any nonzero $\boldsymbol{z} \in \mathbb{R}^{N_{\ell-1}}$ and any j-th row of $\boldsymbol{W}^\ell$. Then

$$P(\mathcal{N}^\ell(\mathbf{x}) \text{ dies}) = P(\exists\, \ell \in \{1, \ldots, L-1\} \text{ s.t. } \phi(\mathcal{N}^\ell(\mathbf{x})) = \mathbf{0}\ \forall \mathbf{x} \in \mathcal{D}).$$

Proof

Lemma 1

Let $\mathcal{N}^L(\mathbf{x})$ be a ReLU NN of L-layers. Suppose weights are independently from distributions satisfying $P(\boldsymbol{W}_j^\ell \boldsymbol{z} = \mathbf{0}) = 0$ for any nonzero $\boldsymbol{z} \in \mathbb{R}^{N_{\ell-1}}$ and any j-th row of $\boldsymbol{W}^\ell$. Then

$$P(\mathcal{N}^\ell(\mathbf{x}) \text{ dies}) = P(\exists\, \ell \in \{1, \ldots, L-1\} \text{ s.t. } \phi(\mathcal{N}^\ell(\mathbf{x})) = \mathbf{0}\ \forall \mathbf{x} \in \mathcal{D}).$$

- For a given $\mathbf{x}$,

$$P\left(\boldsymbol{W}_s^j \phi(\mathcal{N}^{j-1}(\mathbf{x})) + \boldsymbol{b}_s^j < 0 | \tilde{A}_{j-1,\mathbf{x}}^c\right) = \frac{1}{2},$$

where $\tilde{A}_{\ell,\mathbf{x}}^c = \{\forall 1 \leq j < \ell,\ \phi(\mathcal{N}^j(\mathbf{x})) \neq \mathbf{0}\}$

Dead Networks would Collapse

Theorem 2

Suppose the ReLU NN dies. Then for any loss $\mathcal{L}$, the network is optimized to a constant function by any gradient based method.

Dead Networks would Collapse

Theorem 2

Suppose the ReLU NN dies. Then for any loss $\mathcal{L}$, the network is optimized to a constant function by any gradient based method.

Proof

- Lemma 1 $\Rightarrow \exists\, \ell \in \{1, \ldots, L-1\}$ s.t. $\phi(\mathcal{N}^{\ell}(\mathbf{x})) = \mathbf{0}\ \forall \mathbf{x} \in \mathcal{D}$
- Gradients of $\mathcal{L}$ wrt the weights/biases in the $1, \ldots, l$-th layers vanish

Dead Networks would Collapse

Theorem 2

Suppose the ReLU NN dies. Then for any loss $\mathcal{L}$, the network is optimized to a constant function by any gradient based method.

Proof

- Lemma 1 $\Rightarrow \exists\, \ell \in \{1, \ldots, L-1\}$ s.t. $\phi(\mathcal{N}^{\ell}(\mathbf{x})) = \mathbf{0}\ \forall \mathbf{x} \in \mathcal{D}$
- Gradients of $\mathcal{L}$ wrt the weights/biases in the $1, \ldots, l$-th layers vanish

Assuming training data are iid from $P_{\mathcal{D}}$, the optimized network is

$$\mathcal{N}^{L}(\mathbf{x}; \theta^*) = \underset{\boldsymbol{c} \in \mathbb{R}^{N_L}}{\operatorname{argmin}}\, \mathbb{E}_{\mathbf{x} \sim P_{\mathcal{D}}} \left[\ell(\boldsymbol{c}, f(\mathbf{x})) \right]$$

- MSE/$L^2 \Rightarrow \mathbb{E}[f(\mathbf{x})]$
- MAE/$L^1 \Rightarrow$ median of $f(\mathbf{x})$

Probability of Dying when $d_{in} = 1$

Theorem 3

Let $\mathcal{N}^L(\mathbf{x})$ be a bias-free ReLU NN with $L \geq 2$ layers, each having N neurons at $d_{\text{in}} = 1$. Suppose weights are independently initialized from continuous symmetric distributions around 0. Then

$$1 - \prod_{\ell=1}^{L-1}(1 - (1/2)^N) \geq P(\mathcal{N}^L(x) \text{ dies})$$

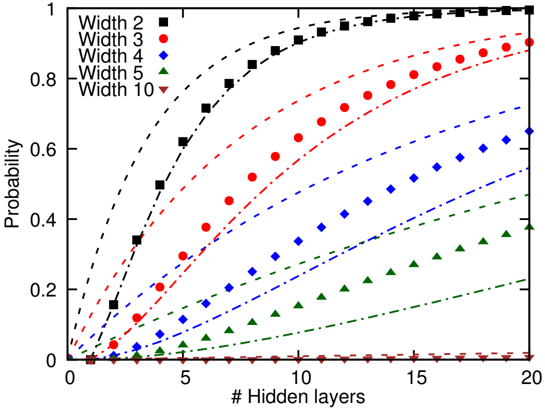
$$\geq 1 - (\mathcal{P}_{22})^{L-2} - \frac{(1 - 2^{-N+1})(1 - 2^{-N})}{1 + (N-1)2^{-N}}((\mathcal{P}_{22})^{L-2} - (\mathcal{P}_{33})^{L-2})$$

where $\mathcal{P}_{22} = 1 - \frac{1}{2^N}$ and $\mathcal{P}_{33} = 1 - \frac{1}{2^{N-1}} - \frac{N-1}{4^N}$.

Numerical Test

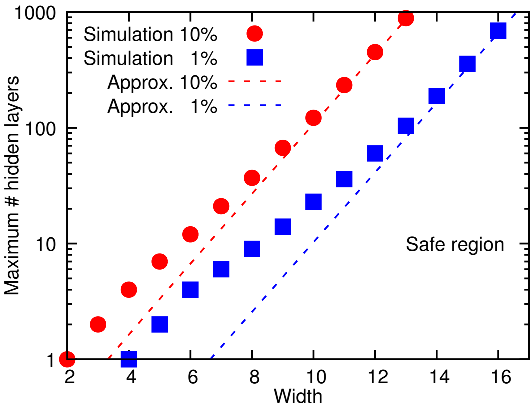
- A ReLU NN with $d_{in} = 1$
- Weights randomly initialized from symmetric distributions
- Biases are initialized to 0

More likely to die when it is deeper and narrower



Safe Operating Region for a ReLU NN

Keep the dying probability $< 10\%$ or 1%



Overview

References

Fukumizu, K., & Amari, S. I. (2000).
Local minima and plateaus in hierarchical structures of multilayer perceptrons.
Neural networks, 13(3), 317-327.

Sima, J. (2002).
Training a single sigmoidal neuron is hard.
Neural computation, 14(11), 2709-2728.

Kawaguchi, K. (2016).
Deep learning without poor local minima.
NIPS (pp. 586-594).

Mhaskar, H. N., & Poggio, T. (2016).
Deep vs. shallow networks: An approximation theory perspective.
Analysis and Applications, 14(06), 829-848.

Hanin, B., & Sellke, M. (2017).
Approximating Continuous Functions by ReLU Nets of Minimal Width.
arXiv preprint arXiv:1710.11278.

Lu, L., Su, Y., & Karniadakis, G. E. (2018).
Collapse of deep and narrow neural nets.
arXiv preprint arXiv:1808.04947.