Quantization of ReLU neural networks from an approximation theory point of view

Antoine Gonon

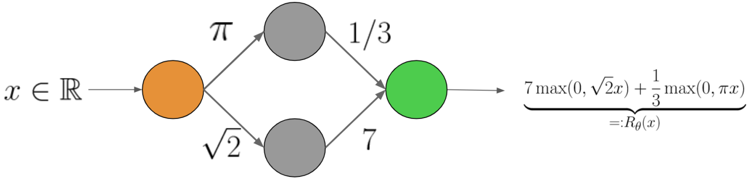
LIP, ENS Lyon

Joint work with: Nicolas Brisebarre, Rémi Gribonval, Elisa Riccietti

May 25, 2022

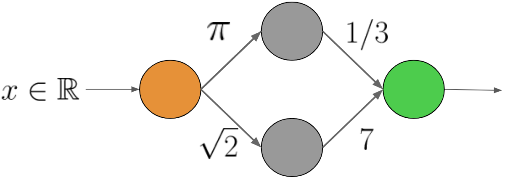
Context: quantization versus approximation

goal function f, accuracy $\varepsilon > 0$, real parameters θ of a ReLU network s.t. $\|f - R_\theta\| \leqslant \varepsilon$



Context: quantization versus approximation

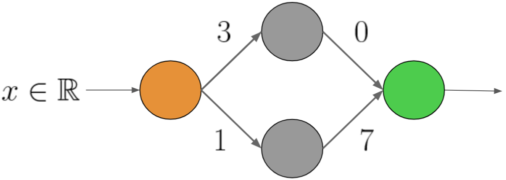
goal function f, accuracy $\varepsilon > 0$, real parameters θ of a ReLU network s.t. $\|f - R_\theta\| \leqslant \varepsilon$



$$\underbrace{7\max(0,\sqrt{2}x) + \frac{1}{3}\max(0,\pi x)}_{=:R_\theta(x)}$$

$$\|f - R_{Q(\theta)}\| \leqslant \|f - R_\theta\| + \underbrace{\|R_\theta - R_{Q(\theta)}\|}_{\textit{quantization error}}$$

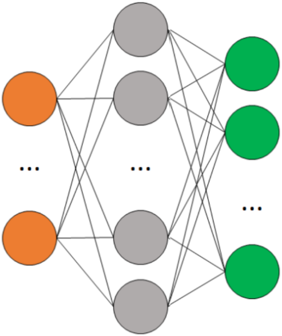
Tradeoff: number of bits vs. quantization error $\|R_\theta - R_{Q(\theta)}\|$*?*



$$\underbrace{7\max(0,x)}_{=:R_{Q(\theta)}(x)}$$

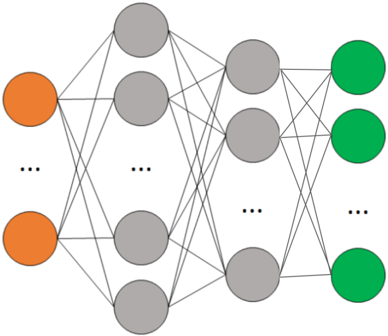
Context: approximation with increasing complexity

complexity $\nearrow \Longrightarrow$ **approximation** $\searrow$



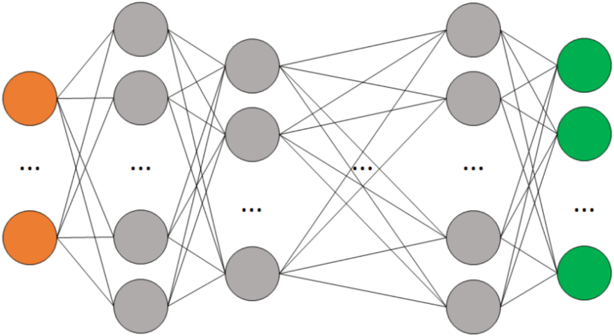
Context: approximation with increasing complexity

complexity $\nearrow \Longrightarrow$ **approximation error** $\searrow$



Context: approximation with increasing complexity

complexity $\nearrow \Longrightarrow$ **approximation error** $\searrow$



Problem

Σ_M set of networks increasingly complex with M

$$f \ \gamma\text{-smooth} \implies d(f, \Sigma_M) \lesssim M^{-\gamma}$$

Problem

Σ_M set of networks increasingly complex with M

$$f \ \gamma\text{-smooth} \implies d(f, \Sigma_M) \lesssim M^{-\gamma}$$

$$\boxed{\gamma^{*\text{approx}}(\mathcal{C}|\Sigma) := \text{largest } \gamma > 0 \text{ s.t. } \sup_{f \in \mathcal{C}} d(f, \Sigma_M) \underset{M \to \infty}{=} \mathcal{O}(M^{-\gamma})}$$

Problem

Σ_M set of networks increasingly complex with M

$$f \ \gamma\text{-smooth} \implies d(f, \Sigma_M) \lesssim M^{-\gamma}$$

$$\boxed{\gamma^{*\text{approx}}(\mathcal{C}|\Sigma) := \text{largest } \gamma > 0 \text{ s.t. } \sup_{f \in \mathcal{C}} d(f, \Sigma_M) \underset{M \to \infty}{=} \mathcal{O}(M^{-\gamma})}$$

Context: $\gamma^{*\text{approx}}(\mathcal{C}|\Sigma)$ known

- $\mathcal{C}$: bounded set of "smooth" functions (Sobolev, Besov)
- Σ: ReLU neural networks with *weights in* $\mathbb{R}$

Problem

Σ_M set of networks increasingly complex with M

$$f \ \gamma\text{-smooth} \Longrightarrow d(f, \Sigma_M) \lesssim M^{-\gamma}$$

$$\boxed{\gamma^{*\text{approx}}(\mathcal{C}|\Sigma) := \text{largest } \gamma > 0 \text{ s.t. } \sup_{f \in \mathcal{C}} d(f, \Sigma_M) \underset{M\to\infty}{=} \mathcal{O}(M^{-\gamma})}$$

Context: $\gamma^{*\text{approx}}(\mathcal{C}|\Sigma)$ known

- $\mathcal{C}$: bounded set of "smooth" functions (Sobolev, Besov)
- Σ: ReLU neural networks with *weights in* $\mathbb{R}$

Problem: *quantized* ReLU neural networks?

Tradeoff quantization/approximation?

Our approach: bound on the Lipschitz constant of the parameterization

Problem: Tradeoff number of bits/quantization error

$$\textit{Control of } \|R_\theta - R_{Q(\theta)}\|?$$

[1] Neural network approximation. R. DeVore et al. 2021.

Our approach: bound on the Lipschitz constant of the parameterization

Problem: Tradeoff number of bits/quantization error

Control of $\|R_\theta - R_{Q(\theta)}\|$*?*

Known result[1]**:** On every bounded set of parameters Θ, there exists $K_\Theta > 0$ s.t. for every $\theta, \theta' \in \Theta$:

$$\|R_\theta - R_{\theta'}\|_{L^p} \leqslant K_\Theta \|\theta - \theta'\|_\infty$$

Explicit bounds on K_Θ*?*

[1] Neural network approximation. R. DeVore et al. 2021.

Our approach: bound on the Lipschitz constant of the parameterization

Problem: Tradeoff number of bits/quantization error

$$\textit{Control of } \|R_\theta - R_{Q(\theta)}\| \textit{?}$$

Known result[1]**:** On every bounded set of parameters Θ, there exists $K_\Theta > 0$ s.t. for every $\theta, \theta' \in \Theta$:

$$\|R_\theta - R_{\theta'}\|_{L^p} \leqslant K_\Theta \|\theta - \theta'\|_\infty$$

$$\textit{Explicit bounds on } K_\Theta \textit{?}$$

Our contribution: explicit bounds in terms of the depth, width and bound on the weights of the network

[1] Neural network approximation. R. DeVore et al. 2021.

Contribution

Bounds on the Lipschitz parameterization of ReLU networks

Under mild assumptions, there exists $c > 0$ s.t.

$$\boxed{\frac{1}{c} LB^{L-1} \leqslant K_{\Theta_{L,W}(B)} \leqslant cWL \times LB^{L-1}}$$

- depth $L \in \mathbb{N}^*$
- width $W \in \mathbb{N}^*$
- bound $B \geqslant 1$ on $\theta = (W_1, \ldots, W_L, b_1, \ldots, b_L)$: $\|W_\ell\|_2, \|b_\ell\|_2 \leqslant B$

Consequence 1: naive uniform quantization

Naive uniform quantization

$\eta > 0$ such that $Q_\eta(x) = \lfloor x/\eta \rfloor \eta$ applied coordinatewise satisfies:

$$\|R_\theta(x) - R_{Q_\eta(\theta)}(x)\|_p \leqslant \varepsilon.$$

$\boxed{\textbf{Number of bits} \propto \textbf{depth } L\textbf{:}\ \textit{necessary and sufficient}}$

Consequence 1: naive uniform quantization

Naive uniform quantization

$\eta > 0$ such that $Q_\eta(x) = \lfloor x/\eta \rfloor \eta$ applied coordinatewise satisfies:

$$\max_{x \in [0,1]^d} \|R_\theta(x) - R_{Q_\eta(\theta)}(x)\|_p \leqslant \varepsilon.$$

$\boxed{\textbf{Number of bits} \propto \textbf{depth } L\textbf{:} \ \textit{necessary and sufficient}}$

Consequence 1: naive uniform quantization

Naive uniform quantization

$\eta > 0$ such that $Q_\eta(x) = \lfloor x/\eta \rfloor \eta$ applied coordinatewise satisfies:

$$\max_{\theta \in \Theta_{L,W}(B)} \max_{x \in [0,1]^d} \|R_\theta(x) - R_{Q_\eta(\theta)}(x)\|_p \leqslant \varepsilon.$$

$\boxed{\textbf{Number of bits} \propto \textbf{depth } L\textbf{: } \textit{necessary and sufficient}}$

Consequence 2: approximation speed of quantized ReLU networks

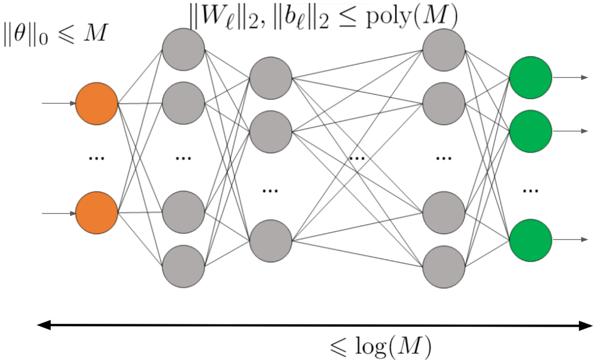
$$\boxed{\gamma^{*\text{approx}}(\mathcal{C}|\Sigma) := \text{largest } \gamma > 0 \text{ s.t. } \sup_{f \in \mathcal{C}} d(f, \Sigma_M) \underset{M \to \infty}{=} \mathcal{O}(M^{-\gamma})}$$

Increasingly complex $\Sigma = (\Sigma_M)_{M \in \mathbb{N}}$ with weights in $\mathbb{R}$

$$\gamma^{*\text{approx}}(\mathcal{C}|Q(\Sigma)) = \gamma^{*\text{approx}}(\mathcal{C}|\Sigma)?$$

Consequence 2: approximation speed of quantized ReLU networks

$(\log M)^2$ **bits per parameter are enough** for $\Sigma_M :=$ functions represented by a ReLU network:



Consequence 3: we recover and generalize known results

Recovered, improved and generalized: approximation results [2,3] of *quantized* ReLU networks

- improvement: number of bits
- generalization: to any $\mathcal{C}$ instead of Sovolev and L^{∞} spaces

[2] Deep Neural Network Approximation Theory. D. Elbrächter et al. 2021.
[3] Y. Ding et al. On the Universal Approximability and Complexity Bounds of Quantized ReLU Neural Networks. 2019.

Conclusion

$$\boxed{LB^{L-1} \lesssim K_{\Theta_{L,W}(B)} \lesssim WL \times LB^{L-1}}$$

- **Number of bits must be linear in the depth:** if $Q_\eta(x) = \lfloor x/\eta \rfloor \eta$ gives ε-accuracy $\max_{\theta \in \Theta_{L,W}(B)} \max_{x \in [0,1]^d} \|R_\theta(x) - R_{Q_\eta(\theta)}(x)\|_p \leqslant \varepsilon$
- $(\log M)^2$ **bits/parameter are enough:** same approximation speeds with quantized networks using uniform scalar quantization
- **Recovery, improvement and generalization** of known approximation results [4,5] on quantized ReLU networks

[4] Deep Neural Network Approximation Theory. D. Elbrächter et al. 2021.
[5] Y. Ding et al. On the Universal Approximability and Complexity Bounds of Quantized ReLU Neural Networks. 2019.

Perspective

Theory = number of bits must be linear in the depth: if $Q_\eta(x) = \lfloor x/\eta \rfloor \eta$ gives ε-accuracy $\max_{\theta \in \Theta_{L,W}(B)} \max_{x \in [0,1]^d} \|R_\theta(x) - R_{Q_\eta(\theta)}(x)\|_p \leqslant \varepsilon$

Practice = 1 bit is enough[6]: quantization-aware training, same performance on MNIST for a 3 hidden-layers with 1024 neurons per layer

[6] BinaryConnect: Training Deep Neural Networks with binary weights during propagations. Courbariaux et al. 2015.

Perspective

Theory = number of bits must be linear in the depth: if $Q_\eta(x) = \lfloor x/\eta \rfloor \eta$ gives ε-accuracy $\max_{\theta \in \Theta_{L,W}(B)} \max_{x \in [0,1]^d} \|R_\theta(x) - R_{Q_\eta(\theta)}(x)\|_p \leqslant \varepsilon$

Practice = 1 bit is enough[6]**:** quantization-aware training, same performance on MNIST for a 3 hidden-layers with 1024 neurons per layer

Fill the gap theory/practice?

[6] BinaryConnect: Training Deep Neural Networks with binary weights during propagations. Courbariaux et al. 2015.

Perspective

Theory = number of bits must be linear in the depth: if $Q_\eta(x) = \lfloor x/\eta \rfloor \eta$ gives ε-accuracy $\max_{\theta \in \Theta_{L,W}(B)} \max_{x \in [0,1]^d} \|R_\theta(x) - R_{Q_\eta(\theta)}(x)\|_p \leqslant \varepsilon$

Practice = 1 bit is enough[6]: quantization-aware training, same performance on MNIST for a 3 hidden-layers with 1024 neurons per layer

Fill the gap theory/practice?

- Less naive quantization?
- ε-accuracy on a smaller set $\Theta \subset \Theta_{L,W}(B)$? e.g., *parameters that can be learned in practice*
- not interested in every $\varepsilon > 0$?
- adapt to the architecture?

[6] BinaryConnect: Training Deep Neural Networks with binary weights during propagations. Courbariaux et al. 2015.

Perspective

Theory = number of bits must be linear in the depth: if $Q_\eta(x) = \lfloor x/\eta \rfloor \eta$ gives ε-accuracy $\max_{\theta \in \Theta_{L,W}(B)} \max_{x \in [0,1]^d} \|R_\theta(x) - R_{Q_\eta(\theta)}(x)\|_p \leqslant \varepsilon$

Practice = 1 bit is enough[6]: quantization-aware training, same performance on MNIST for a 3 hidden-layers with 1024 neurons per layer

Fill the gap theory/practice?

- Less naive quantization?
- ε-accuracy on a smaller set $\Theta \subset \Theta_{L,W}(B)$? e.g., *parameters that can be learned in practice*
- not interested in every $\varepsilon > 0$?
- adapt to the architecture?

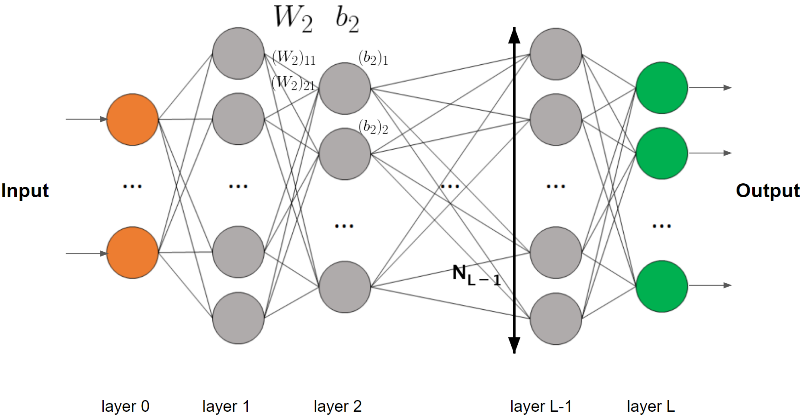
Thank you!

[6] BinaryConnect: Training Deep Neural Networks with binary weights during propagations. Courbariaux et al. 2015.

Notations : ReLU neural networks

Architecture: $(L, \mathbf{N})$ where

- $L \in \mathbb{N}$ is the number of layers : the depth
- $\mathbf{N} = (N_0, \ldots, N_L) \in \mathbb{N}^{L+1}$ with N_ℓ the width (number of neurons) of layer ℓ

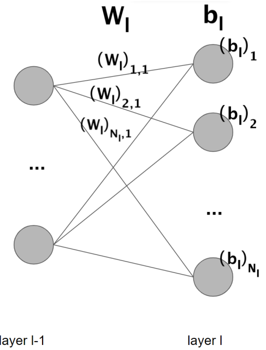


Notations : ReLU neural networks

Architecture: $(L, \mathbf{N})$ where

- $L \in \mathbb{N}$ is the number of layers : the depth
- $\mathbf{N} = (N_0, \ldots, N_L) \in \mathbb{N}^{L+1}$ with N_ℓ the width (number of neurons) of layer ℓ

Parameters: $\theta = (W_1, \ldots, W_L, b_1, \ldots, b_L)$ with $W_\ell \in \mathbb{R}^{N_\ell \times N_{\ell-1}}$ and $b_\ell \in \mathbb{R}^\ell$



Notations : ReLU neural networks

Architecture: $(L, \mathbf{N})$ where

- $L \in \mathbb{N}$ is the number of layers : the depth
- $\mathbf{N} = (N_0, \ldots, N_L) \in \mathbb{N}^{L+1}$ with N_ℓ the width (number of neurons) of layer ℓ

Parameters: $\theta = (W_1, \ldots, W_L, b_1, \ldots, b_L)$ with $W_\ell \in \mathbb{R}^{N_\ell \times N_{\ell-1}}$ and $b_\ell \in \mathbb{R}^\ell$

Realization of the network: $\boxed{R_\theta : \mathbb{R}^{N_0} \to \mathbb{R}^{N_L} \text{ the realization of } \theta\text{:}}$

$$R_\theta(x) = W_L \rho(\cdots(W_2 \rho(W_1 x + b_1) + b_2)\cdots) + b_L \text{ avec } \rho(x) = \max(0, x).$$