Quantitative Gaussian Approximation of Randomly Initialized Deep Neural Networks

Dario Trevisan
(Joint with A. Basteri)

Università di Pisa
dario.trevisan@unipi.it

Seminar Math4DSAIML
Trento
14 Oct 2022

Plan

1. Why random neural networks?
2. Our result
3. Numerical simulations
4. Extensions and future work

Plan

1. Why random neural networks?
2. Our result
3. Numerical simulations
4. Extensions and future work

Motivation

- Contemporary machine learning has seen a surge in applications of deep neural networks in
  - speech and visual recognition (classification)
  - feature extraction
  - sample generation
- The effort of understanding why deep learning methods work leads to new mathematical results in the areas of
  - probability
  - statistics
  - statistical physics
  - but also functional analysis, geometry, optimal control ...

Motivation

- Contemporary machine learning has seen a surge in applications of deep neural networks in
  - speech and visual recognition (classification)
  - feature extraction
  - sample generation

- The effort of understanding why deep learning methods work leads to new mathematical results in the areas of
  - probability
  - statistics
  - statistical physics
  - but also functional analysis, geometry, optimal control ...

Neural networks

Artificial neural networks are biologically-inspired ways to parametrize functions

$$f : \mathbb{R}^{n_0} \to \mathbb{R}^{n_L}$$

as stacked compositions of

- linear (or affine) maps
- non-linear functions (usually acting componentwise).

Much terminology is borrowed from neuroscience, e.g.

neurons, activation functions, connections, training etc.,

as well as some fundamental structures (e.g. convolutional architectures are inspired by the retina).

Neural networks

Artificial neural networks are biologically-inspired ways to parametrize functions

$$f : \mathbb{R}^{n_0} \to \mathbb{R}^{n_L}$$

as stacked compositions of

- linear (or affine) maps
- non-linear functions (usually acting componentwise).

Much terminology is borrowed from neuroscience, e.g.

neurons, activation functions, connections, training etc.,

as well as some fundamental structures (e.g. convolutional architectures are inspired by the retina).

Neural networks

Artificial neural networks are biologically-inspired ways to parametrize functions

$$f : \mathbb{R}^{n_0} \to \mathbb{R}^{n_L}$$

as stacked compositions of

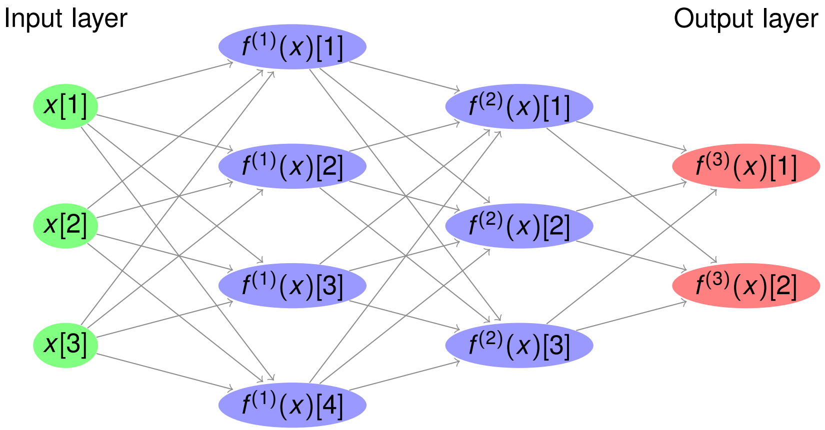
- linear (or affine) maps
- non-linear functions (usually acting componentwise).

Much terminology is borrowed from neuroscience, e.g.

neurons, activation functions, connections, training etc.,

as well as some fundamental structures (e.g. convolutional architectures are inspired by the retina).

Graphical representation of a fully connected feed-forward neural network with input size $n_0 = 3$, output size $n_3 = 2$ and layer sizes $n_1 = 4$, $n_2 = 3$:



Random neural networks

A successful approach focuses on the *scaling limit* of large neural networks whose parameters are randomly sampled.

Several reasons (besides interesting mathematics):

- Bayesian approach: prior distribution on model parameters (weights and biases to be updated after observations (training set in supervised learning).
- Large neural networks in practice are trained via iterative optimisation algorithms (SGD, stochastic gradient descent) which require careful (random!) initialization.
- It turns out that training only a fraction of the parameters (the last layer) of a randomly initialized network give still good performances in applications (reservoir computing).

Random neural networks

A successful approach focuses on the *scaling limit* of large neural networks whose parameters are randomly sampled.

Several reasons (besides interesting mathematics):

- Bayesian approach: prior distribution on model parameters (weights and biases to be updated after observations (training set in supervised learning).
- Large neural networks in practice are trained via iterative optimisation algorithms (SGD, stochastic gradient descent) which require careful (random!) initialization.
- It turns out that training only a fraction of the parameters (the last layer) of a randomly initialized network give still good performances in applications (reservoir computing).

Random neural networks

A successful approach focuses on the *scaling limit* of large neural networks whose parameters are randomly sampled.

Several reasons (besides interesting mathematics):

- Bayesian approach: prior distribution on model parameters (weights and biases to be updated after observations (training set in supervised learning).
- Large neural networks in practice are trained via iterative optimisation algorithms (SGD, stochastic gradient descent) which require careful (random!) initialization.
- It turns out that training only a fraction of the parameters (the last layer) of a randomly initialized network give still good performances in applications (reservoir computing).

Random neural networks

A successful approach focuses on the *scaling limit* of large neural networks whose parameters are randomly sampled.

Several reasons (besides interesting mathematics):

- Bayesian approach: prior distribution on model parameters (weights and biases to be updated after observations (training set in supervised learning).

- Large neural networks in practice are trained via iterative optimisation algorithms (SGD, stochastic gradient descent) which require careful (random!) initialization.

- It turns out that training only a fraction of the parameters (the last layer) of a randomly initialized network give still good performances in applications (reservoir computing).

Random neural networks

A successful approach focuses on the *scaling limit* of large neural networks whose parameters are randomly sampled.

Several reasons (besides interesting mathematics):

- Bayesian approach: prior distribution on model parameters (weights and biases to be updated after observations (training set in supervised learning).
- Large neural networks in practice are trained via iterative optimisation algorithms (SGD, stochastic gradient descent) which require careful (random!) initialization.
- It turns out that training only a fraction of the parameters (the last layer) of a randomly initialized network give still good performances in applications (reservoir computing).

Related literature

The study of random neural networks is indeed not new. Some milestones:

1958 Rosenblatt pioneering works with the perceptron

1996 Neal first proved that random wide shallow networks (one hidden layer) may converge to a Gaussian process

2018 Matthews et al. Lee et al. extended Neal to deep architectures (more hidden layers)

2019 Lee et al. realized that also after (lazy) training Gaussian behaviour is preserved (Neural Tangent Kernel NTK theory)

2018 Mei et al. in parallel study the mean field limit of large deep networks.

Related literature

The study of random neural networks is indeed not new. Some milestones:

1958 Rosenblatt pioneering works with the perceptron

1996 Neal first proved that random wide shallow networks (one hidden layer) may converge to a Gaussian process

2018 Matthews et al. Lee et al. extended Neal to deep architectures (more hidden layers)

2019 Lee et al. realized that also after (lazy) training Gaussian behaviour is preserved (Neural Tangent Kernel NTK theory)

2018 Mei et al. in parallel study the mean field limit of large deep networks.

Related literature

The study of random neural networks is indeed not new. Some milestones:

1958 Rosenblatt pioneering works with the perceptron

1996 Neal first proved that random wide shallow networks (one hidden layer) may converge to a Gaussian process

2018 Matthews et al. Lee et al. extended Neal to deep architectures (more hidden layers)

2019 Lee et al. realized that also after (lazy) training Gaussian behaviour is preserved (Neural Tangent Kernel NTK theory)

2018 Mei et al. in parallel study the mean field limit of large deep networks.

Related literature

The study of random neural networks is indeed not new. Some milestones:

1958 Rosenblatt pioneering works with the perceptron

1996 Neal first proved that random wide shallow networks (one hidden layer) may converge to a Gaussian process

2018 Matthews et al. Lee et al. extended Neal to deep architectures (more hidden layers)

2019 Lee et al. realized that also after (lazy) training Gaussian behaviour is preserved (Neural Tangent Kernel NTK theory)

2018 Mei et al. in parallel study the mean field limit of large deep networks.

Related literature

The study of random neural networks is indeed not new. Some milestones:

1958 Rosenblatt pioneering works with the perceptron

1996 Neal first proved that random wide shallow networks (one hidden layer) may converge to a Gaussian process

2018 Matthews et al. Lee et al. extended Neal to deep architectures (more hidden layers)

2019 Lee et al. realized that also after (lazy) training Gaussian behaviour is preserved (Neural Tangent Kernel NTK theory)

2018 Mei et al. in parallel study the mean field limit of large deep networks.

Related literature

The study of random neural networks is indeed not new. Some milestones:

1958 Rosenblatt pioneering works with the perceptron

1996 Neal first proved that random wide shallow networks (one hidden layer) may converge to a Gaussian process

2018 Matthews et al. Lee et al. extended Neal to deep architectures (more hidden layers)

2019 Lee et al. realized that also after (lazy) training Gaussian behaviour is preserved (Neural Tangent Kernel NTK theory)

2018 Mei et al. in parallel study the mean field limit of large deep networks.

Plan

1. Why random neural networks?
2. Our result
3. Numerical simulations
4. Extensions and future work

Our result in brief

1. We provide quantitative proof of the Gaussian behaviour of deep fully connected neural networks with random parameters at initialization.
2. We complement the works by Matthews et al., Lee et al., and later ones providing explicit rates for the convergence for deep networks.
3. We use the Wasserstein distance of order 2 (we believe in fact that for any order $p \geq 1$ similar rates should hold as well).

Our result in brief

1. We provide quantitative proof of the Gaussian behaviour of deep fully connected neural networks with random parameters at initialization.
2. We complement the works by Matthews et al., Lee et al., and later ones providing explicit rates for the convergence for deep networks.
3. We use the Wasserstein distance of order 2 (we believe in fact that for any order $p \geq 1$ similar rates should hold as well).

Our result in brief

1. We provide quantitative proof of the Gaussian behaviour of deep fully connected neural networks with random parameters at initialization.
2. We complement the works by Matthews et al., Lee et al., and later ones providing explicit rates for the convergence for deep networks.
3. We use the Wasserstein distance of order 2 (we believe in fact that for any order $p \geq 1$ similar rates should hold as well).

Our result in brief

1. We provide quantitative proof of the Gaussian behaviour of deep fully connected neural networks with random parameters at initialization.
2. We complement the works by Matthews et al., Lee et al., and later ones providing explicit rates for the convergence for deep networks.
3. We use the Wasserstein distance of order 2 (we believe in fact that for any order $p \geq 1$ similar rates should hold as well).

Notation: Wasserstein distance of order 2

Given probabilities p, q on $\mathbb{R}^d$, define

$$\mathcal{W}_2(p, q) = \inf \left\{ \sqrt{\mathbb{E}\left[\|X - Y\|^2\right]} \, : X,\ Y \text{ random variables with } \mathbb{P}_X = p, \mathbb{P}_Y = q \right\}$$

- With a slight abuse we write $\mathcal{W}_2(X, Y)$ instead of $\mathcal{W}_2(\mathbb{P}_X, \mathbb{P}_Y)$.
- The triangle inequality holds:

$$\mathcal{W}_2(X, Z) \leq \mathcal{W}_2(X, Y) + \mathcal{W}_2(Y, Z).$$

- A sequence $(X_n)_n$ converges to X, i.e.,

$$\lim_{n\to\infty} \mathcal{W}_2(X_n, X) = 0$$

if and only if

$$\lim_{n\to\infty} X_n \quad \text{in law} \quad \text{and} \quad \lim_{n\to\infty} \mathbb{E}\left[X_n \otimes X_n\right] = \mathbb{E}\left[X \otimes X\right].$$

Notation: Wasserstein distance of order 2

Given probabilities p, q on $\mathbb{R}^d$, define

$$\mathcal{W}_2(p,q) = \inf \left\{ \sqrt{\mathbb{E}\left[\|X - Y\|^2\right]} \,:\, X,\ Y \text{ random variables with } \mathbb{P}_X = p, \mathbb{P}_Y = q \right\}$$

- With a slight abuse we write $\mathcal{W}_2(X, Y)$ instead of $\mathcal{W}_2(\mathbb{P}_X, \mathbb{P}_Y)$.
- The triangle inequality holds:

$$\mathcal{W}_2(X,Z) \leq \mathcal{W}_2(X,Y) + \mathcal{W}_2(Y,Z).$$

- A sequence $(X_n)_n$ converges to X, i.e.,

$$\lim_{n\to\infty} \mathcal{W}_2(X_n, X) = 0$$

if and only if

$$\lim_{n\to\infty} X_n \quad \text{in law} \qquad \text{and} \qquad \lim_{n\to\infty} \mathbb{E}\left[X_n \otimes X_n\right] = \mathbb{E}\left[X \otimes X\right].$$

Notation: Wasserstein distance of order 2

Given probabilities p, q on $\mathbb{R}^d$, define

$$\mathcal{W}_2(p,q) = \inf \left\{ \sqrt{\mathbb{E}\left[\|X - Y\|^2\right]} : X, Y \text{ random variables with } \mathbb{P}_X = p, \mathbb{P}_Y = q \right\}$$

- With a slight abuse we write $\mathcal{W}_2(X, Y)$ instead of $\mathcal{W}_2(\mathbb{P}_X, \mathbb{P}_Y)$.
- The triangle inequality holds:

$$\mathcal{W}_2(X,Z) \le \mathcal{W}_2(X,Y) + \mathcal{W}_2(Y,Z).$$

- A sequence $(X_n)_n$ converges to X, i.e.,

$$\lim_{n\to\infty} \mathcal{W}_2(X_n, X) = 0$$

if and only if

$$\lim_{n\to\infty} X_n \quad \text{in law} \quad \text{and} \quad \lim_{n\to\infty} \mathbb{E}\left[X_n \otimes X_n\right] = \mathbb{E}\left[X \otimes X\right].$$

Notation: Wasserstein distance of order 2

Given probabilities p, q on $\mathbb{R}^d$, define

$$\mathcal{W}_2(p,q) = \inf \left\{ \sqrt{\mathbb{E}\left[\|X - Y\|^2\right]} : X, Y \text{ random variables with } \mathbb{P}_X = p, \mathbb{P}_Y = q \right\}$$

- With a slight abuse we write $\mathcal{W}_2(X, Y)$ instead of $\mathcal{W}_2(\mathbb{P}_X, \mathbb{P}_Y)$.
- The triangle inequality holds:

$$\mathcal{W}_2(X, Z) \leq \mathcal{W}_2(X, Y) + \mathcal{W}_2(Y, Z).$$

- A sequence $(X_n)_n$ converges to X, i.e.,

$$\lim_{n\to\infty} \mathcal{W}_2(X_n, X) = 0$$

if and only if

$$\lim_{n\to\infty} X_n \quad \text{in law} \quad \text{and} \quad \lim_{n\to\infty} \mathbb{E}\left[X_n \otimes X_n\right] = \mathbb{E}\left[X \otimes X\right].$$

Notation: Wasserstein distance of order 2

Given probabilities p, q on $\mathbb{R}^d$, define

$$\mathcal{W}_2(p,q) = \inf\left\{ \sqrt{\mathbb{E}\left[\|X-Y\|^2\right]} : X,\ Y \text{ random variables with } \mathbb{P}_X = p, \mathbb{P}_Y = q \right\}$$

- With a slight abuse we write $\mathcal{W}_2(X,Y)$ instead of $\mathcal{W}_2(\mathbb{P}_X,\mathbb{P}_Y)$.
- The triangle inequality holds:

$$\mathcal{W}_2(X,Z) \le \mathcal{W}_2(X,Y) + \mathcal{W}_2(Y,Z).$$

- A sequence $(X_n)_n$ converges to X, i.e.,

$$\lim_{n\to\infty} \mathcal{W}_2(X_n, X) = 0$$

  if and only if

$$\lim_{n\to\infty} X_n \quad \text{in law} \quad \text{and} \quad \lim_{n\to\infty} \mathbb{E}\left[X_n \otimes X_n\right] = \mathbb{E}\left[X \otimes X\right].$$

Notation: Gaussian variables

- Recall that a (real) Gaussian random variable X with mean μ and variance $\sigma^2 > 0$ has absolutely continuous law $\mathbb{P}_X$ with density

$$\exp\left(-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right)\frac{1}{\sqrt{2\pi\sigma^2}},$$

  while if $\sigma^2 = 0$, $X = \mu$ is constant.
- A Gaussian variable with values in $\mathbb{R}^d$ by definition is such that

$$\langle v, X\rangle = \sum_{i=1}^{d} v[i]X[i]$$

  is real Gaussian for every (deterministic) $v \in \mathbb{R}^d$,
- Given any symmetric positive semi-definite $K \in \mathbb{R}^{d\times d}$, write

$$\mathcal{N}(K)$$

  for the law of any centred Gaussian distribution on $\mathbb{R}^S$ with covariance K, i.e.,

$$\mathbb{E}\left[X[i]\right] = 0, \quad \mathbb{E}\left[X[i]X[j]\right] = \Sigma[i,j] \quad \text{for every } i, j.$$

Notation: Gaussian variables

- Recall that a (real) Gaussian random variable X with mean μ and variance $\sigma^2 > 0$ has absolutely continuous law $\mathbb{P}_X$ with density

$$\exp\left(-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right)\frac{1}{\sqrt{2\pi\sigma^2}},$$

while if $\sigma^2 = 0$, $X = \mu$ is constant.

- A Gaussian variable with values in $\mathbb{R}^d$ by definition is such that

$$\langle v, X \rangle = \sum_{i=1}^{d} v[i]X[i]$$

is real Gaussian for every (deterministic) $v \in \mathbb{R}^d$,

- Given any symmetric positive semi-definite $K \in \mathbb{R}^{d\times d}$, write

$$\mathcal{N}(K)$$

for the law of any centred Gaussian distribution on $\mathbb{R}^S$ with covariance K, i.e.,

$$\mathbb{E}[X[i]] = 0, \quad \mathbb{E}[X[i]X[j]] = \Sigma[i,j] \quad \text{for every } i, j.$$

Notation: Gaussian variables

- Recall that a (real) Gaussian random variable X with mean μ and variance $\sigma^2 > 0$ has absolutely continuous law $\mathbb{P}_X$ with density

$$\exp\left(-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right)\frac{1}{\sqrt{2\pi\sigma^2}},$$

  while if $\sigma^2 = 0$, $X = \mu$ is constant.
- A Gaussian variable with values in $\mathbb{R}^d$ by definition is such that

$$\langle v, X\rangle = \sum_{i=1}^{d} v[i]X[i]$$

  is real Gaussian for every (deterministic) $v \in \mathbb{R}^d$,
- Given any symmetric positive semi-definite $K \in \mathbb{R}^{d\times d}$, write

$$\mathcal{N}(K)$$

  for the law of any centred Gaussian distribution on $\mathbb{R}^S$ with covariance K, i.e.,

$$\mathbb{E}\left[X[i]\right] = 0, \quad \mathbb{E}\left[X[i]X[j]\right] = \Sigma[i,j] \quad \text{for every } i, j.$$

Notation: Gaussian variables

- Recall that a (real) Gaussian random variable X with mean μ and variance $\sigma^2 > 0$ has absolutely continuous law $\mathbb{P}_X$ with density

$$\exp\left(-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right)\frac{1}{\sqrt{2\pi\sigma^2}},$$

while if $\sigma^2 = 0$, $X = \mu$ is constant.
- A Gaussian variable with values in $\mathbb{R}^d$ by definition is such that

$$\langle v, X\rangle = \sum_{i=1}^{d} v[i]X[i]$$

is real Gaussian for every (deterministic) $v \in \mathbb{R}^d$,
- Given any symmetric positive semi-definite $K \in \mathbb{R}^{d\times d}$, write

$$\mathcal{N}(K)$$

for the law of any centred Gaussian distribution on $\mathbb{R}^S$ with covariance K, i.e.,

$$\mathbb{E}\left[X[i]\right] = 0, \quad \mathbb{E}\left[X[i]X[j]\right] = \Sigma[i,j] \quad \text{for every } i, j.$$

Notation: neural networks

We consider a (fully connected) neural network $f : \mathbb{R}^{n_0} \to \mathbb{R}^{n_L}$, with parameters:

- the total number of layers (including input and output): $L + 1$
- layer sizes n_0 (input), $n_1, \ldots, n_{L-1}$ (hidden), n_L output
- parameters: weights $\mathbf{W} = (W^{(\ell)})_{\ell=0}^{L-1}$ and biases $\mathbf{b} = (b^{(\ell)})_{\ell=0}^{L-1}$,

$$W^{(\ell)} \in \mathbb{R}^{n_{\ell+1} \times n_\ell}, \quad b^{(\ell)} \in \mathbb{R}^{n_{\ell+1}},$$

- (Lipschitz) activation function $\sigma : \mathbb{R} \to \mathbb{R}$, e.g. ReLU $\sigma(z) = \max\{0, z\}$.

Recursive definition:

$$f^{(1)} : \mathbb{R}^{n_0} \to \mathbb{R}^{n_1}, \quad f^{(1)}(x) = W^{(0)}x + b^{(0)},$$

and, for $\ell = 2, \ldots, L$,

$$f^{(\ell)} : \mathbb{R}^{n_0} \to \mathbb{R}^{n_\ell}, \quad f^{(\ell)}(x) = W^{(\ell-1)}\sigma(f^{(\ell-1)}(x)) + b^{(\ell-1)},$$

where the activation function σ is understood componentwise.

Notation: neural networks

We consider a (fully connected) neural network $f : \mathbb{R}^{n_0} \to \mathbb{R}^{n_L}$, with parameters:

- the total number of layers (including input and output): $L+1$
- layer sizes n_0 (input), $n_1, \ldots, n_{L-1}$ (hidden), n_L output
- parameters: weights $\mathbf{W} = (W^{(\ell)})_{\ell=0}^{L-1}$ and biases $\mathbf{b} = (b^{(\ell)})_{\ell=0}^{L-1}$,

$$W^{(\ell)} \in \mathbb{R}^{n_{\ell+1} \times n_\ell}, \quad b^{(\ell)} \in \mathbb{R}^{n_{\ell+1}},$$

- (Lipschitz) activation function $\sigma : \mathbb{R} \to \mathbb{R}$, e.g. ReLU $\sigma(z) = \max\{0, z\}$.

Recursive definition:

$$f^{(1)} : \mathbb{R}^{n_0} \to \mathbb{R}^{n_1}, \quad f^{(1)}(x) = W^{(0)}x + b^{(0)},$$

and, for $\ell = 2, \ldots, L$,

$$f^{(\ell)} : \mathbb{R}^{n_0} \to \mathbb{R}^{n_\ell}, \quad f^{(\ell)}(x) = W^{(\ell-1)}\sigma(f^{(\ell-1)}(x)) + b^{(\ell-1)},$$

where the activation function σ is understood componentwise.

Notation: neural networks

We consider a (fully connected) neural network $f : \mathbb{R}^{n_0} \to \mathbb{R}^{n_L}$, with parameters:

- the total number of layers (including input and output): $L+1$
- layer sizes n_0 (input), n_1, …, n_{L-1} (hidden), n_L output
- parameters: weights $\mathbf{W} = (W^{(\ell)})_{\ell=0}^{L-1}$ and biases $\mathbf{b} = (b^{(\ell)})_{\ell=0}^{L-1}$,

$$W^{(\ell)} \in \mathbb{R}^{n_{\ell+1} \times n_\ell}, \quad b^{(\ell)} \in \mathbb{R}^{n_{\ell+1}},$$

- (Lipschitz) activation function $\sigma : \mathbb{R} \to \mathbb{R}$, e.g. ReLU $\sigma(z) = \max\{0, z\}$.

Recursive definition:

$$f^{(1)} : \mathbb{R}^{n_0} \to \mathbb{R}^{n_1}, \quad f^{(1)}(x) = W^{(0)}x + b^{(0)},$$

and, for $\ell = 2, \dots, L$,

$$f^{(\ell)} : \mathbb{R}^{n_0} \to \mathbb{R}^{n_\ell}, \quad f^{(\ell)}(x) = W^{(\ell-1)}\sigma(f^{(\ell-1)}(x)) + b^{(\ell-1)},$$

where the activation function σ is understood componentwise.

Notation: neural networks

We consider a (fully connected) neural network $f : \mathbb{R}^{n_0} \to \mathbb{R}^{n_L}$, with parameters:

- the total number of layers (including input and output): $L+1$
- layer sizes n_0 (input), n_1, …, n_{L-1} (hidden), n_L output
- parameters: weights $\mathbf{W} = (W^{(\ell)})_{\ell=0}^{L-1}$ and biases $\mathbf{b} = (b^{(\ell)})_{\ell=0}^{L-1}$,

$$W^{(\ell)} \in \mathbb{R}^{n_{\ell+1} \times n_\ell}, \quad b^{(\ell)} \in \mathbb{R}^{n_{\ell+1}},$$

- (Lipschitz) activation function $\sigma : \mathbb{R} \to \mathbb{R}$, e.g. ReLU $\sigma(z) = \max\{0, z\}$.

Recursive definition:

$$f^{(1)} : \mathbb{R}^{n_0} \to \mathbb{R}^{n_1}, \quad f^{(1)}(x) = W^{(0)}x + b^{(0)},$$

and, for $\ell = 2, \ldots, L$,

$$f^{(\ell)} : \mathbb{R}^{n_0} \to \mathbb{R}^{n_\ell}, \quad f^{(\ell)}(x) = W^{(\ell-1)}\sigma(f^{(\ell-1)}(x)) + b^{(\ell-1)},$$

where the activation function σ is understood componentwise.

Notation: neural networks

We consider a (fully connected) neural network $f : \mathbb{R}^{n_0} \to \mathbb{R}^{n_L}$, with parameters:

- the total number of layers (including input and output): $L+1$
- layer sizes n_0 (input), $n_1, \ldots, n_{L-1}$ (hidden), n_L output
- parameters: weights $\mathbf{W} = (W^{(\ell)})_{\ell=0}^{L-1}$ and biases $\mathbf{b} = (b^{(\ell)})_{\ell=0}^{L-1}$,

$$W^{(\ell)} \in \mathbb{R}^{n_{\ell+1} \times n_\ell}, \quad b^{(\ell)} \in \mathbb{R}^{n_{\ell+1}},$$

- (Lipschitz) activation function $\sigma : \mathbb{R} \to \mathbb{R}$, e.g. ReLU $\sigma(z) = \max\{0, z\}$.

Recursive definition:

$$f^{(1)} : \mathbb{R}^{n_0} \to \mathbb{R}^{n_1}, \quad f^{(1)}(x) = W^{(0)}x + b^{(0)},$$

and, for $\ell = 2, \ldots, L$,

$$f^{(\ell)} : \mathbb{R}^{n_0} \to \mathbb{R}^{n_\ell}, \quad f^{(\ell)}(x) = W^{(\ell-1)}\sigma(f^{(\ell-1)}(x)) + b^{(\ell-1)},$$

where the activation function σ is understood componentwise.

Notation: neural networks

We consider a (fully connected) neural network $f : \mathbb{R}^{n_0} \to \mathbb{R}^{n_L}$, with parameters:

- the total number of layers (including input and output): $L+1$
- layer sizes n_0 (input), n_1, …, n_{L-1} (hidden), n_L output
- parameters: weights $\mathbf{W} = (W^{(\ell)})_{\ell=0}^{L-1}$ and biases $\mathbf{b} = (b^{(\ell)})_{\ell=0}^{L-1}$,

$$W^{(\ell)} \in \mathbb{R}^{n_{\ell+1} \times n_\ell}, \quad b^{(\ell)} \in \mathbb{R}^{n_{\ell+1}},$$

- (Lipschitz) activation function $\sigma : \mathbb{R} \to \mathbb{R}$, e.g. ReLU $\sigma(z) = \max\{0, z\}$.

Recursive definition:

$$f^{(1)} : \mathbb{R}^{n_0} \to \mathbb{R}^{n_1}, \quad f^{(1)}(x) = W^{(0)}x + b^{(0)},$$

and, for $\ell = 2, \ldots, L$,

$$f^{(\ell)} : \mathbb{R}^{n_0} \to \mathbb{R}^{n_\ell}, \quad f^{(\ell)}(x) = W^{(\ell-1)}\sigma(f^{(\ell-1)}(x)) + b^{(\ell-1)},$$

where the activation function σ is understood componentwise.

Theorem (Basteri and T.)

Consider weights **W** and biases **b** that are independent Gaussian random variables, centred with

$$\mathbb{E}\left[(W_{i,j}^{(\ell)})^2\right] = \frac{1}{n_\ell}, \quad \mathbb{E}\left[(b_i^{(\ell)})^2\right] = 1, \quad \text{for every } \ell, i \text{ and } j.$$

Then, for every set of k inputs $\mathcal{X} = \{x_i\}_{i=1}^k \subseteq \mathbb{R}^{n_0}$, the law of the output $f^{(L)}[\mathcal{X}] = (f^{(L)}(x_i))_{i=1}^k$ is close to a centred Gaussian random variable $G^{(L)}[\mathcal{X}]$:

$$\mathcal{W}_2\left(f^{(L)}[\mathcal{X}], G^{(L)}[\mathcal{X}]\right) \leq C\sqrt{n_L}\sum_{\ell=1}^{L-1}\frac{1}{\sqrt{n_\ell}}.$$

The constant $C \in (0, \infty)$ depends on σ, $\mathcal{X}$ and the number of layers L, but not on the hidden or output layer sizes $(n_\ell)_{\ell=1}^L$.

Theorem (Basteri and T.)

Consider weights **W** and biases **b** that are independent Gaussian random variables, centred with

$$\mathbb{E}\left[(W_{i,j}^{(\ell)})^2\right] = \frac{1}{n_\ell}, \quad \mathbb{E}\left[(b_i^{(\ell)})^2\right] = 1, \quad \text{for every } \ell, i \text{ and } j.$$

Then, for every set of k inputs $\mathcal{X} = \{x_i\}_{i=1}^k \subseteq \mathbb{R}^{n_0}$, the law of the output $f^{(L)}[\mathcal{X}] = (f^{(L)}(x_i))_{i=1}^k$ is close to a centred Gaussian random variable $G^{(L)}[\mathcal{X}]$:

$$\mathcal{W}_2\left(f^{(L)}[\mathcal{X}], G^{(L)}[\mathcal{X}]\right) \leq C\sqrt{n_L}\sum_{\ell=1}^{L-1} \frac{1}{\sqrt{n_\ell}}.$$

The constant $C \in (0, \infty)$ depends on σ, $\mathcal{X}$ and the number of layers L, but not on the hidden or output layer sizes $(n_\ell)_{\ell=1}^L$.

Theorem (Basteri and T.)

Consider weights **W** and biases **b** that are independent Gaussian random variables, centred with

$$\mathbb{E}\left[(W_{i,j}^{(\ell)})^2\right] = \frac{1}{n_\ell}, \quad \mathbb{E}\left[(b_i^{(\ell)})^2\right] = 1, \quad \text{for every } \ell, i \text{ and } j.$$

Then, for every set of k inputs $\mathcal{X} = \{x_i\}_{i=1}^k \subseteq \mathbb{R}^{n_0}$, the law of the output $f^{(L)}[\mathcal{X}] = (f^{(L)}(x_i))_{i=1}^k$ is close to a centred Gaussian random variable $G^{(L)}[\mathcal{X}]$:

$$\mathcal{W}_2\left(f^{(L)}[\mathcal{X}], G^{(L)}[\mathcal{X}]\right) \leq C\sqrt{n_L}\sum_{\ell=1}^{L-1}\frac{1}{\sqrt{n_\ell}}.$$

The constant $C \in (0, \infty)$ depends on σ, $\mathcal{X}$ and the number of layers L, but not on the hidden or output layer sizes $(n_\ell)_{\ell=1}^L$.

Neural Network Gaussian process

- All n_L output neurons in the Gaussian approximation $G^{(L)}[\mathcal{X}]$ are i.i.d. variables (for any input).
- The covariance $K^{(L)}[\mathcal{X}]$ of $G^{(L)}[\mathcal{X}]$ depends on the activation function σ, the input $\mathcal{X}$ and the output dimension n_L (not on the hidden layer sizes $(n_\ell)_{\ell=1}^{L-1}$).
- In fact, $K^{(L)}[\mathcal{X}]$ is recursively computable (for simplicity let $n_L = 1$):

$$K^{(1)}[x, y] = \frac{1}{n_0} \langle x, y \rangle + 1 = \frac{1}{n_0} \sum_{i=1}^{n_0} x[i]y[i] + 1.$$

For $\ell = 2, \ldots, L$, define $(G^{(\ell-1)}(x))_{x \in \mathcal{X}}$ as a centred Gaussian random variable with covariance $K^{(\ell-1)}[\mathcal{X}]$ and let

$$K^{(\ell)}(x, y) = \mathbb{E}\left[\sigma(G^{(\ell-1)}(x))\sigma(G^{(\ell-1)}(y))\right] + 1.$$

Neural Network Gaussian process

- All n_L output neurons in the Gaussian approximation $G^{(L)}[\mathcal{X}]$ are i.i.d. variables (for any input).
- The covariance $K^{(L)}[\mathcal{X}]$ of $G^{(L)}[\mathcal{X}]$ depends on the activation function σ, the input $\mathcal{X}$ and the output dimension n_L (not on the hidden layer sizes $(n_\ell)_{\ell=1}^{L-1}$).
- In fact, $K^{(L)}[\mathcal{X}]$ is recursively computable (for simplicity let $n_L = 1$):

$$K^{(1)}[x, y] = \frac{1}{n_0} \langle x, y \rangle + 1 = \frac{1}{n_0} \sum_{i=1}^{n_0} x[i]y[i] + 1.$$

For $\ell = 2, \ldots, L$, define $(G^{(\ell-1)}(x))_{x \in \mathcal{X}}$ as a centred Gaussian random variable with covariance $K^{(\ell-1)}[\mathcal{X}]$ and let

$$K^{(\ell)}(x, y) = \mathbb{E}\left[\sigma(G^{(\ell-1)}(x))\sigma(G^{(\ell-1)}(y))\right] + 1.$$

Neural Network Gaussian process

- All n_L output neurons in the Gaussian approximation $G^{(L)}[\mathcal{X}]$ are i.i.d. variables (for any input).
- The covariance $K^{(L)}[\mathcal{X}]$ of $G^{(L)}[\mathcal{X}]$ depends on the activation function σ, the input $\mathcal{X}$ and the output dimension n_L (not on the hidden layer sizes $(n_\ell)_{\ell=1}^{L-1}$).
- In fact, $K^{(L)}[\mathcal{X}]$ is recursively computable (for simplicity let $n_L = 1$):

$$K^{(1)}[x, y] = \frac{1}{n_0} \langle x, y \rangle + 1 = \frac{1}{n_0} \sum_{i=1}^{n_0} x[i]y[i] + 1.$$

For $\ell = 2, \ldots, L$, define $(G^{(\ell-1)}(x))_{x \in \mathcal{X}}$ as a centred Gaussian random variable with covariance $K^{(\ell-1)}[\mathcal{X}]$ and let

$$K^{(\ell)}(x, y) = \mathbb{E}\left[\sigma(G^{(\ell-1)}(x))\sigma(G^{(\ell-1)}(y))\right] + 1.$$

Some features of our result

- The inequality

$$\mathcal{W}_2\left(f^{(L)}[\mathcal{X}], G^{(L)}[\mathcal{X}]\right) \le C\sqrt{n_L}\sum_{\ell=1}^{L-1}\frac{1}{\sqrt{n_\ell}}$$

entails convergence towards the Gaussian law in the wide limit $n_\ell \to \infty$ for $\ell = 1, \ldots, L-1$.

- The constant C is explicit, also more general variances for weights and biases can be considered.
- In the deep limit $L \to \infty$ each contribution $\sqrt{n_L}/\sqrt{n_\ell}$ naturally associated to the ℓ-th hidden layer is weighted by an exponential factor (product of the standard deviations of weights).

Some features of our result

- The inequality

$$\mathcal{W}_2\left(f^{(L)}[\mathcal{X}], G^{(L)}[\mathcal{X}]\right) \leq C\sqrt{n_L}\sum_{\ell=1}^{L-1}\frac{1}{\sqrt{n_\ell}}$$

  entails convergence towards the Gaussian law in the wide limit $n_\ell \to \infty$ for $\ell = 1, \ldots, L-1$.
- The constant C is explicit, also more general variances for weights and biases can be considered.
- In the deep limit $L \to \infty$ each contribution $\sqrt{n_L}/\sqrt{n_\ell}$ naturally associated to the ℓ-th hidden layer is weighted by an exponential factor (product of the standard deviations of weights).

Some features of our result

- The inequality

$$\mathcal{W}_2\left(f^{(L)}[\mathcal{X}], G^{(L)}[\mathcal{X}]\right) \leq C\sqrt{n_L}\sum_{\ell=1}^{L-1}\frac{1}{\sqrt{n_\ell}}$$

  entails convergence towards the Gaussian law in the wide limit $n_\ell \to \infty$ for $\ell = 1, \ldots, L-1$.
- The constant C is explicit, also more general variances for weights and biases can be considered.
- In the deep limit $L \to \infty$ each contribution $\sqrt{n_L}/\sqrt{n_\ell}$ naturally associated to the ℓ-th hidden layer is weighted by an exponential factor (product of the standard deviations of weights).

Further properties of $\mathcal{W}_2$

We collect some useful (but elementary) properties of $\mathcal{W}_2$.

- If Z is independent of X and Y, then

$$\mathcal{W}_2(X+Z, Y+Z) \leq \mathcal{W}_2(X, Y).$$

- Convexity of squared $\mathcal{W}_2$: given random variables X, Y, Z, then

$$\mathcal{W}_2^2(X, Y) \leq \int_{\mathbb{R}^T} \mathcal{W}_2^2(\mathbb{P}_{X|Z=z}, \mathbb{P}_Y) d\mathbb{P}_Z(z)$$

- if X, Y are centred Gaussian random variables with covariances $\Sigma(X)$, $\Sigma(Y)$, then

$$\mathcal{W}_2(X, Y) \leq \left\| \sqrt{\Sigma(X)} - \sqrt{\Sigma(Y)} \right\|,$$

with $\|\cdot\|$ the operator norm and $\sqrt{\cdot}$ in the sense of functional calculus.

Further properties of $\mathcal{W}_2$

We collect some useful (but elementary) properties of $\mathcal{W}_2$.

- If Z is independent of X and Y, then

$$\mathcal{W}_2(X+Z, Y+Z) \leq \mathcal{W}_2(X, Y).$$

- Convexity of squared $\mathcal{W}_2$: given random variables X, Y, Z, then

$$\mathcal{W}_2^2(X, Y) \leq \int_{\mathbb{R}^T} \mathcal{W}_2^2(\mathbb{P}_{X|Z=z}, \mathbb{P}_Y) d\mathbb{P}_Z(z)$$

- if X, Y are centred Gaussian random variables with covariances $\Sigma(X)$, $\Sigma(Y)$, then

$$\mathcal{W}_2(X, Y) \leq \left\| \sqrt{\Sigma(X)} - \sqrt{\Sigma(Y)} \right\|,$$

with $\|\cdot\|$ the operator norm and $\sqrt{\cdot}$ in the sense of functional calculus.

Further properties of $\mathcal{W}_2$

We collect some useful (but elementary) properties of $\mathcal{W}_2$.

- If Z is independent of X and Y, then

$$\mathcal{W}_2(X+Z, Y+Z) \leq \mathcal{W}_2(X, Y).$$

- Convexity of squared $\mathcal{W}_2$: given random variables X, Y, Z, then

$$\mathcal{W}_2^2(X, Y) \leq \int_{\mathbb{R}^T} \mathcal{W}_2^2(\mathbb{P}_{X|Z=z}, \mathbb{P}_Y) d\mathbb{P}_Z(z)$$

- if X, Y are centred Gaussian random variables with covariances $\Sigma(X)$, $\Sigma(Y)$, then

$$\mathcal{W}_2(X, Y) \leq \left\| \sqrt{\Sigma(X)} - \sqrt{\Sigma(Y)} \right\|,$$

  with $\|\cdot\|$ the operator norm and $\sqrt{\cdot}$ in the sense of functional calculus.

Idea of proof

The Gaussian limit is due to a combination, in each layer, of the central limit theorem (CLT) scaling for the weights and the almost independence of the neurons.

We argue by induction over the layers:

- For one hidden layer exact independence holds $\rightarrow$ straightforward application of CLT.
- We use the triangle inequality for $\mathcal{W}_2$ and the inductive assumption $\rightarrow$ the Gaussian approximation yields exact independence.
- We bound the error terms using the convexity inequality for the squared $\mathcal{W}_2$ and the explicit optimal transport cost between Gaussians.

Base case

The case $\ell = 1$ is straightforward, since

$$f^{(1)}(x) = W^{(0)}x + b^{(0)}$$

is a linear function of the Gaussian variable $W^{(0)}$ and $b^{(0)}$, thus $f^{(1)}[\mathcal{X}]$ has Gaussian law, centred with covariance

$$\begin{aligned}
\Sigma\left(f^{(1)}[\mathcal{X}]\right) &= \Sigma\left((W^{(0)} \otimes \mathrm{Id}_k)\mathcal{X} + b^{(0)} \otimes 1_k\right) \\
&= \Sigma\left((W^{(0)} \otimes \mathrm{Id}_k)\mathcal{X}\right) + \Sigma\left(b^{(0)} \otimes 1_k\right) \quad \text{by independence,} \\
&= \mathrm{Id}_{n_1} \otimes K^{(1)}[\mathcal{X}, \mathcal{X}],
\end{aligned}$$

Induction step

We assume the thesis for $1 \leq \ell < L - 1$ and prove it for $\ell + 1$.

- Consider any probability space where random variables with the same laws as $f^{(\ell)} = f^{(\ell)}[\mathcal{X}]$ and $G^{(\ell)} = G^{(\ell)}[\mathcal{X}]$ are jointly defined.
- (Possibly enlarging the space) assume that $W^{(\ell)}$ and $b^{(\ell)}$ are also defined and independent of $f^{(\ell)}$ and $G^{(\ell)}$.
- Define auxiliary random variables

$$h^{(\ell+1)} = (W^{(\ell)} \otimes \mathrm{Id}_k)\sigma\left(G^{(\ell)}\right), \quad g^{(\ell+1)} = h^{(\ell+1)} + b^{(\ell)} \otimes 1_k.$$

- By the triangle inequality,

$$\mathcal{W}_2\left(f^{(\ell+1)}, G^{(\ell+1)}\right) \leq \mathcal{W}_2\left(f^{(\ell+1)}, g^{(\ell+1)}\right) + \mathcal{W}_2\left(g^{(\ell+1)}, G^{(\ell+1)}\right),$$

and bound separately the two terms.

Induction step

We assume the thesis for $1 \leq \ell < L-1$ and prove it for $\ell+1$.

- Consider any probability space where random variables with the same laws as $f^{(\ell)} = f^{(\ell)}[\mathcal{X}]$ and $G^{(\ell)} = G^{(\ell)}[\mathcal{X}]$ are jointly defined.
- (Possibly enlarging the space) assume that $W^{(\ell)}$ and $b^{(\ell)}$ are also defined and independent of $f^{(\ell)}$ and $G^{(\ell)}$.
- Define auxiliary random variables

$$h^{(\ell+1)} = (W^{(\ell)} \otimes \mathrm{Id}_k)\sigma\left(G^{(\ell)}\right), \quad g^{(\ell+1)} = h^{(\ell+1)} + b^{(\ell)} \otimes 1_k.$$

- By the triangle inequality,

$$\mathcal{W}_2\left(f^{(\ell+1)}, G^{(\ell+1)}\right) \leq \mathcal{W}_2\left(f^{(\ell+1)}, g^{(\ell+1)}\right) + \mathcal{W}_2\left(g^{(\ell+1)}, G^{(\ell+1)}\right),$$

and bound separately the two terms.

Induction step

We assume the thesis for $1 \leq \ell < L-1$ and prove it for $\ell+1$.

- Consider any probability space where random variables with the same laws as $f^{(\ell)} = f^{(\ell)}[\mathcal{X}]$ and $G^{(\ell)} = G^{(\ell)}[\mathcal{X}]$ are jointly defined.
- (Possibly enlarging the space) assume that $W^{(\ell)}$ and $b^{(\ell)}$ are also defined and independent of $f^{(\ell)}$ and $G^{(\ell)}$.
- Define auxiliary random variables

$$h^{(\ell+1)} = (W^{(\ell)} \otimes \mathrm{Id}_k)\sigma\left(G^{(\ell)}\right), \quad g^{(\ell+1)} = h^{(\ell+1)} + b^{(\ell)} \otimes 1_k.$$

- By the triangle inequality,

$$\mathcal{W}_2\left(f^{(\ell+1)}, G^{(\ell+1)}\right) \leq \mathcal{W}_2\left(f^{(\ell+1)}, g^{(\ell+1)}\right) + \mathcal{W}_2\left(g^{(\ell+1)}, G^{(\ell+1)}\right),$$

and bound separately the two terms.

Induction step

We assume the thesis for $1 \leq \ell < L - 1$ and prove it for $\ell + 1$.

- Consider any probability space where random variables with the same laws as $f^{(\ell)} = f^{(\ell)}[\mathcal{X}]$ and $G^{(\ell)} = G^{(\ell)}[\mathcal{X}]$ are jointly defined.
- (Possibly enlarging the space) assume that $W^{(\ell)}$ and $b^{(\ell)}$ are also defined and independent of $f^{(\ell)}$ and $G^{(\ell)}$.
- Define auxiliary random variables

$$h^{(\ell+1)} = (W^{(\ell)} \otimes \mathrm{Id}_k)\sigma\left(G^{(\ell)}\right), \quad g^{(\ell+1)} = h^{(\ell+1)} + b^{(\ell)} \otimes 1_k.$$

- By the triangle inequality,

$$\mathcal{W}_2\left(f^{(\ell+1)}, G^{(\ell+1)}\right) \leq \mathcal{W}_2\left(f^{(\ell+1)}, g^{(\ell+1)}\right) + \mathcal{W}_2\left(g^{(\ell+1)}, G^{(\ell+1)}\right),$$

and bound separately the two terms.

First term

- We bound

$$\begin{aligned}\mathcal{W}_2^2\left(f^{(\ell+1)}, g^{(\ell+1)}\right) &= \mathcal{W}_2^2\left(W^{(\ell)}\sigma(f^{(\ell)}) + b^{(\ell)}, W^{(\ell)}\sigma(G^{(\ell)}) + b^{(\ell)}\right)\\ &\le \mathcal{W}_2^2\left(W^{(\ell)}\sigma(f^{(\ell)}), W^{(\ell)}\sigma(G^{(\ell)})\right)\\ &\le \mathbb{E}\left[\left\| W^{(\ell)}\sigma(f^{(\ell)}) - W^{(\ell)}\sigma(G^{(\ell)})\right\|^2\right]\end{aligned}$$

- By conditioning upon $f^{(\ell)}$ and $G^{(\ell)}$, we obtain

$$\mathbb{E}\left[\left\| W^{(\ell)}\sigma(f^{(\ell)}) - W^{(\ell)}\sigma(G^{(\ell)})\right\|^2 \,\middle|\, f^{(\ell)}, G^{(\ell)}\right] = \frac{n_{\ell+1}}{n_\ell}\left\|\sigma(f^{(\ell)}) - \sigma(G^{(\ell)})\right\|^2.$$

- Finally, since σ is Lipschitz,

$$\left\|\sigma(f^{(\ell)}) - \sigma(G^{(\ell)})\right\|^2 \le \mathrm{Lip}(\sigma)^2\left\| f^{(\ell)} - G^{(\ell)}\right\|^2.$$

First term

- We bound

$$\mathcal{W}_2^2\left(f^{(\ell+1)}, g^{(\ell+1)}\right) = \mathcal{W}_2^2\left(W^{(\ell)}\sigma(f^{(\ell)}) + b^{(\ell)}, W^{(\ell)}\sigma(G^{(\ell)}) + b^{(\ell)}\right)$$
$$\leq \mathcal{W}_2^2\left(W^{(\ell)}\sigma(f^{(\ell)}), W^{(\ell)}\sigma(G^{(\ell)})\right)$$
$$\leq \mathbb{E}\left[\left\| W^{(\ell)}\sigma(f^{(\ell)}) - W^{(\ell)}\sigma(G^{(\ell)})\right\|^2\right]$$

- By conditioning upon $f^{(\ell)}$ and $G^{(\ell)}$, we obtain

$$\mathbb{E}\left[\left\| W^{(\ell)}\sigma(f^{(\ell)}) - W^{(\ell)}\sigma(G^{(\ell)})\right\|^2 \,\middle|\, f^{(\ell)}, G^{(\ell)}\right] = \frac{n_{\ell+1}}{n_\ell}\left\|\sigma(f^{(\ell)}) - \sigma(G^{(\ell)})\right\|^2.$$

- Finally, since σ is Lipschitz,

$$\left\|\sigma(f^{(\ell)}) - \sigma(G^{(\ell)})\right\|^2 \leq \mathrm{Lip}(\sigma)^2 \left\| f^{(\ell)} - G^{(\ell)}\right\|^2.$$

First term

- We bound

$$\mathcal{W}_2^2\left(f^{(\ell+1)}, g^{(\ell+1)}\right) = \mathcal{W}_2^2\left(W^{(\ell)}\sigma(f^{(\ell)}) + b^{(\ell)}, W^{(\ell)}\sigma(G^{(\ell)}) + b^{(\ell)}\right)$$
$$\leq \mathcal{W}_2^2\left(W^{(\ell)}\sigma(f^{(\ell)}), W^{(\ell)}\sigma(G^{(\ell)})\right)$$
$$\leq \mathbb{E}\left[\left\|W^{(\ell)}\sigma(f^{(\ell)}) - W^{(\ell)}\sigma(G^{(\ell)})\right\|^2\right]$$

- By conditioning upon $f^{(\ell)}$ and $G^{(\ell)}$, we obtain

$$\mathbb{E}\left[\left\|W^{(\ell)}\sigma(f^{(\ell)}) - W^{(\ell)}\sigma(G^{(\ell)})\right\|^2 \,\middle|\, f^{(\ell)}, G^{(\ell)}\right] = \frac{n_{\ell+1}}{n_\ell}\left\|\sigma(f^{(\ell)}) - \sigma(G^{(\ell)})\right\|^2.$$

- Finally, since σ is Lipschitz,

$$\left\|\sigma(f^{(\ell)}) - \sigma(G^{(\ell)})\right\|^2 \leq \mathrm{Lip}(\sigma)^2\left\|f^{(\ell)} - G^{(\ell)}\right\|^2.$$

First term

- We bound

$$\begin{aligned}\mathcal{W}_2^2\left(f^{(\ell+1)}, g^{(\ell+1)}\right) &= \mathcal{W}_2^2\left(W^{(\ell)}\sigma(f^{(\ell)}) + b^{(\ell)}, W^{(\ell)}\sigma(G^{(\ell)}) + b^{(\ell)}\right)\\ &\leq \mathcal{W}_2^2\left(W^{(\ell)}\sigma(f^{(\ell)}), W^{(\ell)}\sigma(G^{(\ell)})\right)\\ &\leq \mathbb{E}\left[\left\|W^{(\ell)}\sigma(f^{(\ell)}) - W^{(\ell)}\sigma(G^{(\ell)})\right\|^2\right]\end{aligned}$$

- By conditioning upon $f^{(\ell)}$ and $G^{(\ell)}$, we obtain

$$\mathbb{E}\left[\left\|W^{(\ell)}\sigma(f^{(\ell)}) - W^{(\ell)}\sigma(G^{(\ell)})\right\|^2 \,\middle|\, f^{(\ell)}, G^{(\ell)}\right] = \frac{n_{\ell+1}}{n_\ell}\left\|\sigma(f^{(\ell)}) - \sigma(G^{(\ell)})\right\|^2.$$

- Finally, since σ is Lipschitz,

$$\left\|\sigma(f^{(\ell)}) - \sigma(G^{(\ell)})\right\|^2 \leq \mathrm{Lip}(\sigma)^2\left\|f^{(\ell)} - G^{(\ell)}\right\|^2.$$

First term

- We bound

$$\begin{aligned}\mathcal{W}_2^2\left(f^{(\ell+1)}, g^{(\ell+1)}\right) &= \mathcal{W}_2^2\left(W^{(\ell)}\sigma(f^{(\ell)}) + b^{(\ell)}, W^{(\ell)}\sigma(G^{(\ell)}) + b^{(\ell)}\right)\\ &\leq \mathcal{W}_2^2\left(W^{(\ell)}\sigma(f^{(\ell)}), W^{(\ell)}\sigma(G^{(\ell)})\right)\\ &\leq \mathbb{E}\left[\left\| W^{(\ell)}\sigma(f^{(\ell)}) - W^{(\ell)}\sigma(G^{(\ell)})\right\|^2\right]\end{aligned}$$

- By conditioning upon $f^{(\ell)}$ and $G^{(\ell)}$, we obtain

$$\mathbb{E}\left[\left\| W^{(\ell)}\sigma(f^{(\ell)}) - W^{(\ell)}\sigma(G^{(\ell)})\right\|^2 \,\middle|\, f^{(\ell)}, G^{(\ell)}\right] = \frac{n_{\ell+1}}{n_\ell}\left\|\sigma(f^{(\ell)}) - \sigma(G^{(\ell)})\right\|^2.$$

- Finally, since σ is Lipschitz,

$$\left\|\sigma(f^{(\ell)}) - \sigma(G^{(\ell)})\right\|^2 \leq \mathrm{Lip}(\sigma)^2\left\| f^{(\ell)} - G^{(\ell)}\right\|^2.$$

Second term

- We prove
$$\mathcal{W}_2^2\left(g^{(\ell+1)}, G^{(\ell+1)}\right) \leq \frac{n_{\ell+1}}{n_\ell} C^{(\ell+1)}, \tag{1}$$
for an (explicit) finite constant $C^{(\ell+1)}$ depending on $\mathcal{X}$, σ and only.
- We assume that there is no bias (otherwise we remove it easily) and a Gaussian variable $G^{(\ell+1)}[\mathcal{X}]$ is also defined in the same space.
- Conditioning upon $G^{(\ell)} = z$, the random variable
$$g^{(\ell+1)} = (W^{(\ell)} \otimes \mathrm{Id}_k)\sigma\left(G^{(\ell)}\right) = (W^{(\ell)} \otimes \mathrm{Id}_k)\sigma(z)$$
has centred Gaussian law with covariance $\mathrm{Id}_{n_{\ell+1}} \otimes \overline{\sigma(z)}$ and
$$\overline{\sigma(z)}[j_1, j_2] = \frac{1}{n_\ell}\sum_{m=1}^{n_\ell} \sigma(z[m, j_1]) \otimes \sigma(z[m, j_2]).$$
- Using the bound for $\mathcal{W}_2$ between Gaussians,
$$\mathcal{W}_2^2\left(\mathbb{P}_{g^{(\ell+1)}|G^{(\ell)}=z}, \mathbb{P}_{G^{(\ell+1)}}\right) \leq n_{\ell+1}\left\|\sqrt{\overline{\sigma(z)}} - \sqrt{K^{(\ell+1)}}\right\|^2.$$

Second term

- We prove
$$\mathcal{W}_2^2\left(g^{(\ell+1)}, G^{(\ell+1)}\right) \le \frac{n_{\ell+1}}{n_\ell} C^{(\ell+1)}, \tag{1}$$
for an (explicit) finite constant $C^{(\ell+1)}$ depending on $\mathcal{X}$, σ and only.
- We assume that there is no bias (otherwise we remove it easily) and a Gaussian variable $G^{(\ell+1)}[\mathcal{X}]$ is also defined in the same space.
- Conditioning upon $G^{(\ell)} = z$, the random variable
$$g^{(\ell+1)} = (W^{(\ell)} \otimes \mathrm{Id}_k)\sigma\left(G^{(\ell)}\right) = (W^{(\ell)} \otimes \mathrm{Id}_k)\sigma(z)$$
has centred Gaussian law with covariance $\mathrm{Id}_{n_{\ell+1}} \otimes \overline{\sigma(z)}$ and
$$\overline{\sigma(z)}[j_1, j_2] = \frac{1}{n_\ell}\sum_{m=1}^{n_\ell} \sigma(z[m, j_1]) \otimes \sigma(z[m, j_2]).$$
- Using the bound for $\mathcal{W}_2$ between Gaussians,
$$\mathcal{W}_2^2\left(\mathbb{P}_{g^{(\ell+1)}|G^{(\ell)}=z}, \mathbb{P}_{G^{(\ell+1)}}\right) \le n_{\ell+1}\left\|\sqrt{\overline{\sigma(z)}} - \sqrt{K^{(\ell+1)}}\right\|^2.$$

Second term

- We prove

$$\mathcal{W}_2^2\left(g^{(\ell+1)}, G^{(\ell+1)}\right) \le \frac{n_{\ell+1}}{n_\ell} C^{(\ell+1)}, \tag{1}$$

  for an (explicit) finite constant $C^{(\ell+1)}$ depending on $\mathcal{X}$, σ and only.
- We assume that there is no bias (otherwise we remove it easily) and a Gaussian variable $G^{(\ell+1)}[\mathcal{X}]$ is also defined in the same space.
- Conditioning upon $G^{(\ell)} = z$, the random variable

$$g^{(\ell+1)} = (W^{(\ell)} \otimes \mathrm{Id}_k)\sigma\left(G^{(\ell)}\right) = (W^{(\ell)} \otimes \mathrm{Id}_k)\sigma(z)$$

  has centred Gaussian law with covariance $\mathrm{Id}_{n_{\ell+1}} \otimes \overline{\sigma(z)}$ and

$$\overline{\sigma(z)}[j_1, j_2] = \frac{1}{n_\ell}\sum_{m=1}^{n_\ell} \sigma(z[m, j_1]) \otimes \sigma(z[m, j_2]).$$

- Using the bound for $\mathcal{W}_2$ between Gaussians,

$$\mathcal{W}_2^2\left(\mathbb{P}_{g^{(\ell+1)}|G^{(\ell)}=z}, \mathbb{P}_{G^{(\ell+1)}}\right) \le n_{\ell+1}\left\|\sqrt{\overline{\sigma(z)}} - \sqrt{K^{(\ell+1)}}\right\|^2.$$

Second term

- We prove
$$\mathcal{W}_2^2\left(g^{(\ell+1)}, G^{(\ell+1)}\right) \leq \frac{n_{\ell+1}}{n_\ell} C^{(\ell+1)}, \tag{1}$$
for an (explicit) finite constant $C^{(\ell+1)}$ depending on $\mathcal{X}$, σ and only.
- We assume that there is no bias (otherwise we remove it easily) and a Gaussian variable $G^{(\ell+1)}[\mathcal{X}]$ is also defined in the same space.
- Conditioning upon $G^{(\ell)} = z$, the random variable
$$g^{(\ell+1)} = (W^{(\ell)} \otimes \mathrm{Id}_k)\sigma\left(G^{(\ell)}\right) = (W^{(\ell)} \otimes \mathrm{Id}_k)\sigma\left(z\right)$$
has centred Gaussian law with covariance $\mathrm{Id}_{n_{\ell+1}} \otimes \overline{\sigma(z)}$ and
$$\overline{\sigma(z)}[j_1, j_2] = \frac{1}{n_\ell}\sum_{m=1}^{n_\ell} \sigma\left(z[m, j_1]\right) \otimes \sigma\left(z[m, j_2]\right).$$
- Using the bound for $\mathcal{W}_2$ between Gaussians,
$$\mathcal{W}_2^2\left(\mathbb{P}_{g^{(\ell+1)}|G^{(\ell)}=z}, \mathbb{P}_{G^{(\ell+1)}}\right) \leq n_{\ell+1}\left\|\sqrt{\overline{\sigma(z)}} - \sqrt{K^{(\ell+1)}}\right\|^2.$$

By convexity of the squared $\mathcal{W}_2$,

$$\mathcal{W}_2^2\left(g^{(\ell+1)}, G^{(\ell+1)}\right) \leq n_{\ell+1}\mathbb{E}\left[\left\|\sqrt{\sigma(G^{\ell})} - \sqrt{K^{(\ell+1)}}\right\|^2\right].$$

The desired conclusion follows from a general lemma.

Lemma

Let $X = (X[i])_{i=1}^n$ be i.i.d. random variables with values in $\mathbb{R}^k$ (not identically null). Let $M = \mathbb{E}[X[1] \otimes X[1]]$ and define the $\mathbb{R}^{k\times k}$ valued variable

$$M_n = \frac{1}{n}\sum_{i=1}^{n} X[i] \otimes X[i].$$

Then,

$$\mathbb{E}\left[\left\|\sqrt{M_n} - \sqrt{M}\right\|^2\right] \leq \frac{\mathbb{E}\left[\|X[1] \otimes X[1] - M\|^2\right]}{n\lambda^+(M)},$$

where $\lambda^+(M) > 0$ denotes the smallest strictly positive eigenvalue of M.

By convexity of the squared $\mathcal{W}_2$,

$$\mathcal{W}_2^2\left(g^{(\ell+1)}, G^{(\ell+1)}\right) \leq n_{\ell+1}\mathbb{E}\left[\left\|\sqrt{\sigma(G^\ell)} - \sqrt{K^{(\ell+1)}}\right\|^2\right].$$

The desired conclusion follows from a general lemma.

Lemma

Let $X = (X[i])_{i=1}^n$ be i.i.d. random variables with values in $\mathbb{R}^k$ (not identically null). Let $M = \mathbb{E}[X[1] \otimes X[1]]$ and define the $\mathbb{R}^{k\times k}$ valued variable

$$M_n = \frac{1}{n}\sum_{i=1}^n X[i] \otimes X[i].$$

Then,

$$\mathbb{E}\left[\left\|\sqrt{M_n} - \sqrt{M}\right\|^2\right] \leq \frac{\mathbb{E}\left[\|X[1] \otimes X[1] - M\|^2\right]}{n\lambda^+(M)},$$

where $\lambda^+(M) > 0$ denotes the smallest strictly positive eigenvalue of M.

By convexity of the squared $\mathcal{W}_2$,

$$\mathcal{W}_2^2\left(g^{(\ell+1)}, G^{(\ell+1)}\right) \leq n_{\ell+1}\mathbb{E}\left[\left\|\sqrt{\sigma(G^\ell)} - \sqrt{K^{(\ell+1)}}\right\|^2\right].$$

The desired conclusion follows from a general lemma.

Lemma

Let $X = (X[i])_{i=1}^n$ be i.i.d. random variables with values in $\mathbb{R}^k$ (not identically null). Let $M = \mathbb{E}\left[X[1] \otimes X[1]\right]$ and define the $\mathbb{R}^{k\times k}$ valued variable

$$M_n = \frac{1}{n}\sum_{i=1}^n X[i] \otimes X[i].$$

Then,

$$\mathbb{E}\left[\left\|\sqrt{M_n} - \sqrt{M}\right\|^2\right] \leq \frac{\mathbb{E}\left[\left\|X[1] \otimes X[1] - M\right\|^2\right]}{n\lambda^+(M)},$$

where $\lambda^+(M) > 0$ denotes the smallest strictly positive eigenvalue of M.

Convergence in functional spaces

- As $k \to \infty$ we should obtain convergence e.g. in $\mathcal{C}^0(\mathcal{X})$ with $\mathcal{X} \subseteq \mathbb{R}^{n_0}$ compact. The problem is that $C = C(\mathcal{X})$ diverges as $k \to \infty$.
- The question for shallow networks has been addressed, but explicit rates for deeper networks are missing.
- We obtain an abstract bound

$$\mathcal{W}_2\left(f^{(L)}[\mathcal{X}], G^{(L)}[\mathcal{X}]\right) \leq \inf_{\varepsilon>0}\left\{C(\mathcal{X})\varepsilon^{\gamma} + C(\mathcal{K})\sqrt{n_L}\sum_{\ell=1}^{L-1}\frac{1}{\sqrt{n_\ell}}\right\},$$

where $\gamma \in (0,1)$ and $\mathcal{K} = \{x_i\}_{i=1}^{K}$ is such that

$$\sup_{y\in\mathcal{X}}\inf_{x\in K}\|x-y\| \leq \varepsilon.$$

Convergence in functional spaces

- As $k \to \infty$ we should obtain convergence e.g. in $\mathcal{C}^0(\mathcal{X})$ with $\mathcal{X} \subseteq \mathbb{R}^{n_0}$ compact. The problem is that $C = C(\mathcal{X})$ diverges as $k \to \infty$.

- The question for shallow networks has been addressed, but explicit rates for deeper networks are missing.

- We obtain an abstract bound

$$\mathcal{W}_2\left(f^{(L)}[\mathcal{X}], G^{(L)}[\mathcal{X}]\right) \le \inf_{\varepsilon>0}\left\{C(\mathcal{X})\varepsilon^{\gamma} + C(\mathcal{K})\sqrt{n_L}\sum_{\ell=1}^{L-1}\frac{1}{\sqrt{n_\ell}}\right\},$$

where $\gamma \in (0,1)$ and $\mathcal{K} = \{x_i\}_{i=1}^{K}$ is such that

$$\sup_{y\in\mathcal{X}}\inf_{x\in K}\|x-y\| \le \varepsilon.$$

Convergence in functional spaces

- As $k \to \infty$ we should obtain convergence e.g. in $\mathcal{C}^0(\mathcal{X})$ with $\mathcal{X} \subseteq \mathbb{R}^{n_0}$ compact. The problem is that $C = C(\mathcal{X})$ diverges as $k \to \infty$.

- The question for shallow networks has been addressed, but explicit rates for deeper networks are missing.

- We obtain an abstract bound

$$\mathcal{W}_2\left(f^{(L)}[\mathcal{X}], G^{(L)}[\mathcal{X}]\right) \leq \inf_{\varepsilon>0}\left\{C(\mathcal{X})\varepsilon^{\gamma} + C(\mathcal{K})\sqrt{n_L}\sum_{\ell=1}^{L-1}\frac{1}{\sqrt{n_\ell}}\right\},$$

where $\gamma \in (0,1)$ and $\mathcal{K} = \{x_i\}_{i=1}^{K}$ is such that

$$\sup_{y\in\mathcal{X}} \inf_{x\in K} \|x-y\| \leq \varepsilon.$$

Convergence in functional spaces

- As $k \to \infty$ we should obtain convergence e.g. in $\mathcal{C}^0(\mathcal{X})$ with $\mathcal{X} \subseteq \mathbb{R}^{n_0}$ compact. The problem is that $C = C(\mathcal{X})$ diverges as $k \to \infty$.
- The question for shallow networks has been addressed, but explicit rates for deeper networks are missing.
- We obtain an abstract bound

$$\mathcal{W}_2\left(f^{(L)}[\mathcal{X}], G^{(L)}[\mathcal{X}]\right) \leq \inf_{\varepsilon>0}\left\{C(\mathcal{X})\varepsilon^{\gamma} + C(\mathcal{K})\sqrt{n_L}\sum_{\ell=1}^{L-1}\frac{1}{\sqrt{n_\ell}}\right\},$$

where $\gamma \in (0,1)$ and $\mathcal{K} = \{x_i\}_{i=1}^K$ is such that

$$\sup_{y\in\mathcal{X}}\inf_{x\in K}\|x-y\| \leq \varepsilon.$$

Plan

1. Why random neural networks?
2. Our result
3. Numerical simulations
4. Extensions and future work

Numerical simulations

To explore the scope of our result, we fix the parameters $(n_\ell)_{\ell=1}^{L-1}$, compute $N \gg 1$ (pseudo)-samples of

1. Gaussian initialized fully connected neural networks $(f^{(L)}[\mathcal{X}]_i)_{i=1}^N$,
2. centred Gaussian vectors $(G^{(L)}[\mathcal{X}]_i)_{i=1}^N$ (with the prescribed covariance)

and compute the Wasserstein distance between the empirical measures (matching problem).

It is known that

$$\mathcal{W}_2\left(\frac{1}{N}\sum_{i=1}^N \delta_{f^{(L)}[\mathcal{X}]_i}, \frac{1}{N}\sum_{i=1}^N \delta_{G^{(L)}[\mathcal{X}]_i}\right) \approx \mathcal{W}_2\left(f^{(L)}[\mathcal{X}], G^{(L)}[\mathcal{X}]\right) + N^{-\alpha}.$$

with $\alpha = 1/(n_L|\mathcal{X}|)$ (if $n_L|\mathcal{X}| \geq 3$).

$\Rightarrow$ Simulations become less precise if $n_L|\mathcal{X}|$ is large (curse of dimensionality).

Numerical simulations

To explore the scope of our result, we fix the parameters $(n_\ell)_{\ell=1}^{L-1}$, compute $N \gg 1$ (pseudo)-samples of

1. Gaussian initialized fully connected neural networks $(f^{(L)}[\mathcal{X}]_i)_{i=1}^N$,
2. centred Gaussian vectors $(G^{(L)}[\mathcal{X}]_i)_{i=1}^N$ (with the prescribed covariance)

and compute the Wasserstein distance between the empirical measures (matching problem).

It is known that

$$\mathcal{W}_2\left(\frac{1}{N}\sum_{i=1}^N \delta_{f^{(L)}[\mathcal{X}]_i}, \frac{1}{N}\sum_{i=1}^N \delta_{G^{(L)}[\mathcal{X}]_i}\right) \approx \mathcal{W}_2\left(f^{(L)}[\mathcal{X}], G^{(L)}[\mathcal{X}]\right) + N^{-\alpha}.$$

with $\alpha = 1/(n_L|\mathcal{X}|)$ (if $n_L|\mathcal{X}| \geq 3$).

$\Rightarrow$ Simulations become less precise if $n_L|\mathcal{X}|$ is large (curse of dimensionality).

Numerical simulations

To explore the scope of our result, we fix the parameters $(n_\ell)_{\ell=1}^{L-1}$, compute $N \gg 1$ (pseudo)-samples of

1. Gaussian initialized fully connected neural networks $(f^{(L)}[\mathcal{X}]_i)_{i=1}^N$,
2. centred Gaussian vectors $(G^{(L)}[\mathcal{X}]_i)_{i=1}^N$ (with the prescribed covariance)

and compute the Wasserstein distance between the empirical measures (matching problem).

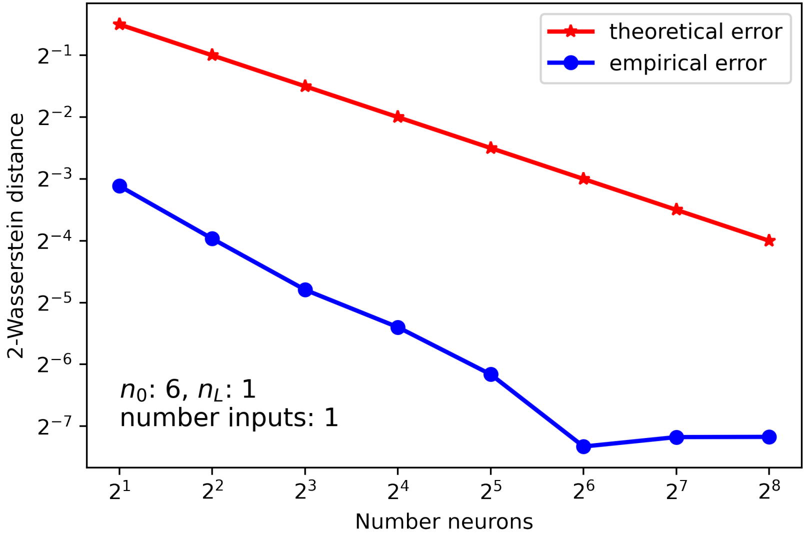
It is known that

$$\mathcal{W}_2\left(\frac{1}{N}\sum_{i=1}^N \delta_{f^{(L)}[\mathcal{X}]_i}, \frac{1}{N}\sum_{i=1}^N \delta_{G^{(L)}[\mathcal{X}]_i}\right) \approx \mathcal{W}_2\left(f^{(L)}[\mathcal{X}], G^{(L)}[\mathcal{X}]\right) + N^{-\alpha}.$$

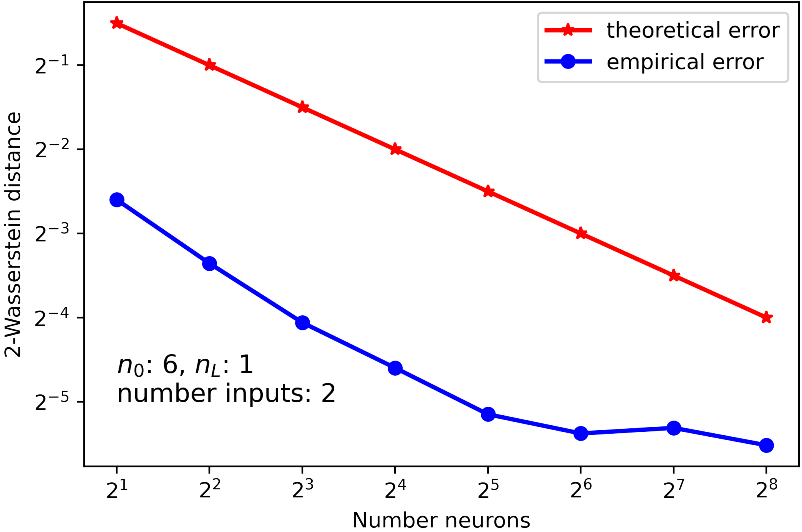
with $\alpha = 1/(n_L|\mathcal{X}|)$ (if $n_L|\mathcal{X}| \geq 3$).

$\Rightarrow$ Simulations become less precise if $n_L|\mathcal{X}|$ is large (curse of dimensionality).

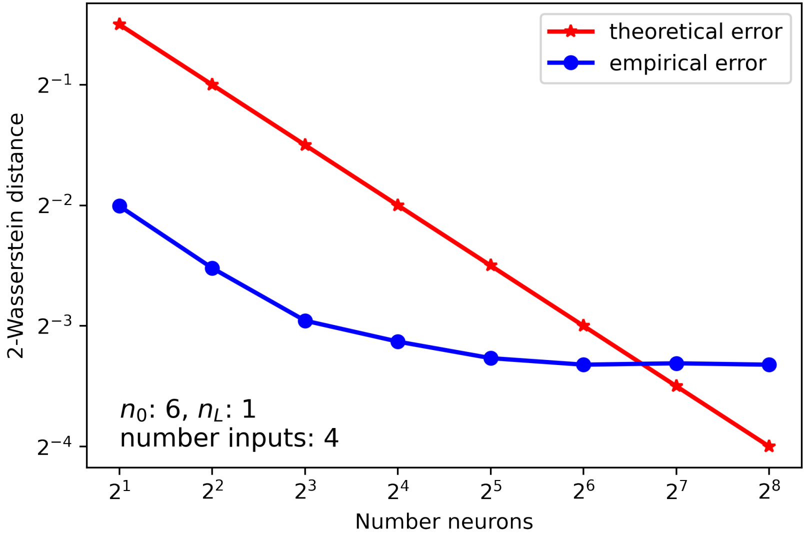
One input, $n_L = 1$



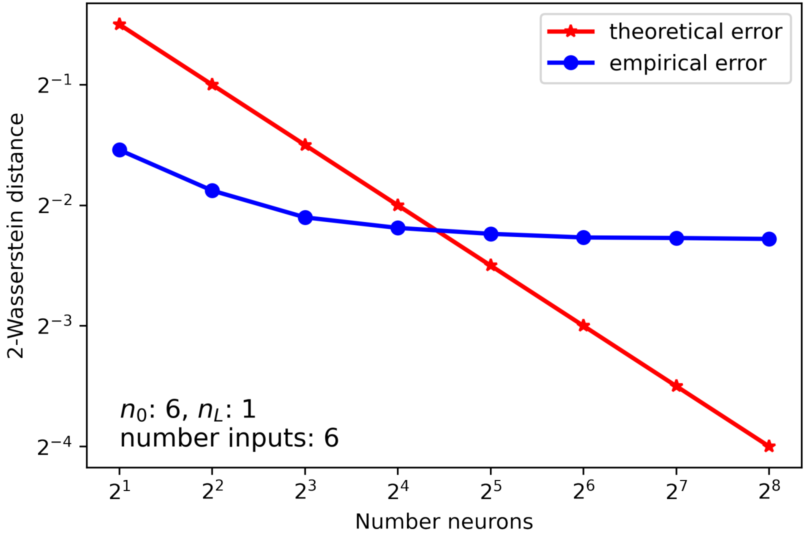
Enlarging the input set



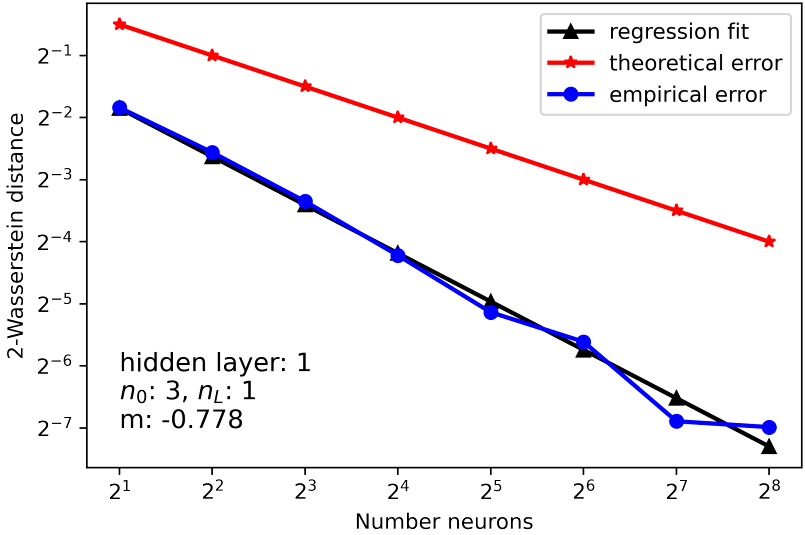
Enlarging the input set



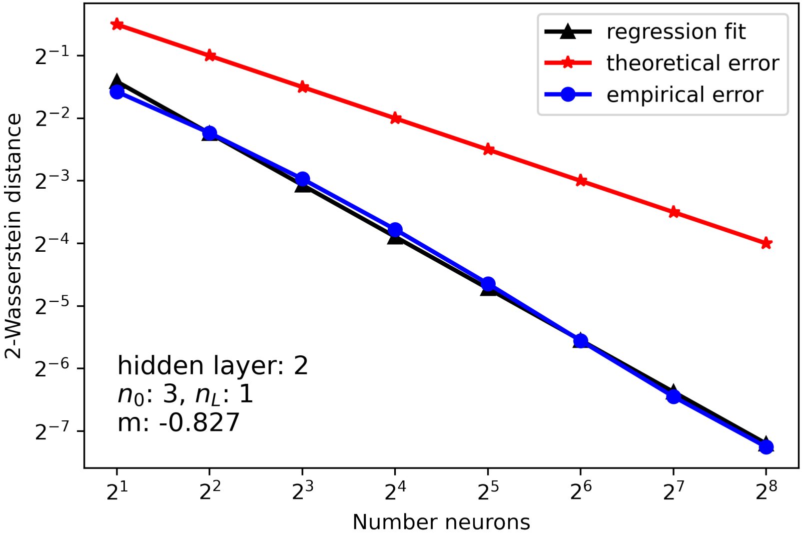
Enlarging the input set



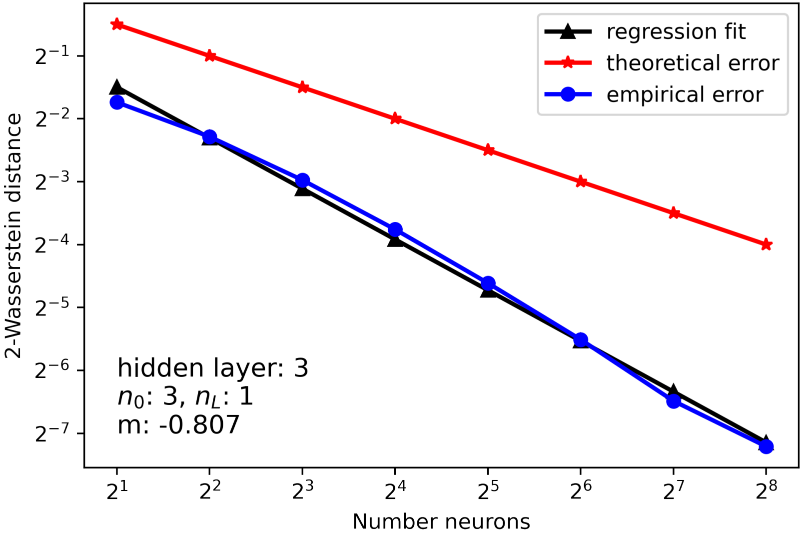
Deeper networks



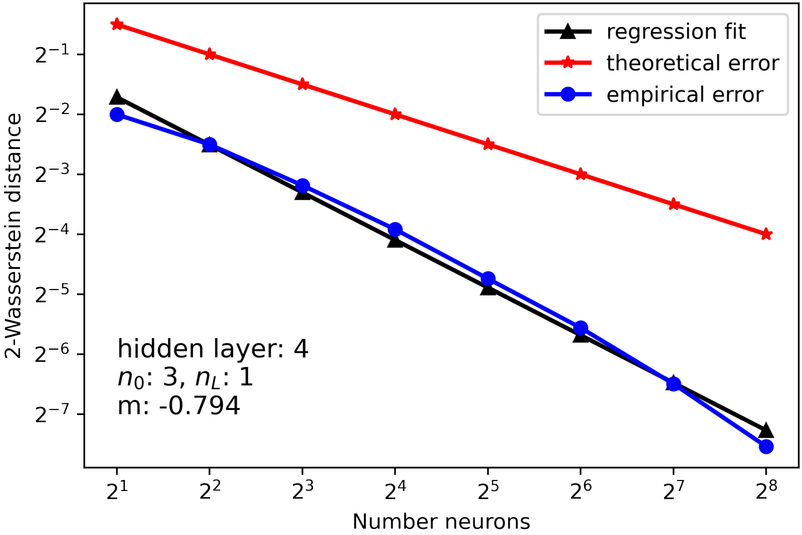
Deeper networks



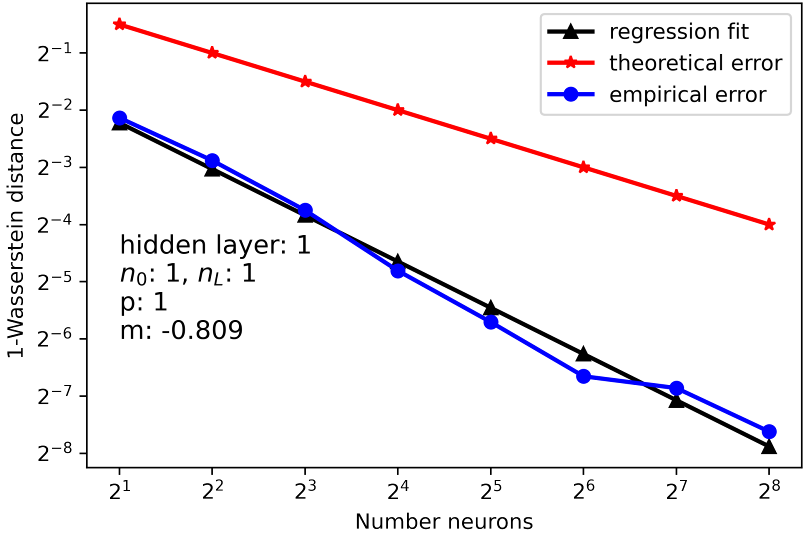
Deeper networks



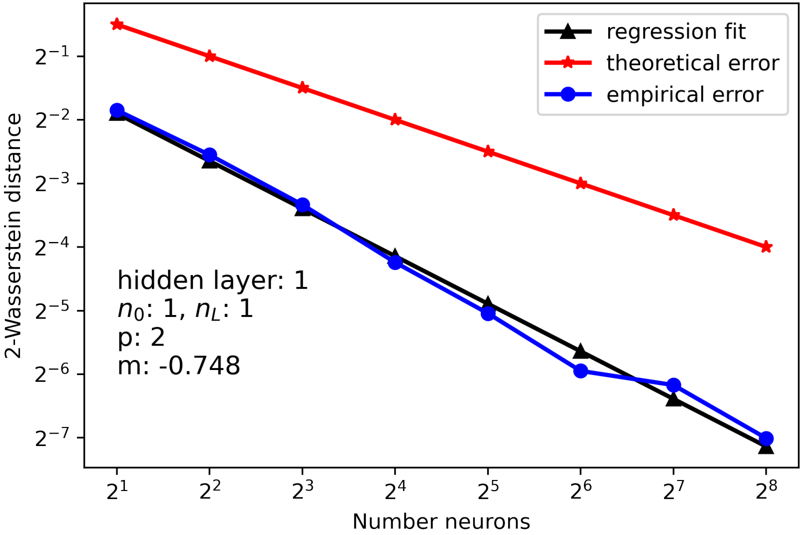
Deeper networks



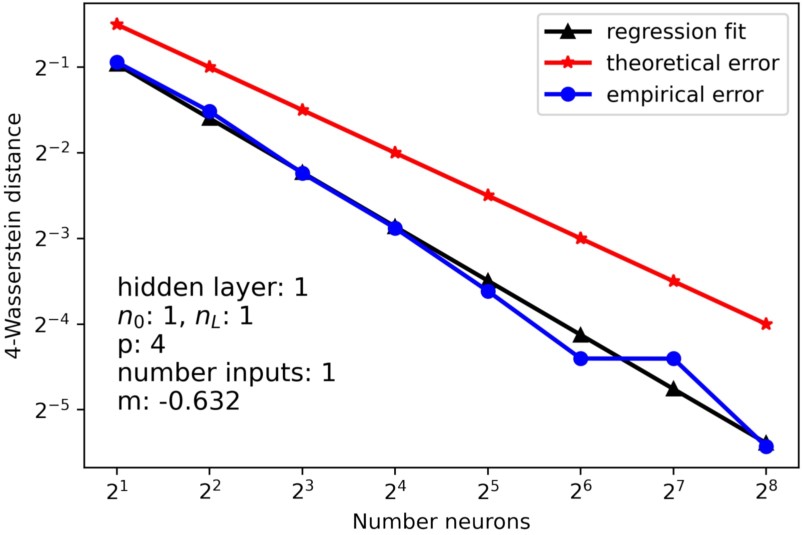
Distances of different order p



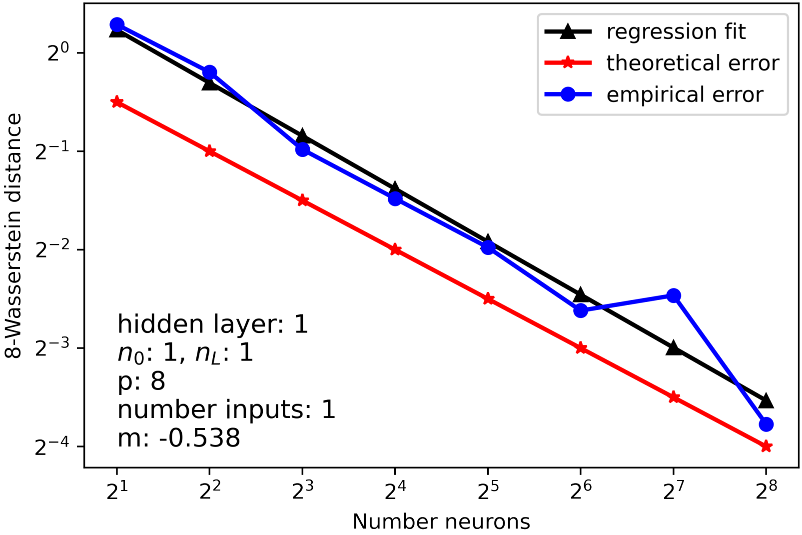
Distances of different order p



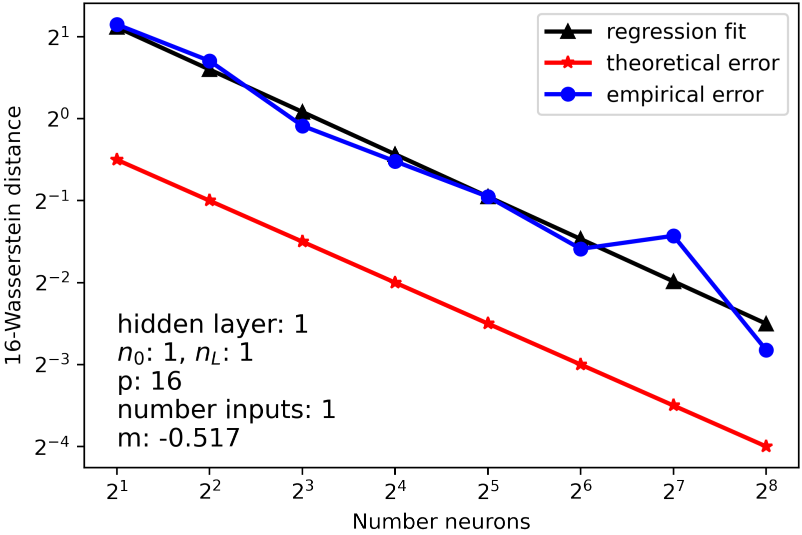
Distances of different order p



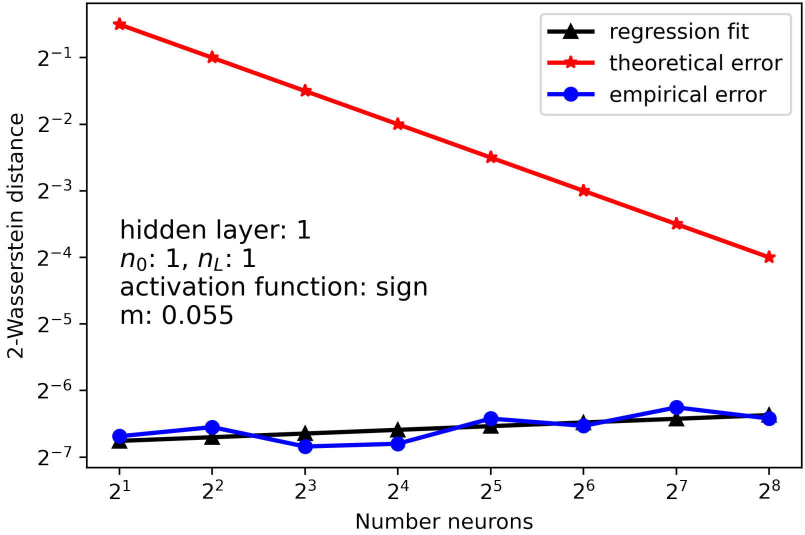
Distances of different order p



Distances of different order p



A non Lipschitz activation



Plan

1. Why random neural networks?
2. Our result
3. Numerical simulations
4. Extensions and future work

Possible extensions and open questions

We keep technicalities at minimum:

- $\mathcal{W}_2$ could be replaced with $\mathcal{W}_p$
- the proof should also extend from fully connected architectures to convolutional or recurrent ones
- one should allow for non-Gaussian laws for the parameters, such as discrete or even stable laws (where the Gaussian CLT fails)

Some interesting questions to address:

- Is the bound sharp (possibly allowing for discrete random parameters)?
- Properties of the optimal transport map (e.g. w.r.t. hidden layer sizes)
- What happens during/after training?

Possible extensions and open questions

We keep technicalities at minimum:

- $\mathcal{W}_2$ could be replaced with $\mathcal{W}_p$
- the proof should also extend from fully connected architectures to convolutional or recurrent ones
- one should allow for non-Gaussian laws for the parameters, such as discrete or even stable laws (where the Gaussian CLT fails)

Some interesting questions to address:

- Is the bound sharp (possibly allowing for discrete random parameters)?
- Properties of the optimal transport map (e.g. w.r.t. hidden layer sizes)
- What happens during/after training?

Possible extensions and open questions

We keep technicalities at minimum:

- $\mathcal{W}_2$ could be replaced with $\mathcal{W}_p$
- the proof should also extend from fully connected architectures to convolutional or recurrent ones
- one should allow for non-Gaussian laws for the parameters, such as discrete or even stable laws (where the Gaussian CLT fails)

Some interesting questions to address:

- Is the bound sharp (possibly allowing for discrete random parameters)?
- Properties of the optimal transport map (e.g. w.r.t. hidden layer sizes)
- What happens during/after training?

Supervised learning

In supervised learning (regression/classification) one has a training dataset

$$\mathcal{T} = \{(x_t, y_t)\} \subseteq \mathbb{R}^{n_0} \times \mathbb{R}^{n_L}$$

and a parametrized family of functions $(h(\cdot;\theta))_{\theta\in\Theta}$,

$$h(\cdot;\theta) : \mathbb{R}^{n_0} \to \mathbb{R}^{n_L} \quad x \mapsto h(\cdot;\theta).$$

Aim: find θ "fitting" the training dataset

$$h(x_t;\theta) \approx y_t$$

and also generalizing well to unseen data $x \mapsto h(x;\theta)$. Criteria:

- Full Bayesian: specify a prior distribution on $p(\theta)$ and compute the posterior:

$$p(\theta \,|\, h(x_t;\theta) \approx y_t \quad \forall (x_t, y_t) \in \mathcal{T})$$

- Variational/Decision: introduce a cost function e.g. $(h(x;\theta) - y)^2$ and minimize the *empirical risk* on the training set:

$$\theta^* \in \operatorname{argmin}_\theta \sum_{(x_t,y_t)\in\mathcal{T}} (h(x_t;\theta) - y_t)^2.$$

Supervised learning

In supervised learning (regression/classification) one has a training dataset

$$\mathcal{T} = \{(x_t, y_t)\} \subseteq \mathbb{R}^{n_0} \times \mathbb{R}^{n_L}$$

and a parametrized family of functions $(h(\cdot;\theta))_{\theta\in\Theta}$,

$$h(\cdot;\theta) : \mathbb{R}^{n_0} \to \mathbb{R}^{n_L} \quad x \mapsto h(\cdot;\theta).$$

Aim: find θ “fitting” the training dataset

$$h(x_t;\theta) \approx y_t$$

and also generalizing well to unseen data $x \mapsto h(x;\theta)$. Criteria:

- Full Bayesian: specify a prior distribution on $p(\theta)$ and compute the posterior:

$$p(\theta \,|\, h(x_t;\theta) \approx y_t \quad \forall (x_t, y_t) \in \mathcal{T})$$

- Variational/Decision: introduce a cost function e.g. $(h(x;\theta) - y)^2$ and minimize the *empirical risk* on the training set:

$$\theta^* \in \operatorname{argmin}_\theta \sum_{(x_t,y_t)\in\mathcal{T}} (h(x_t;\theta) - y_t)^2.$$

Supervised learning

In supervised learning (regression/classification) one has a training dataset

$$\mathcal{T} = \{(x_t, y_t)\} \subseteq \mathbb{R}^{n_0} \times \mathbb{R}^{n_L}$$

and a parametrized family of functions $(h(\cdot;\theta))_{\theta\in\Theta}$,

$$h(\cdot;\theta) : \mathbb{R}^{n_0} \to \mathbb{R}^{n_L} \quad x \mapsto h(\cdot;\theta).$$

Aim: find θ "fitting" the training dataset

$$h(x_t;\theta) \approx y_t$$

and also generalizing well to unseen data $x \mapsto h(x;\theta)$. Criteria:

- Full Bayesian: specify a prior distribution on $p(\theta)$ and compute the posterior:

$$p(\theta \,|\, h(x_t;\theta) \approx y_t \quad \forall (x_t, y_t) \in \mathcal{T})$$

- Variational/Decision: introduce a cost function e.g. $(h(x;\theta) - y)^2$ and minimize the *empirical risk* on the training set:

$$\theta^* \in \operatorname{argmin}_\theta \sum_{(x_t,y_t)\in\mathcal{T}} (h(x_t;\theta) - y_t)^2.$$

Supervised learning

In supervised learning (regression/classification) one has a training dataset

$$\mathcal{T} = \{(x_t, y_t)\} \subseteq \mathbb{R}^{n_0} \times \mathbb{R}^{n_L}$$

and a parametrized family of functions $(h(\cdot;\theta))_{\theta\in\Theta}$,

$$h(\cdot;\theta) : \mathbb{R}^{n_0} \to \mathbb{R}^{n_L} \quad x \mapsto h(\cdot;\theta).$$

Aim: find θ “fitting” the training dataset

$$h(x_t;\theta) \approx y_t$$

and also generalizing well to unseen data $x \mapsto h(x;\theta)$. Criteria:

- Full Bayesian: specify a prior distribution on $p(\theta)$ and compute the posterior:
$$p(\theta \,|\, h(x_t;\theta) \approx y_t \quad \forall (x_t, y_t) \in \mathcal{T})$$
- Variational/Decision: introduce a cost function e.g. $(h(x;\theta) - y)^2$ and minimize the *empirical risk* on the training set:
$$\theta^* \in \operatorname{argmin}_\theta \sum_{(x_t,y_t)\in\mathcal{T}} (h(x_t;\theta) - y_t)^2.$$

Bayesian posterior

- Deep networks $f^{(L)}$ are a parametrized family, $\theta = (\mathbf{W}, \mathbf{b})$.
- Random Gaussian initialized weights and biases specify a prior.
- Gaussian processes $G^{(L)}$ also provide a prior (where $\theta = G^{(L)}$ itself).

Conditioning upon $G^{(L)}(x_t) = y_t$ for every $(x_t, y_t) \in \mathcal{T}$ gives also a Gaussian posterior:

$$p(G^{(L)}(x) = \cdot \mid G^{(L)}(x_t) = y_t \quad \forall (x_t, y_t) \in \mathcal{T}) = \mathcal{N}(m(x), \sigma^2(x)),$$

with

$$m(x) = K^{(L)}(x, \mathcal{X}_{\mathcal{T}}) K^{(L)}(\mathcal{X}_{\mathcal{T}}, \mathcal{X}_{\mathcal{T}})^{-1} \mathcal{Y}_{\mathcal{T}}$$
$$\sigma^2(x) = K^{(L)}(x, x) - K^{(L)}(x, \mathcal{X}_{\mathcal{T}}) K^{(L)}(\mathcal{X}_{\mathcal{T}}, \mathcal{X}_{\mathcal{T}})^{-1} K^{(L)}(\mathcal{X}_{\mathcal{T}}, x)$$

Problem: Can we also approximate the neural network posterior?

Bayesian posterior

- Deep networks $f^{(L)}$ are a parametrized family, $\theta = (\mathbf{W}, \mathbf{b})$.
- Random Gaussian initialized weights and biases specify a prior.
- Gaussian processes $G^{(L)}$ also provide a prior (where $\theta = G^{(L)}$ itself).

Conditioning upon $G^{(L)}(x_t) = y_t$ for every $(x_t, y_t) \in \mathcal{T}$ gives also a Gaussian posterior:

$$p(G^{(L)}(x) = \cdot \mid G^{(L)}(x_t) = y_t \quad \forall (x_t, y_t) \in \mathcal{T}) = \mathcal{N}(m(x), \sigma^2(x)),$$

with

$$m(x) = K^{(L)}(x, \mathcal{X}_\mathcal{T}) K^{(L)}(\mathcal{X}_\mathcal{T}, \mathcal{X}_\mathcal{T})^{-1} \mathcal{Y}_\mathcal{T}$$
$$\sigma^2(x) = K^{(L)}(x, x) - K^{(L)}(x, \mathcal{X}_\mathcal{T}) K^{(L)}(\mathcal{X}_\mathcal{T}, \mathcal{X}_\mathcal{T})^{-1} K^{(L)}(\mathcal{X}_\mathcal{T}, x)$$

Problem: Can we also approximate the neural network posterior?

Bayesian posterior

- Deep networks $f^{(L)}$ are a parametrized family, $\theta = (\mathbf{W}, \mathbf{b})$.
- Random Gaussian initialized weights and biases specify a prior.
- Gaussian processes $G^{(L)}$ also provide a prior (where $\theta = G^{(L)}$ itself).

Conditioning upon $G^{(L)}(x_t) = y_t$ for every $(x_t, y_t) \in \mathcal{T}$ gives also a Gaussian posterior:

$$p(G^{(L)}(x) = \cdot \mid G^{(L)}(x_t) = y_t \quad \forall (x_t, y_t) \in \mathcal{T}) = \mathcal{N}(m(x), \sigma^2(x)),$$

with

$$m(x) = K^{(L)}(x, \mathcal{X}_\mathcal{T}) K^{(L)}(\mathcal{X}_\mathcal{T}, \mathcal{X}_\mathcal{T})^{-1} \mathcal{Y}_\mathcal{T}$$
$$\sigma^2(x) = K^{(L)}(x, x) - K^{(L)}(x, \mathcal{X}_\mathcal{T}) K^{(L)}(\mathcal{X}_\mathcal{T}, \mathcal{X}_\mathcal{T})^{-1} K^{(L)}(\mathcal{X}_\mathcal{T}, x)$$

Problem: Can we also approximate the neural network posterior?

Bayesian posterior

- Deep networks $f^{(L)}$ are a parametrized family, $\theta = (\mathbf{W}, \mathbf{b})$.
- Random Gaussian initialized weights and biases specify a prior.
- Gaussian processes $G^{(L)}$ also provide a prior (where $\theta = G^{(L)}$ itself).

Conditioning upon $G^{(L)}(x_t) = y_t$ for every $(x_t, y_t) \in \mathcal{T}$ gives also a Gaussian posterior:

$$p(G^{(L)}(x) = \cdot \mid G^{(L)}(x_t) = y_t \quad \forall (x_t, y_t) \in \mathcal{T}) = \mathcal{N}(m(x), \sigma^2(x)),$$

with

$$m(x) = K^{(L)}(x, \mathcal{X}_\mathcal{T}) K^{(L)}(\mathcal{X}_\mathcal{T}, \mathcal{X}_\mathcal{T})^{-1} \mathcal{Y}_\mathcal{T}$$
$$\sigma^2(x) = K^{(L)}(x, x) - K^{(L)}(x, \mathcal{X}_\mathcal{T}) K^{(L)}(\mathcal{X}_\mathcal{T}, \mathcal{X}_\mathcal{T})^{-1} K^{(L)}(\mathcal{X}_\mathcal{T}, x)$$

Problem: Can we also approximate the neural network posterior?

Bayesian posterior

- Deep networks $f^{(L)}$ are a parametrized family, $\theta = (\mathbf{W}, \mathbf{b})$.
- Random Gaussian initialized weights and biases specify a prior.
- Gaussian processes $G^{(L)}$ also provide a prior (where $\theta = G^{(L)}$ itself).

Conditioning upon $G^{(L)}(x_t) = y_t$ for every $(x_t, y_t) \in \mathcal{T}$ gives also a Gaussian posterior:

$$p(G^{(L)}(x) = \cdot \mid G^{(L)}(x_t) = y_t \quad \forall (x_t, y_t) \in \mathcal{T}) = \mathcal{N}(m(x), \sigma^2(x)),$$

with

$$\begin{aligned} m(x) &= K^{(L)}(x, \mathcal{X}_\mathcal{T}) K^{(L)}(\mathcal{X}_\mathcal{T}, \mathcal{X}_\mathcal{T})^{-1} \mathcal{Y}_\mathcal{T} \\ \sigma^2(x) &= K^{(L)}(x, x) - K^{(L)}(x, \mathcal{X}_\mathcal{T}) K^{(L)}(\mathcal{X}_\mathcal{T}, \mathcal{X}_\mathcal{T})^{-1} K^{(L)}(\mathcal{X}_\mathcal{T}, x) \end{aligned}$$

Problem: Can we also approximate the neural network posterior?

Bayesian posterior

- Deep networks $f^{(L)}$ are a parametrized family, $\theta = (\mathbf{W}, \mathbf{b})$.
- Random Gaussian initialized weights and biases specify a prior.
- Gaussian processes $G^{(L)}$ also provide a prior (where $\theta = G^{(L)}$ itself).

Conditioning upon $G^{(L)}(x_t) = y_t$ for every $(x_t, y_t) \in \mathcal{T}$ gives also a Gaussian posterior:

$$p(G^{(L)}(x) = \cdot \,|\, G^{(L)}(x_t) = y_t \quad \forall (x_t, y_t) \in \mathcal{T}) = \mathcal{N}(m(x), \sigma^2(x)),$$

with

$$\begin{aligned} m(x) &= K^{(L)}(x, \mathcal{X}_{\mathcal{T}}) K^{(L)}(\mathcal{X}_{\mathcal{T}}, \mathcal{X}_{\mathcal{T}})^{-1} \mathcal{Y}_{\mathcal{T}} \\ \sigma^2(x) &= K^{(L)}(x, x) - K^{(L)}(x, \mathcal{X}_{\mathcal{T}}) K^{(L)}(\mathcal{X}_{\mathcal{T}}, \mathcal{X}_{\mathcal{T}})^{-1} K^{(L)}(\mathcal{X}_{\mathcal{T}}, x) \end{aligned}$$

Problem: Can we also approximate the neural network posterior?

Neural Tangent Kernel

Minimization of the empirical risk

$$\theta \mapsto \sum_{(x_t, y_t) \in \mathcal{T}} (h(x_t; \theta) - y_t)^2$$

is usually via (stochastic) gradient descent algorithms (training).

Problem: for $h(\cdot; \theta) = f^{(L)}(\cdot)$ the functional is not convex (local minima, vanishing gradient, ...)

A solution: in the wide limit $\min_{\ell=1,\ldots,L-1} n_\ell \to \infty$ the training $t \mapsto \theta_t$ is (at first order) given by an ODE driven by the gradient of the cost and a (constant) Neural Tangent Kernel $NTK^{(L)}(x, y)$ – explicit and recursively computable:

$$NTK^{(L)}(x, y) = \lim_{\min_{\ell=1,\ldots,L-1} n_\ell \to \infty} \nabla_\theta f^{(L)}(x) \cdot \nabla_\theta f^{(L)}(y).$$

Link with Malliavin calculus?

Neural Tangent Kernel

Minimization of the empirical risk

$$\theta \mapsto \sum_{(x_t, y_t) \in \mathcal{T}} (h(x_t; \theta) - y_t)^2$$

is usually via (stochastic) gradient descent algorithms (training).

Problem: for $h(\cdot; \theta) = f^{(L)}(\cdot)$ the functional is not convex (local minima, vanishing gradient, ...)

A solution: in the wide limit $\min_{\ell=1,\dots,L-1} n_\ell \to \infty$ the training $t \mapsto \theta_t$ is (at first order) given by an ODE driven by the gradient of the cost and a (constant) Neural Tangent Kernel $NTK^{(L)}(x, y)$ – explicit and recursively computable:

$$NTK^{(L)}(x, y) = \lim_{\min_{\ell=1,\dots,L-1} n_\ell \to \infty} \nabla_\theta f^{(L)}(x) \cdot \nabla_\theta f^{(L)}(y).$$

Link with Malliavin calculus?

Neural Tangent Kernel

Minimization of the empirical risk

$$\theta \mapsto \sum_{(x_t, y_t) \in \mathcal{T}} (h(x_t; \theta) - y_t)^2$$

is usually via (stochastic) gradient descent algorithms (training).

Problem: for $h(\cdot; \theta) = f^{(L)}(\cdot)$ the functional is not convex (local minima, vanishing gradient, ...)

A solution: in the wide limit $\min_{\ell=1,\ldots,L-1} n_\ell \to \infty$ the training $t \mapsto \theta_t$ is (at first order) given by an ODE driven by the gradient of the cost and a (constant) Neural Tangent Kernel $NTK^{(L)}(x, y)$ – explicit and recursively computable:

$$NTK^{(L)}(x, y) = \lim_{\min_{\ell=1,\ldots,L-1} n_\ell \to \infty} \nabla_\theta f^{(L)}(x) \cdot \nabla_\theta f^{(L)}(y).$$

Link with Malliavin calculus?

Neural Tangent Kernel

Minimization of the empirical risk

$$\theta \mapsto \sum_{(x_t, y_t) \in \mathcal{T}} (h(x_t; \theta) - y_t)^2$$

is usually via (stochastic) gradient descent algorithms (training).

Problem: for $h(\cdot; \theta) = f^{(L)}(\cdot)$ the functional is not convex (local minima, vanishing gradient, ...)

A solution: in the wide limit $\min_{\ell=1,\ldots,L-1} n_\ell \to \infty$ the training $t \mapsto \theta_t$ is (at first order) given by an ODE driven by the gradient of the cost and a (constant) Neural Tangent Kernel $NTK^{(L)}(x, y)$ – explicit and recursively computable:

$$NTK^{(L)}(x, y) = \lim_{\min_{\ell=1,\ldots,L-1} n_\ell \to \infty} \nabla_\theta f^{(L)}(x) \cdot \nabla_\theta f^{(L)}(y).$$

Link with Malliavin calculus?

Basteri, A., Trevisan, D.
Quantitative Gaussian Approximation of Randomly Initialized Deep Neural Networks
arXiv preprint arXiv:2203.07379 (2022).

Lee, Jaehoon, et al.
Wide neural networks of any depth evolve as linear models under gradient descent
Advances in neural information processing systems 32 (2019).

Williams, Rasmussen.
Gaussian processes for machine learning
Vol. 2. No. 3. Cambridge, MA: MIT press, 2006.

Roberts, D. A., Yaida S., and Hanin B.
The principles of deep learning theory
arXiv preprint arXiv:2106.10165 (2021).