Gaussian Processes as Approximations of Random Neural Networks

Dario Trevisan
(based on joint work with A. Basteri arXiv:2203.07379)

Università di Pisa
dario.trevisan@unipi.it

Analysis and Applied Mathematics Department Seminars
Università Bocconi, Milano
04 Oct 2023

Plan

Plan

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 (NN's) are biologically-inspired parametrized 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 (NN's) are biologically-inspired parametrized 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 (NN's) are biologically-inspired parametrized functions

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

as stacked compositions of

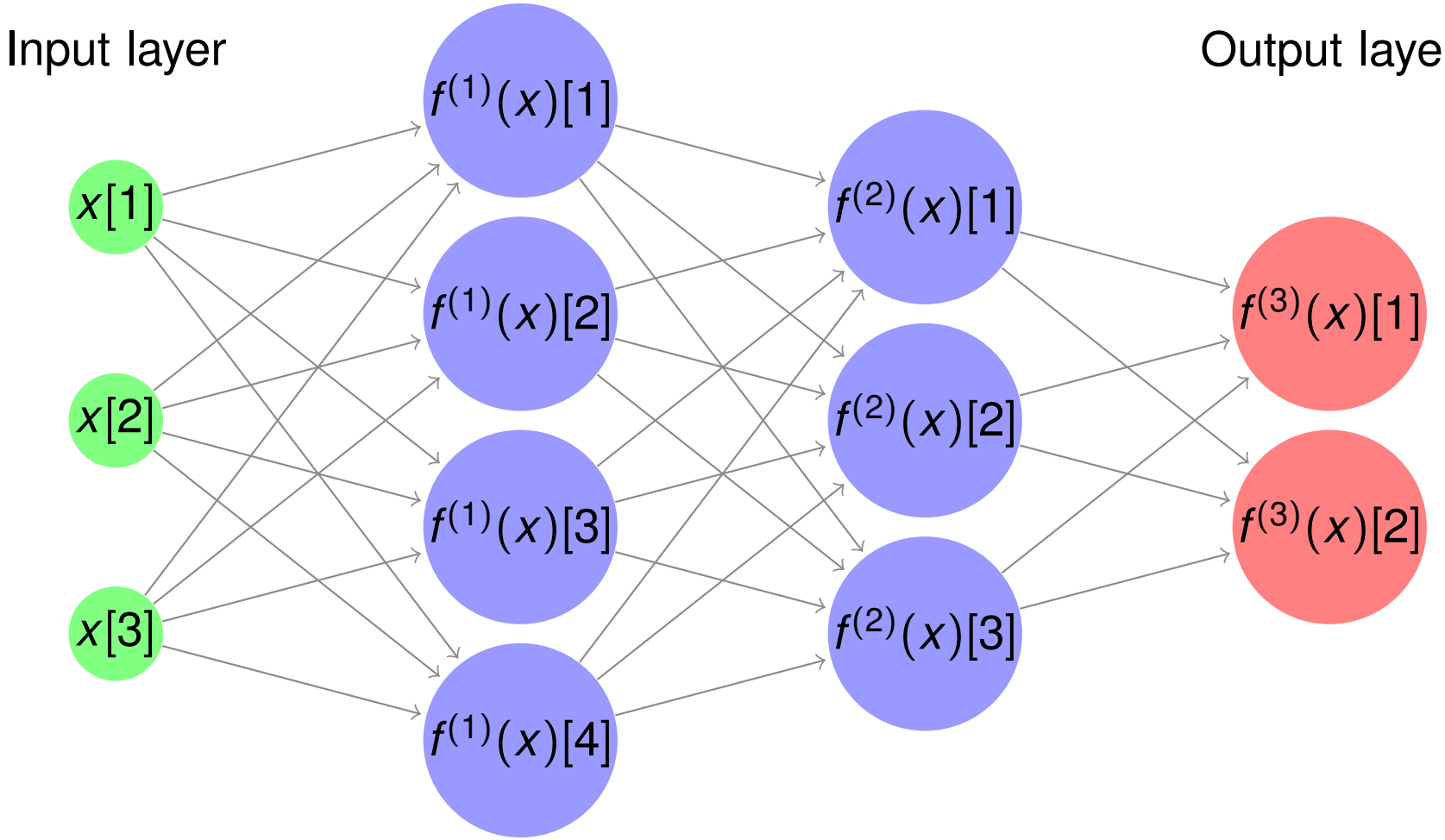
- linear (or affine) maps
- non-linear functions (usually acting componentwise).

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 that are randomly sampled.

Several reasons:

- Bayesian approach: prior distribution over weights and biases
- Training algorithms (SGD) use random initialization.
- Training only a fraction of the parameters (e.g.the last layer) (random features, reservoir computing).

Literature: Rosenblatt (1958), Neal (1996), Matthews (2018), Lee (2019)...

Random neural networks

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

Several reasons:

- Bayesian approach: prior distribution over weights and biases
- Training algorithms (SGD) use random initialization.
- Training only a fraction of the parameters (e.g.the last layer) (random features, reservoir computing).

Literature: Rosenblatt (1958), Neal (1996), Matthews (2018), Lee (2019)...

Random neural networks

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

Several reasons:

- Bayesian approach: prior distribution over weights and biases
- Training algorithms (SGD) use random initialization.
- Training only a fraction of the parameters (e.g.the last layer) (random features, reservoir computing).

Literature: Rosenblatt (1958), Neal (1996), Matthews (2018), Lee (2019)...

Random neural networks

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

Several reasons:

- Bayesian approach: prior distribution over weights and biases
- Training algorithms (SGD) use random initialization.
- Training only a fraction of the parameters (e.g.the last layer) (random features, reservoir computing).

Literature: Rosenblatt (1958), Neal (1996), Matthews (2018), Lee (2019)...

Random neural networks

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

Several reasons:

- Bayesian approach: prior distribution over weights and biases
- Training algorithms (SGD) use random initialization.
- Training only a fraction of the parameters (e.g.the last layer) (random features, reservoir computing).

Literature: Rosenblatt (1958), Neal (1996), Matthews (2018), Lee (2019)...

Random neural networks

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

Several reasons:

- Bayesian approach: prior distribution over weights and biases
- Training algorithms (SGD) use random initialization.
- Training only a fraction of the parameters (e.g.the last layer) (random features, reservoir computing).

Literature: Rosenblatt (1958), Neal (1996), Matthews (2018), Lee (2019)...

Plan

Our main result in brief

1. We provide quantitative bounds for the Gaussian approximation of deep fully connected NN's with random parameters (at initialization).
2. We provide for the first time explicit rates for the convergence of deep networks in the wide limit.
3. The key tool we employ is the Wasserstein distance (of order 2).

Our main result in brief

1. We provide quantitative bounds for the Gaussian approximation of deep fully connected NN's with random parameters (at initialization).

2. We provide for the first time explicit rates for the convergence of deep networks in the wide limit.

3. The key tool we employ is the Wasserstein distance (of order 2).

Our main result in brief

1. We provide quantitative bounds for the Gaussian approximation of deep fully connected NN's with random parameters (at initialization).

2. We provide for the first time explicit rates for the convergence of deep networks in the wide limit.

3. The key tool we employ is the Wasserstein distance (of order 2).

Our main result in brief

1. We provide quantitative bounds for the Gaussian approximation of deep fully connected NN's with random parameters (at initialization).

2. We provide for the first time explicit rates for the convergence of deep networks in the wide limit.

3. The key tool we employ is the Wasserstein distance (of order 2).

Notation: neural networks

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

- 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

- 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

- 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

- 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

- 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

- 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:

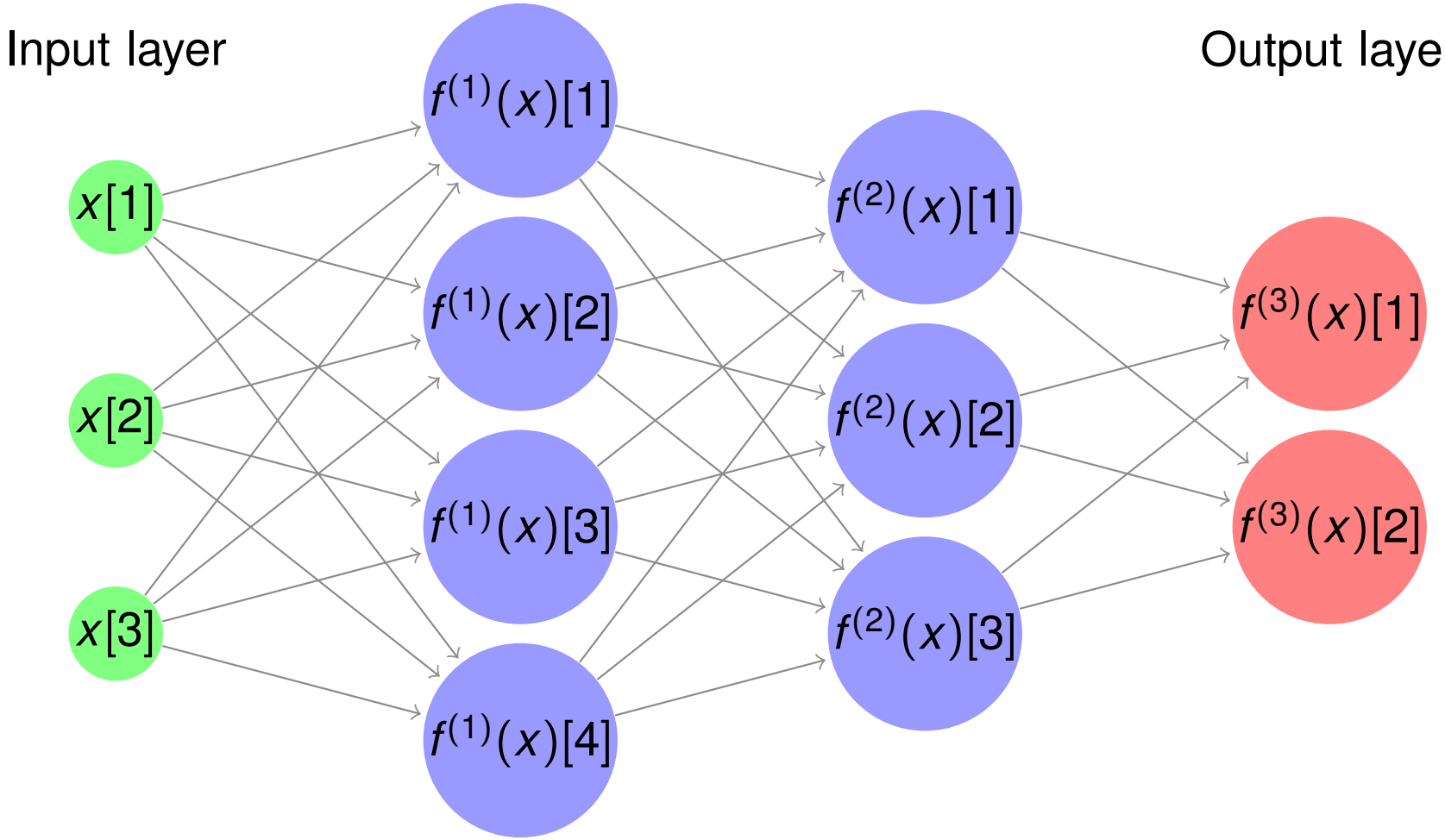
$$f^{(1)} : \mathbb{R}^{n_0} \to \mathbb{R}^{n_1}, \quad f^{(1)}(x) = W^{(0)}x + b^{(0)},$$

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.

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



Notation: Wasserstein distance of order 2

Given probabilities μ, ν on $\mathbb{R}^d$, define

$$\mathcal{W}_2(\mu,\nu) = \inf\left\{ \sqrt{\mathbb{E}\left[\|X-Y\|^2\right]} \ : X,\ Y \text{ r.v.'s with } p(X=\cdot)=\mu,\ p(Y=\cdot)=\nu \right\}.$$

- Slight abuse of notation:

$$\mathcal{W}_2(X,Y) = \mathcal{W}_2(p(X=\cdot), p(Y=\cdot))$$

- 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

$$\lim_{n\to\infty} X_n = X \ \text{ in law} \quad \text{and} \quad \lim_{n\to\infty} \mathbb{E}\left[\|X_n\|^2\right] = \mathbb{E}\left[\|X\|^2\right].$$

Notation: Wasserstein distance of order 2

Given probabilities μ, ν on $\mathbb{R}^d$, define

$$\mathcal{W}_2(\mu,\nu) = \inf\left\{\sqrt{\mathbb{E}\left[\|X-Y\|^2\right]} \,:\, X,\ Y \text{ r.v.'s with } p(X=\cdot)=\mu,\ p(Y=\cdot)=\nu\right\}.$$

- Slight abuse of notation:

$$\mathcal{W}_2(X,Y) = \mathcal{W}_2(p(X=\cdot), p(Y=\cdot))$$

- 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

$$\lim_{n\to\infty} X_n = X \text{ in law} \quad \text{and} \quad \lim_{n\to\infty} \mathbb{E}\left[\|X_n\|^2\right] = \mathbb{E}\left[\|X\|^2\right].$$

Notation: Wasserstein distance of order 2

Given probabilities μ, ν on $\mathbb{R}^d$, define

$$\mathcal{W}_2(\mu,\nu) = \inf\left\{\sqrt{\mathbb{E}\left[\|X-Y\|^2\right]} \,:\, X,\ Y \text{ r.v.'s with } p(X=\cdot)=\mu,\ p(Y=\cdot)=\nu\right\}.$$

- Slight abuse of notation:

$$\mathcal{W}_2(X,Y) = \mathcal{W}_2(p(X=\cdot), p(Y=\cdot))$$

- 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

$$\lim_{n\to\infty} X_n = X \text{ in law} \quad \text{and} \quad \lim_{n\to\infty} \mathbb{E}\left[\|X_n\|^2\right] = \mathbb{E}\left[\|X\|^2\right].$$

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

$$f^{(L)}[\mathcal{X}] = (f^{(L)}(x_i))_{i=1}^k \in \mathbb{R}^{n_L \times k}$$

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

$$f^{(L)}[\mathcal{X}] = (f^{(L)}(x_i))_{i=1}^k \in \mathbb{R}^{n_L \times k}$$

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

$$f^{(L)}[\mathcal{X}] = (f^{(L)}(x_i))_{i=1}^k \in \mathbb{R}^{n_L \times k}$$

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 $(n_\ell)_{\ell=1}^L$.

Remarks

- 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 covariance of $G^{(L)}[\mathcal{X}] \in \mathbb{R}^{n_L \times |\mathcal{X}|}$ (aka NNGP) is explicit and depends on σ, the input $\mathcal{X}$ and the output dimension n_L (not on the hidden layer sizes).

- The rows of $G^{(L)}[\mathcal{X}]$ are i.i.d.

- 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.

Idea of proof

The Gaussian limit is due to a combination, in each layer, of:

- Central Limit Theorem scaling for the weights
- almost independence of the neurons.

We argue by induction over the layers:

- For one hidden layer exact independence $\to$ straightforward CLT.
- Triangle inequality for $\mathcal{W}_2$ and the inductive assumption $\to$ the Gaussian approximation yields exact independence.
- Bound error terms using convexity of the squared $\mathcal{W}_2$ and the explicit optimal transport between Gaussians.

Idea of proof

The Gaussian limit is due to a combination, in each layer, of:

- Central Limit Theorem scaling for the weights
- almost independence of the neurons.

We argue by induction over the layers:

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

Plan

Numerical simulations

To validate our result, fix the parameters $(n_\ell)_{\ell=1}^{L-1}$, sample $N \gg 1$

1. Gaussian randomly initialized NN's $(f^{(L)}[\mathcal{X}]_i)_{i=1}^N$,
2. Gaussian variables $(G^{(L)}[\mathcal{X}]_i)_{i=1}^N$

and compute the Wasserstein distance between the empirical measures.

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 $\alpha \geq 1/(n_L|\mathcal{X}|)$.

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

Numerical simulations

To validate our result, fix the parameters $(n_\ell)_{\ell=1}^{L-1}$, sample $N \gg 1$

1. Gaussian randomly initialized NN's $(f^{(L)}[\mathcal{X}]_i)_{i=1}^N$,
2. Gaussian variables $(G^{(L)}[\mathcal{X}]_i)_{i=1}^N$

and compute the Wasserstein distance between the empirical measures.

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 $\alpha \geq 1/(n_L|\mathcal{X}|)$.

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

Numerical simulations

To validate our result, fix the parameters $(n_\ell)_{\ell=1}^{L-1}$, sample $N \gg 1$

1. Gaussian randomly initialized NN's $(f^{(L)}[\mathcal{X}]_i)_{i=1}^N$,
2. Gaussian variables $(G^{(L)}[\mathcal{X}]_i)_{i=1}^N$

and compute the Wasserstein distance between the empirical measures.

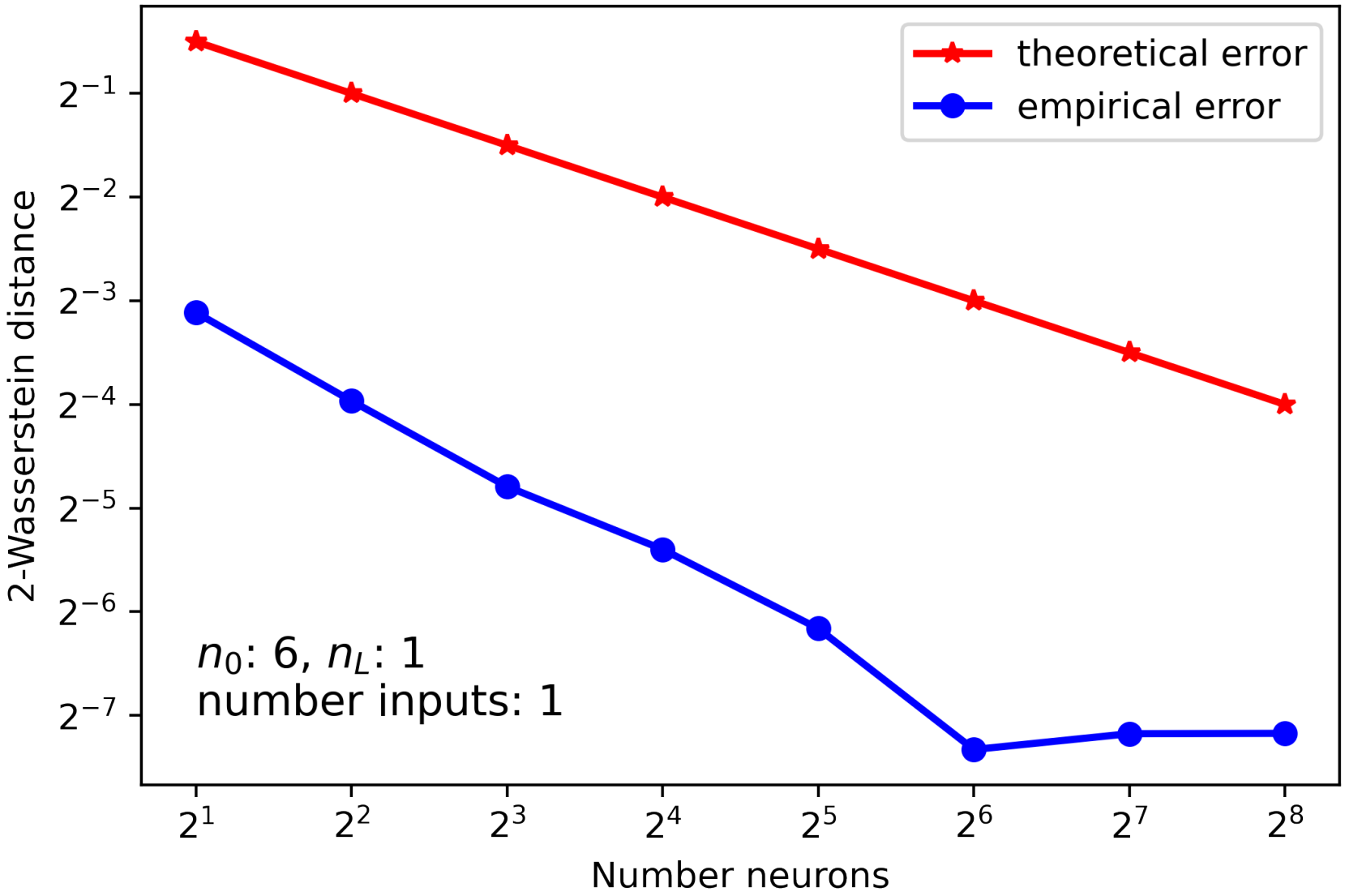
It is known that

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

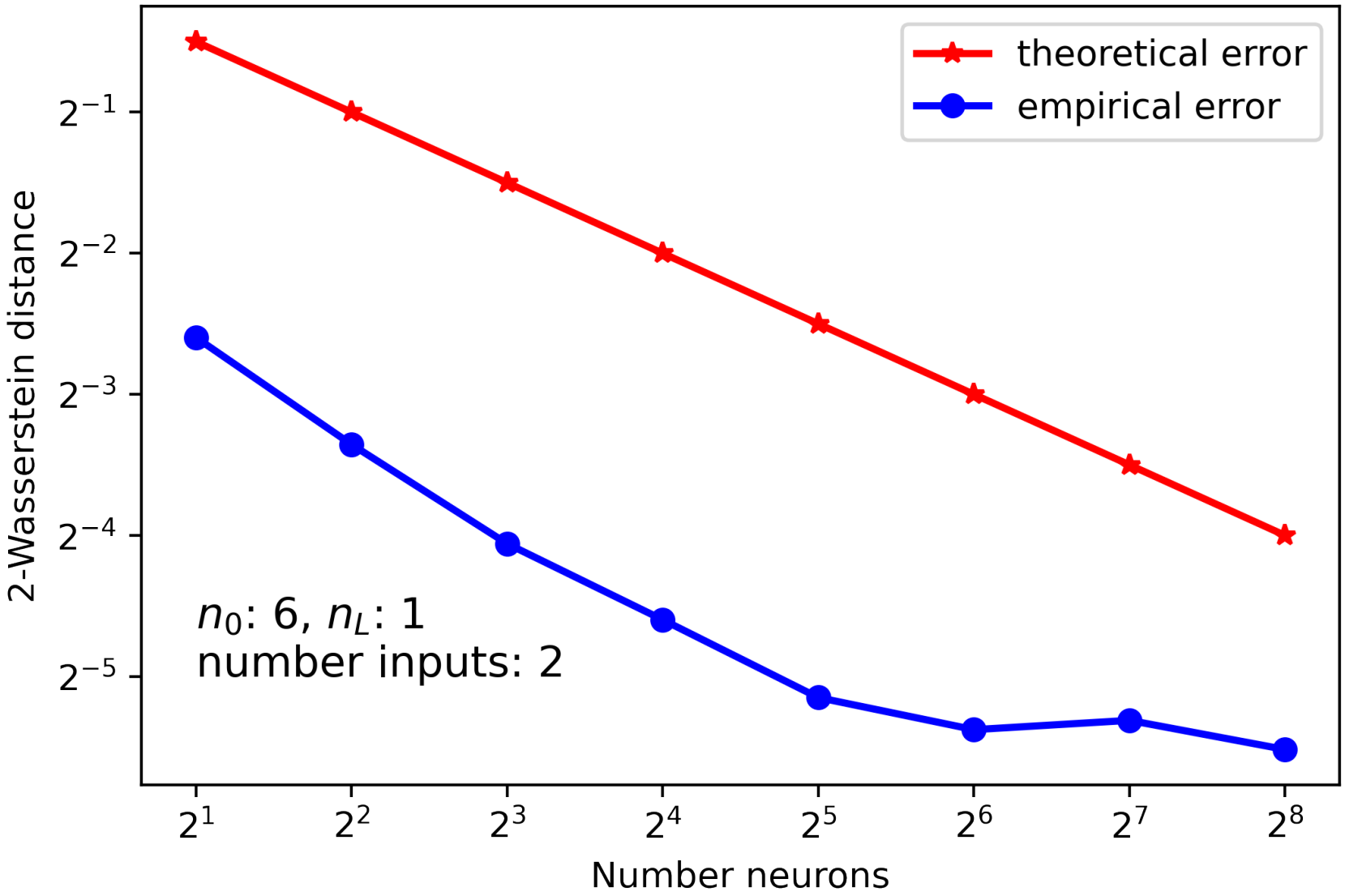
with $\alpha \geq 1/(n_L|\mathcal{X}|)$.

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

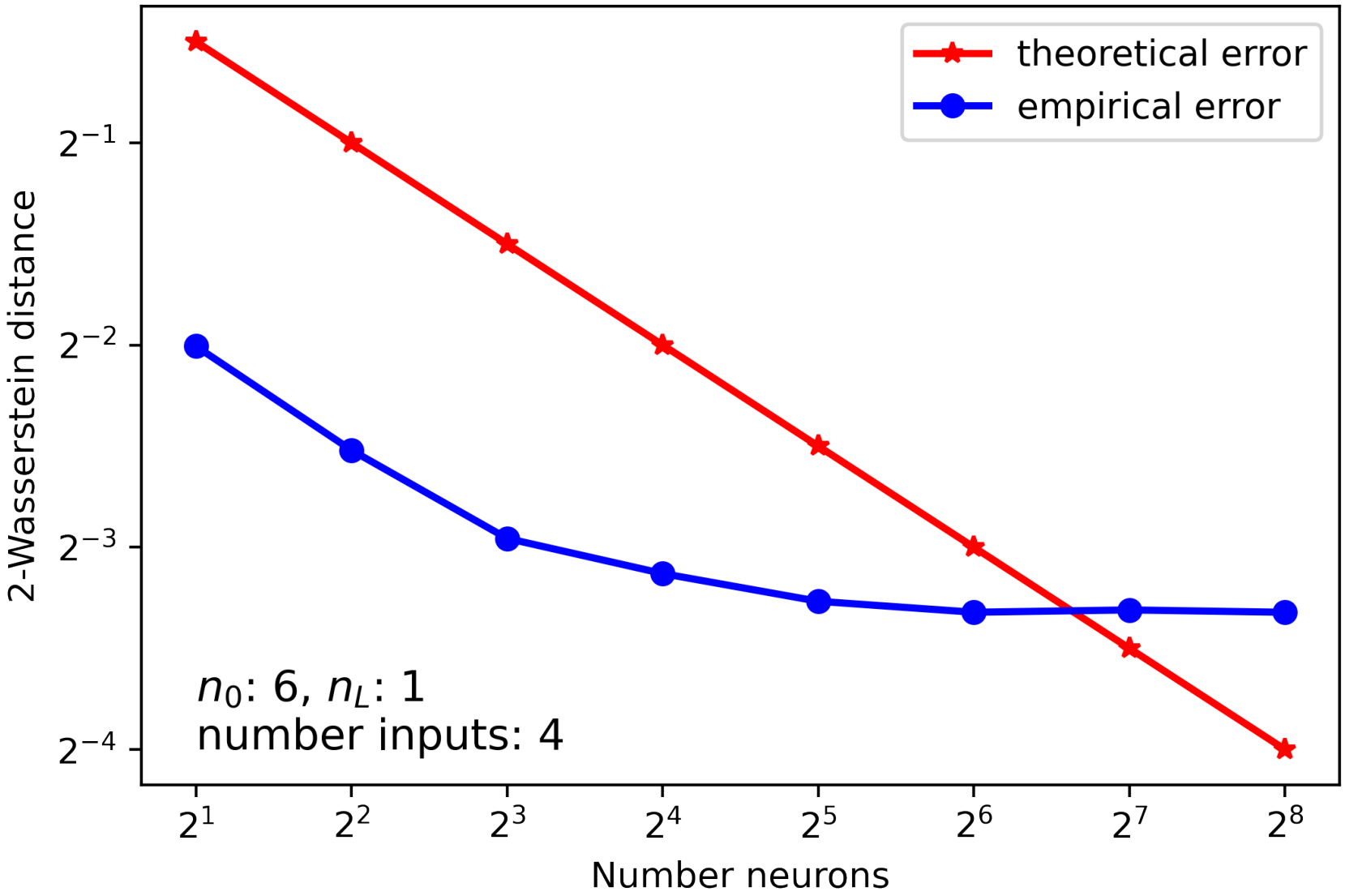
One input, $n_L = 1$



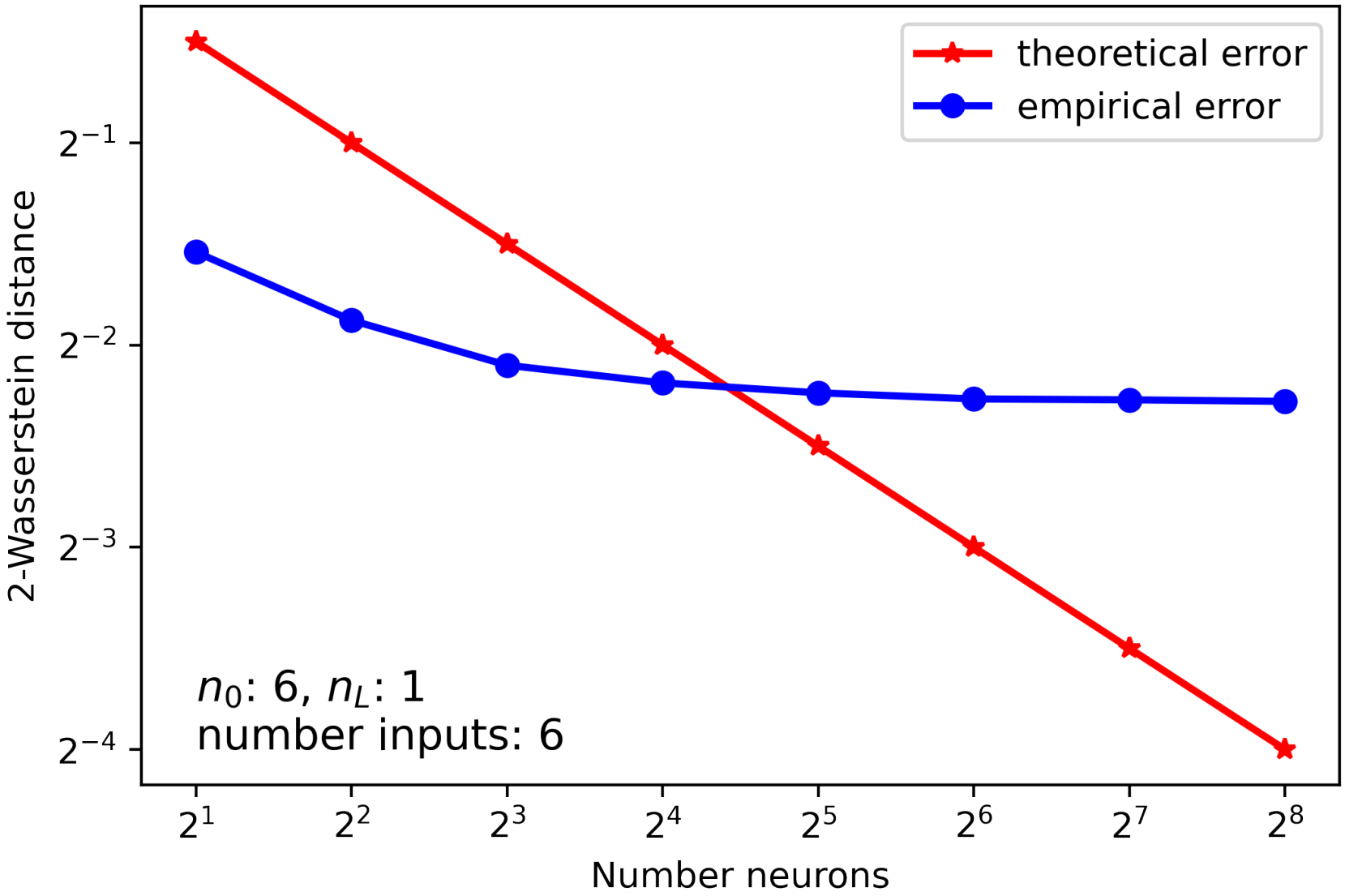
Enlarging the input set



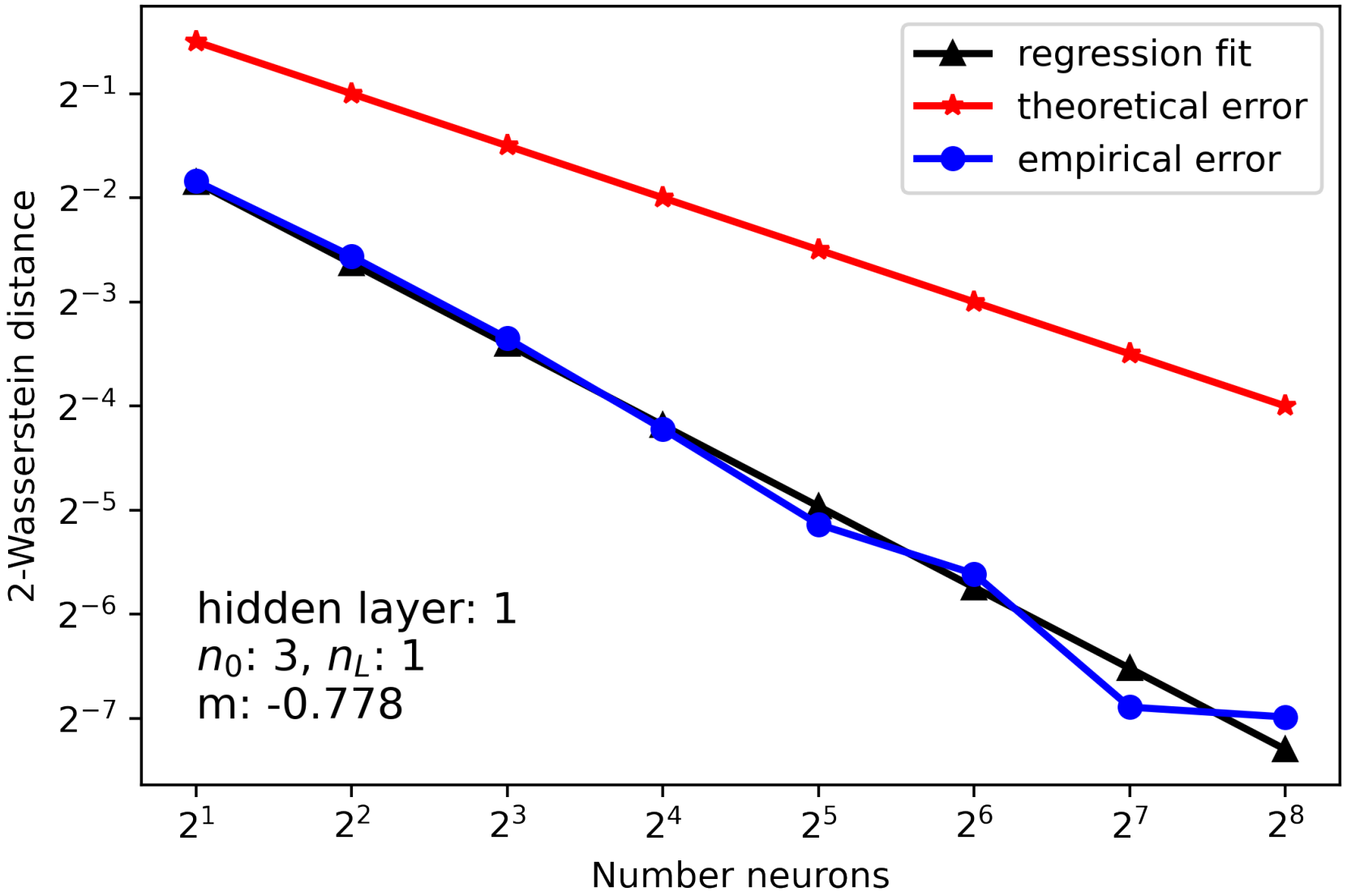
Enlarging the input set



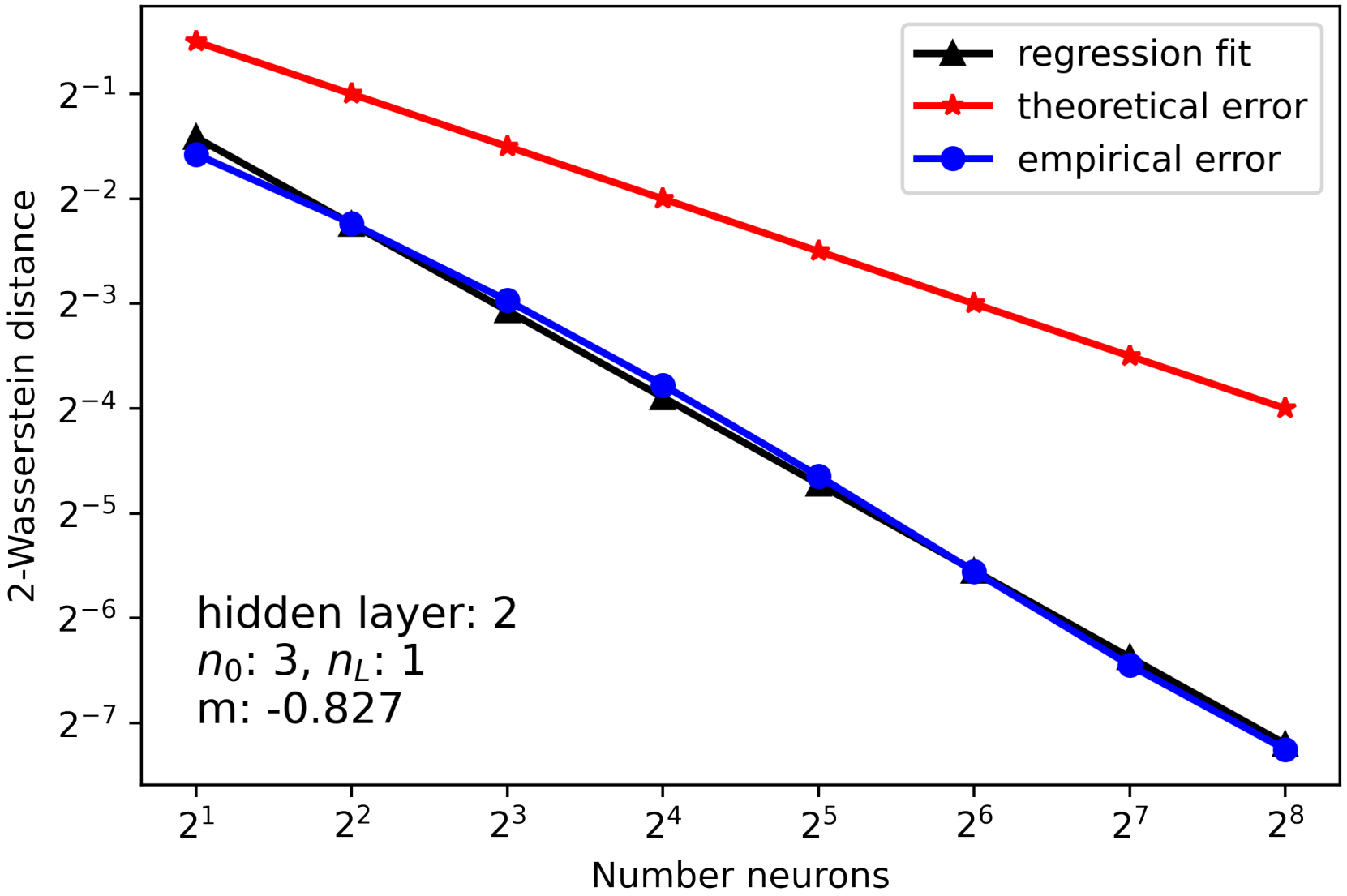
Enlarging the input set



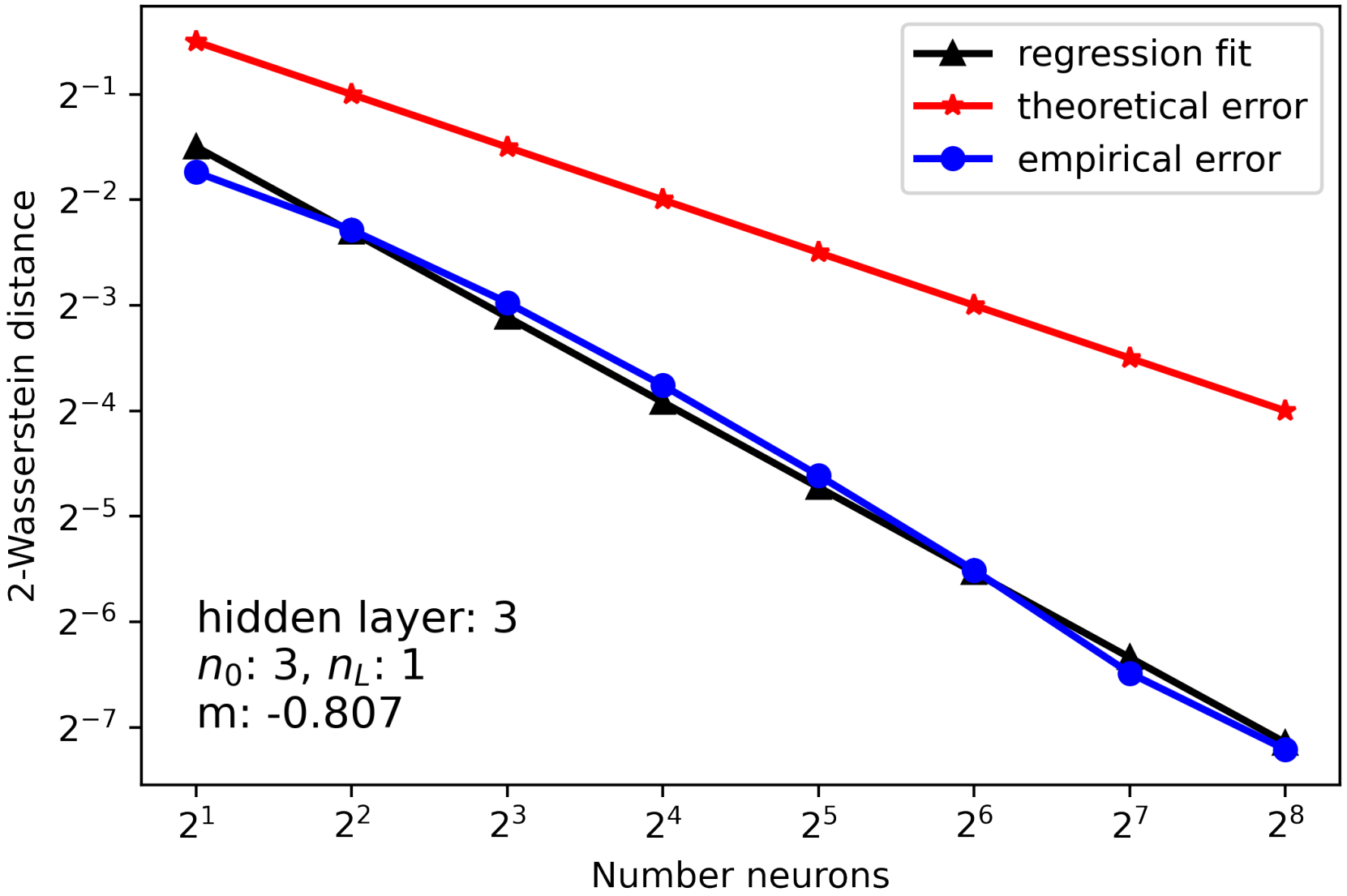
Deeper networks



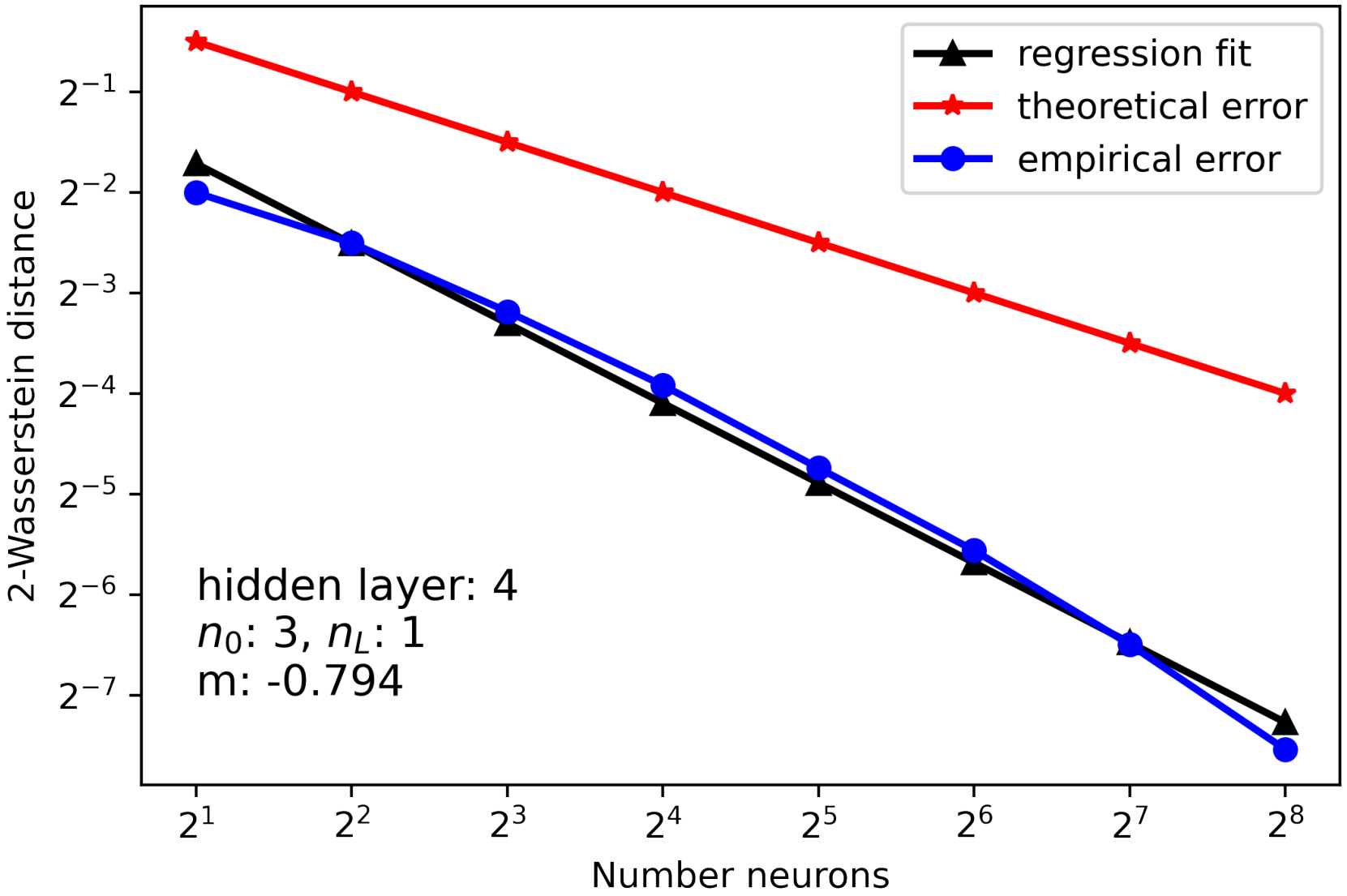
Deeper networks



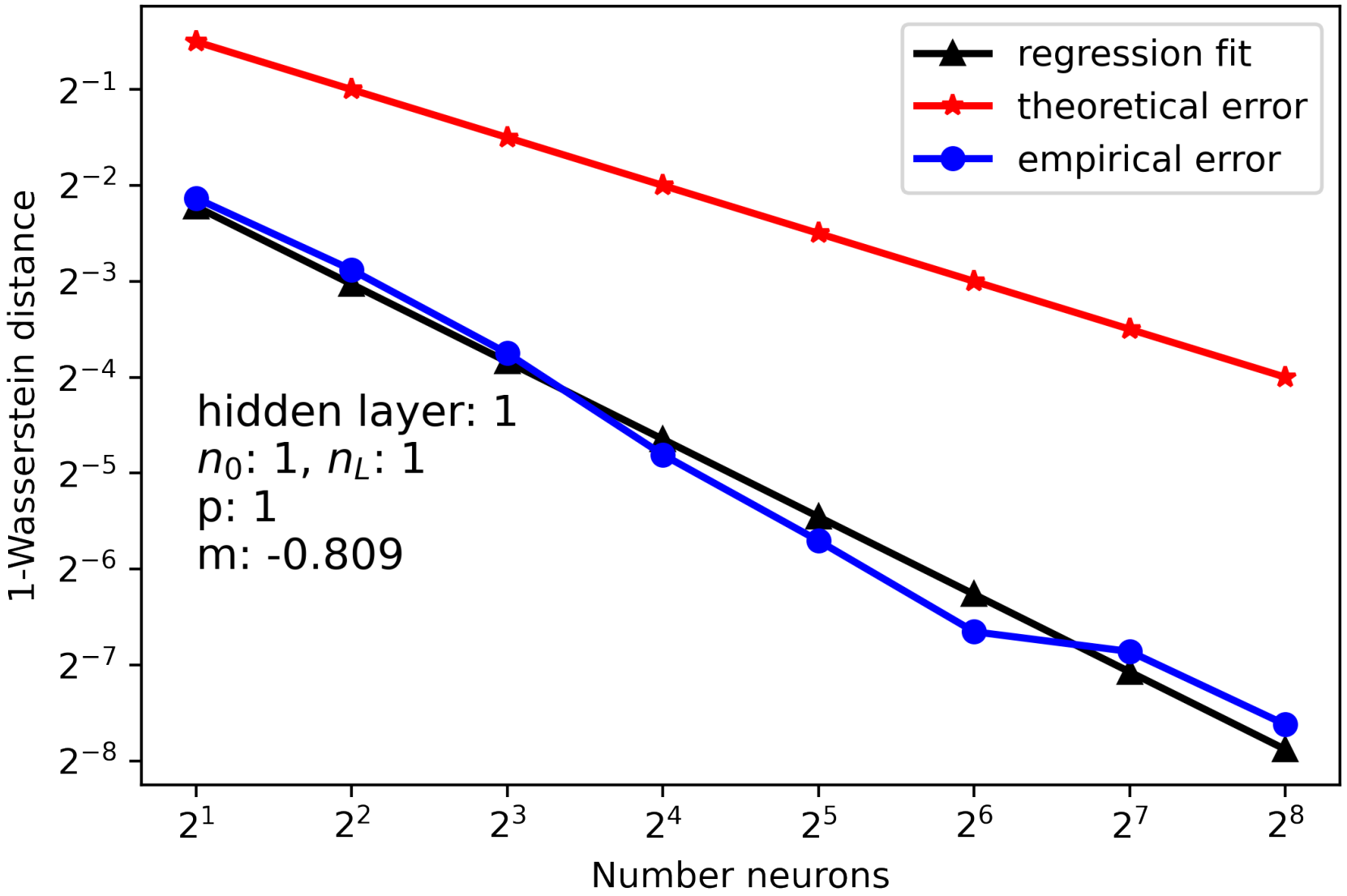
Deeper networks



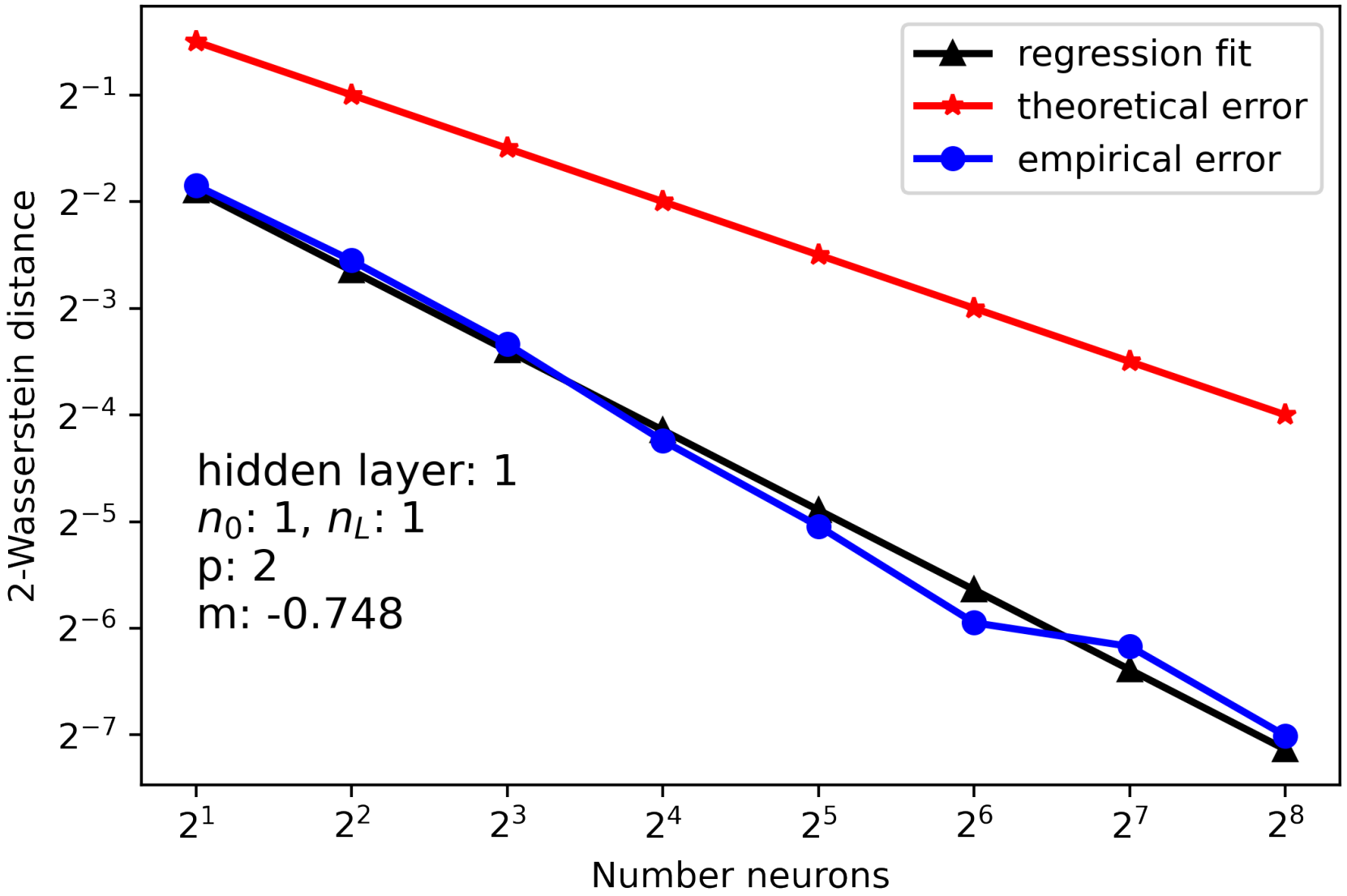
Deeper networks



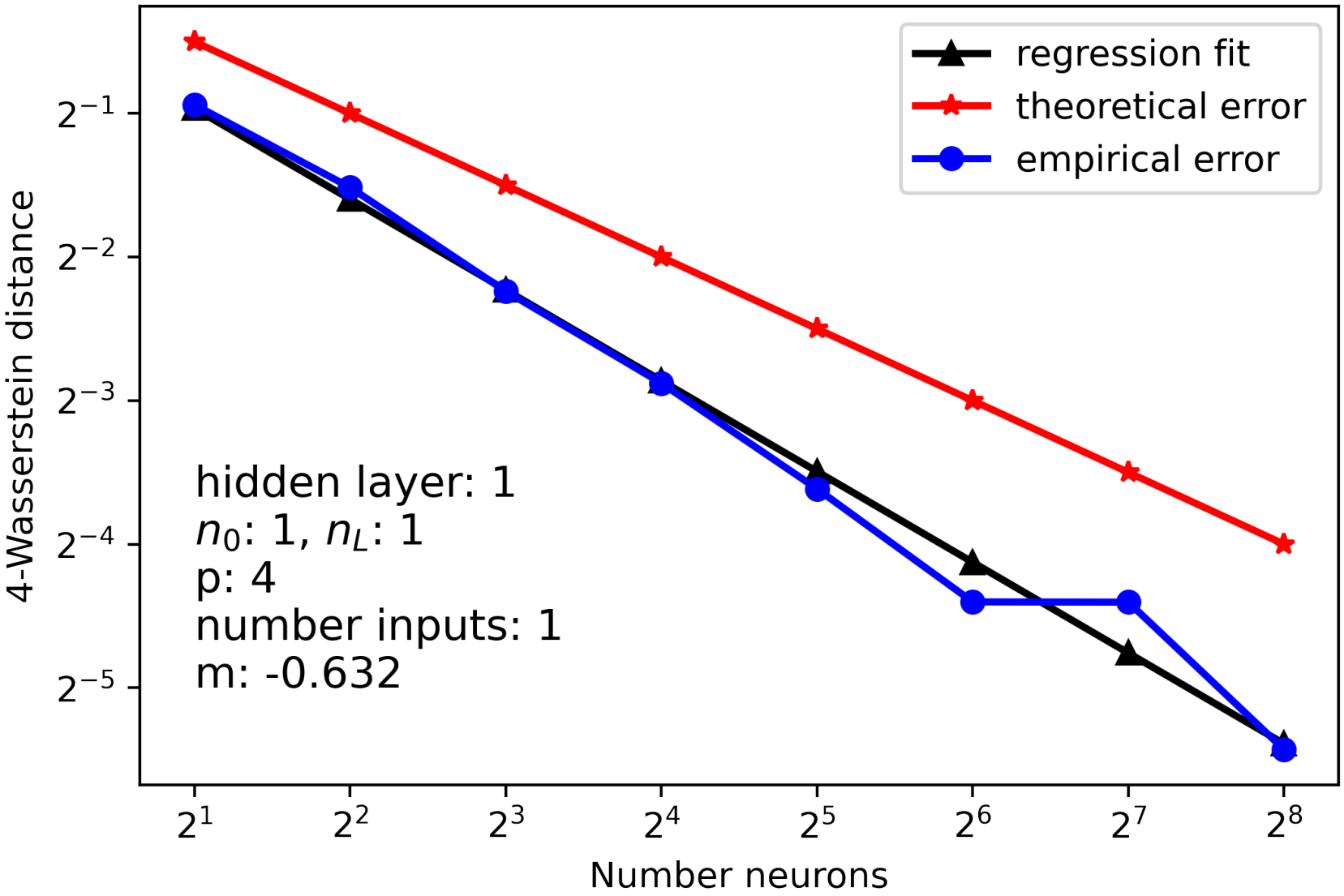
Distances of different order p



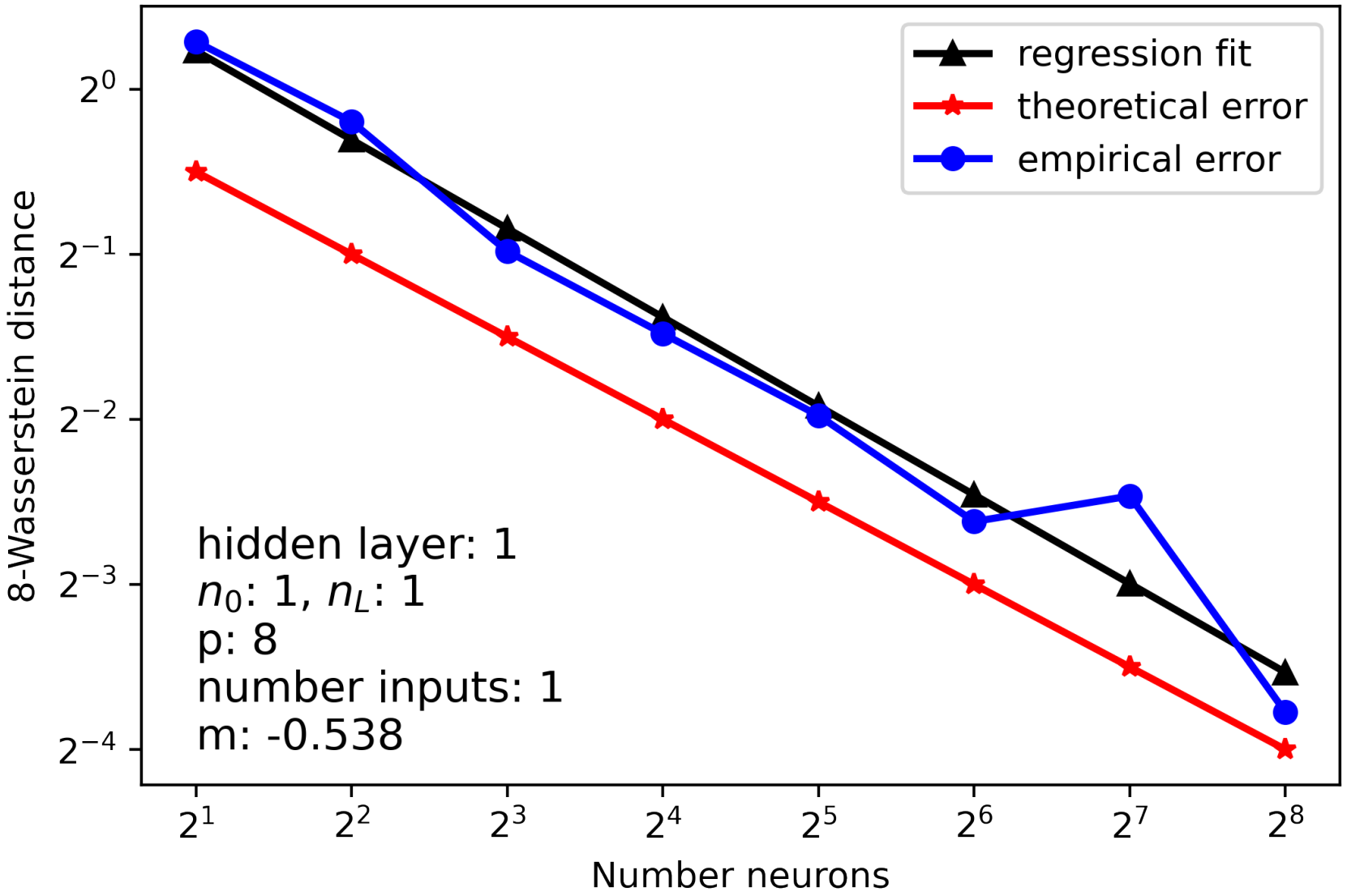
Distances of different order p



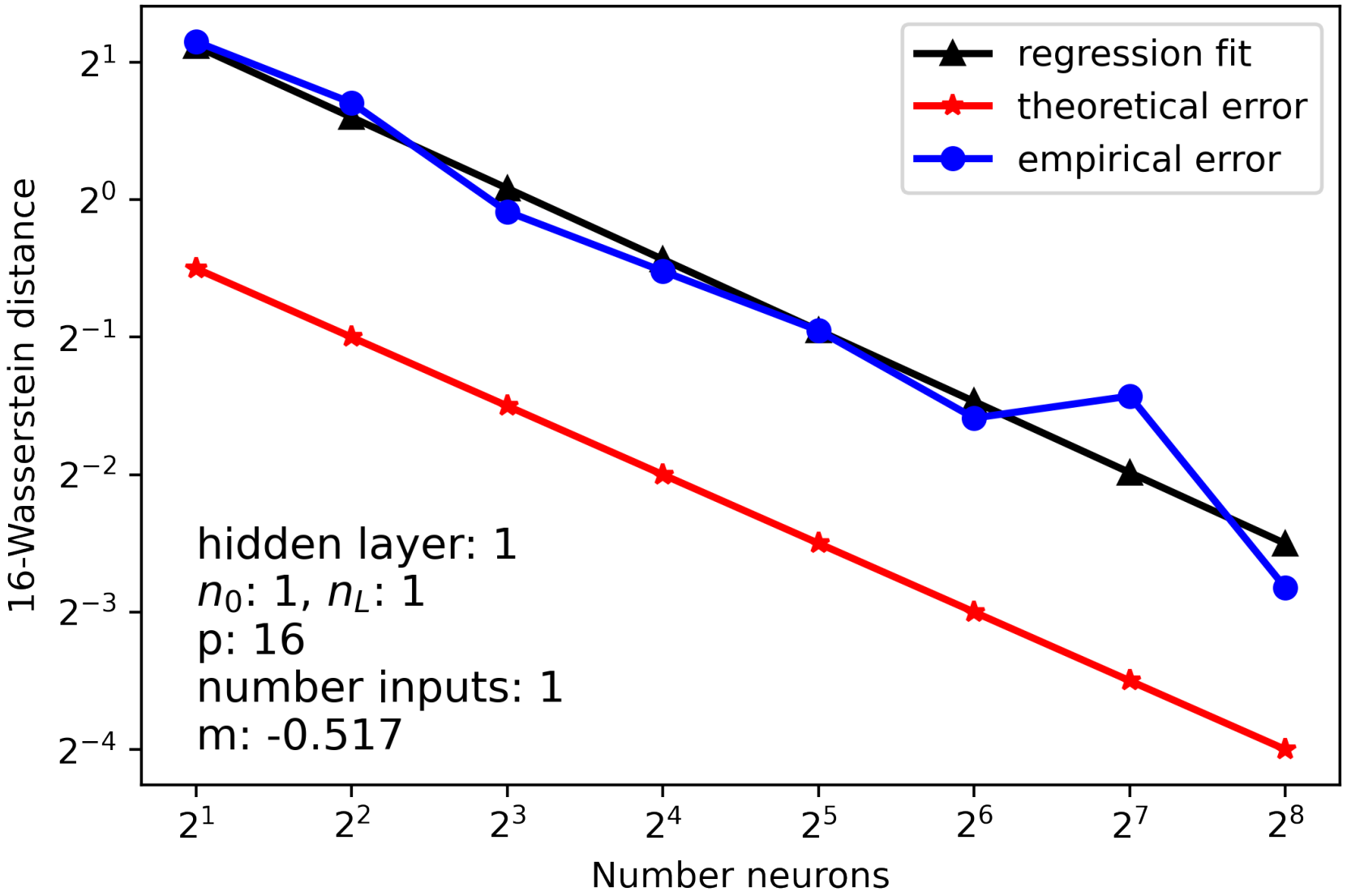
Distances of different order p



Distances of different order p



Distances of different order p



Plan

Supervised learning

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

$$\{(x_i, y_i)\}_{i \in \mathcal{D}} \subseteq \mathbb{R}^{n_0} \times \mathbb{R}^{n_L}$$

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 data

$$h(x_i;\theta^*) \approx y_i \quad \forall i \in \mathcal{D}$$

(hopefully) generalizing well to unseen data $x \mapsto h(x;\theta^*)$.

Supervised learning

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

$$\{(x_i, y_i)\}_{i \in \mathcal{D}} \subseteq \mathbb{R}^{n_0} \times \mathbb{R}^{n_L}$$

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 data

$$h(x_i;\theta^*) \approx y_i \quad \forall i \in \mathcal{D}$$

(hopefully) generalizing well to unseen data $x \mapsto h(x;\theta^*)$.

Two approaches

Variational/frequentist. Fix

- a loss function e.g. *mean squared error* $\|h(x;\theta) - y\|^2$
- a regularization function $R(\theta)$
- *minimize* the empirical risk:

$$\theta^*_V \in \text{argmin}_\theta \sum_{i \in \mathcal{D}} \|h(x_i;\theta) - y_i\|^2 + R(\theta).$$

Bayesian. Fix

- a likelihood $\mathcal{L}(\theta;\mathcal{D}) = p(\mathcal{D} \mid \theta)$
- a prior distribution $p(\theta)$
- compute the posterior

$$p(\theta \mid \mathcal{D}) \propto L(\theta\,;\,\mathcal{D})p(\theta)$$

- *maximum a posteriori* estimate

$$\theta^*_B \in \text{argmax}_\theta\, p(\theta \mid \mathcal{D})$$

Correspondence is up to taking a log and changing sign:

Likelihood	$\mathcal{L}(\theta; \mathcal{D})$	$\leftrightarrow$	$\sum_{i \in \mathcal{D}} \|h(x_i; \theta) - y_i\|^2$	Empirical Loss
Prior	$p(\theta)$	$\leftrightarrow$	$R(\theta)$	Regularization
Posterior	$p(\theta \mid \mathcal{D})$	$\leftrightarrow$	$\sum_{i \in \mathcal{D}} \|h(x_i; \theta) - y_i\|^2 + R(\theta)$	Empirical Risk
MAP	θ^*_B	$\leftrightarrow$	θ^*_V	Minimizer

Remark: the mean squared error

$$\mathcal{L}(\theta; \mathcal{D}) \propto \exp\left(-\sum_{i \in \mathcal{D}} \|h(x_i; \theta) - y_i\|^2\right) = \prod_{i \in \mathcal{D}} \exp\left(-\|h(x_i; \theta) - y_i\|^2\right)$$

yields i.i.d. Gaussian residuals $h(x_i; \theta) - y_i$ (if conditioned upon θ)

Correspondence is up to taking a log and changing sign:

Likelihood	$\mathcal{L}(\theta; \mathcal{D})$	$\leftrightarrow$	$\sum_{i \in \mathcal{D}} \|h(x_i; \theta) - y_i\|^2$	Empirical Loss
Prior	$p(\theta)$	$\leftrightarrow$	$R(\theta)$	Regularization
Posterior	$p(\theta \mid \mathcal{D})$	$\leftrightarrow$	$\sum_{i \in \mathcal{D}} \|h(x_i; \theta) - y_i\|^2 + R(\theta)$	Empirical Risk
MAP	θ^*_B	$\leftrightarrow$	θ^*_V	Minimizer

Remark: the mean squared error

$$\mathcal{L}(\theta; \mathcal{D}) \propto \exp\left(-\sum_{i \in \mathcal{D}} \|h(x_i; \theta) - y_i\|^2\right) = \prod_{i \in \mathcal{D}} \exp\left(-\|h(x_i; \theta) - y_i\|^2\right)$$

yields i.i.d. Gaussian residuals $h(x_i; \theta) - y_i$ (if conditioned upon θ)

Random NN's and Gaussian processes as priors

- Deep networks $f^{(L)}$ are a parametrized family $\theta = (\mathbf{W}, \mathbf{b})$.
- Random Gaussian weights and biases give a prior distribution.
- Gaussian processes $G^{(L)}$ are also a "parametrized" family ($\theta = G^{(L)}$).
- Our inequality shows that the "priors" are close:

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

- What about the "posteriors"?

Lemma

Let

- μ, ν be probabilities on $\mathbb{R}^n$
- with finite second moments $m_2(\mu) = \int \|x\|^2 d\mu$, $m_2(\nu) = \int \|x\|^2 d\nu$
- $g : \mathbb{R}^n \to [0, 1]$ be Lipschitz continuous
- with $\mu(g) := \int g d\mu > 0$, $\nu(g) := \int g d\nu > 0$,

and define

$$\mu_g = \frac{g}{\mu(g)}\mu \quad \nu_g = \frac{g}{\nu(g)}\nu.$$

Then,

$$\mathcal{W}_1(\mu_g, \nu_g) \leq \frac{1}{\mu(g)} \left(1 + \mathrm{Lip}(g)\sqrt{m_2(\nu)} \left(1 + \frac{1}{\nu(g)}\right)\right) \mathcal{W}_2(\mu, \nu).$$

- It holds

$$|\mu(g) - \nu(g)| \leq \mathbb{E}\left[|g(X) - g(Y)|\right] \leq \mathrm{Lip}(g)\mathcal{W}_2(\mu, \nu).$$

- Question: replace $\mathcal{W}_1$ with $\mathcal{W}_2$?

Proof

- We use Kantorovich duality

$$\mathcal{W}_1(\mu_g, \nu_g) = \sup_{\mathrm{Lip}(f)\leq 1} \int f d\mu_g - \int f d\nu_g.$$

- Assume without loss of generality that $f(0) = 0$, hence $|f(x)| \leq \|x\|$. Then

$$\begin{aligned}\int f d\mu_g - \int f d\nu_g &= \frac{1}{\mu(g)} \int fg d(\mu - \nu) + \left(\frac{1}{\mu(g)} - \frac{1}{\nu(g)}\right) \int fg d\nu \\ &\leq \frac{1}{\mu(g)} \int fg d(\mu - \nu) + \frac{\mathrm{Lip}(g)\mathcal{W}_2(\mu, \nu)}{\mu(g)\nu(g)} \int \|x\| d\nu\end{aligned}$$

- Given a $\mathcal{W}_2$-optimal coupling (X, Y) for μ, ν,

$$\begin{aligned}\int fg d(\mu - \nu) &= \mathbb{E}\left[|f(X)g(X) - f(Y)g(Y)|\right] \\ &\leq \mathbb{E}\left[|f(X) - f(Y)|\right] + \mathbb{E}\left[|f(Y)||g(X) - g(Y)|\right] \\ &\leq \mathbb{E}\left[\|X - Y\|\right] + \mathrm{Lip}(g)\mathbb{E}\left[\|Y\|^2\right]^{1/2} \mathbb{E}\left[\|X - Y\|^2\right]^{1/2} \\ &\leq (1 + \mathrm{Lip}(g)\sqrt{m_2(\nu)})\mathcal{W}_2(\mu, \nu).\end{aligned}$$

Application to NN's posterior. Consider

- a likelihood of the form

$$\mathcal{L}(\theta; \mathcal{D}) = g((h(x_i; \theta))_{i \in \mathcal{D}})$$

with $g : \mathbb{R}^{n_L \times |\mathcal{D}|} \to [0, 1]$ Lipschitz, e.g.

$$\mathcal{L}(\theta; \mathcal{D}) = \exp\left(- \sum_{i \in \mathcal{D}} \| h(x_i; \theta) - y_i \|^2 \right)$$

i.e., $g((z_i)_{i \in \mathcal{D}}) = \exp\left(- \sum_{i \in \mathcal{D}} \| z_i - y_i \|^2 \right)$.

- a (finite) test set $(x_j)_{j \in \mathcal{T}}$ and define $\mathcal{X} := (x_i)_{i \in \mathcal{D}} \cup (x_j)_{j \in \mathcal{T}}$.

Then,

- for a NN $h(x; \theta) = f^{(L)}(x)$ (with Gaussian weights and biases) the posterior of $f^{(L)}[\mathcal{X}]$ is

$$p(f^{(L)}[\mathcal{X}] = z | \mathcal{D}) \propto g((z_i)_{i \in \mathcal{D}}) p(f^{(L)}[\mathcal{X}] = z).$$

- for a Gaussian process prior $h(x; \theta) = G^{(L)}(x)$, the posterior is

$$p(G^{(L)}[\mathcal{X}] = z | \mathcal{D}) \propto g((z_i)_{i \in \mathcal{D}}) p(G^{(L)}[\mathcal{X}] = z).$$

Application to NN's posterior. Consider

- a likelihood of the form

$$\mathcal{L}(\theta; \mathcal{D}) = g((h(x_i; \theta))_{i \in \mathcal{D}})$$

with $g : \mathbb{R}^{n_L \times |\mathcal{D}|} \to [0, 1]$ Lipschitz, e.g.

$$\mathcal{L}(\theta; \mathcal{D}) = \exp\left(- \sum_{i \in \mathcal{D}} \| h(x_i; \theta) - y_i \|^2 \right)$$

i.e., $g((z_i)_{i \in \mathcal{D}}) = \exp\left(- \sum_{i \in \mathcal{D}} \| z_i - y_i \|^2 \right)$.

- a (finite) test set $(x_j)_{j \in \mathcal{T}}$ and define $\mathcal{X} := (x_i)_{i \in \mathcal{D}} \cup (x_j)_{j \in \mathcal{T}}$.

Then,

- for a NN $h(x; \theta) = f^{(L)}(x)$ (with Gaussian weights and biases) the posterior of $f^{(L)}[\mathcal{X}]$ is

$$p(f^{(L)}[\mathcal{X}] = z | \mathcal{D}) \propto g((z_i)_{i \in \mathcal{D}}) p(f^{(L)}[\mathcal{X}] = z).$$

- for a Gaussian process prior $h(x; \theta) = G^{(L)}(x)$, the posterior is

$$p(G^{(L)}[\mathcal{X}] = z | \mathcal{D}) \propto g((z_i)_{i \in \mathcal{D}}) p(G^{(L)}[\mathcal{X}] = z).$$

Corollary (Bayesian posterior bounds)

If $\mathbb{E}\left[g(G^{(L)}((x_i)_{i\in\mathcal{D}})\right] > 0$, then for $n := \min\{n_1, \dots, n_{L-1}\}$ large enough,

$$\mathcal{W}_1\left(p(f^{(L)}[\mathcal{X}] = \cdot|\mathcal{D}), p(G^{(L)}[\mathcal{X}] = \cdot|\mathcal{D})\right) \leq \frac{C}{\sqrt{n}}.$$

where $C \in (0, \infty)$ does not depend on n.

Example: in the mean squared error case

$$\mathcal{L}(\theta; \mathcal{D}) = \exp\left(-\sum_{i\in\mathcal{D}} \|h(x_i;\theta) - y_i\|^2\right),$$

the posterior

$$p(G^{(L)}[\mathcal{X}] = \cdot|\mathcal{D}) = g((z_i)_{i\in\mathcal{D}})p(G^{(L)}[\mathcal{X}] = z)$$

is also Gaussian (actually explicitly computable)

$$\Rightarrow p(f^{(L)}[\mathcal{X}] = \cdot|\mathcal{D}) \quad \text{is quantitatively close to a Gaussian.}$$

Corollary (Bayesian posterior bounds)

If $\mathbb{E}\left[g(G^{(L)}((x_i)_{i\in\mathcal{D}})\right] > 0$, then for $n := \min\{n_1, \dots, n_{L-1}\}$ large enough,

$$\mathcal{W}_1\left(p(f^{(L)}[\mathcal{X}] = \cdot|\mathcal{D}), p(G^{(L)}[\mathcal{X}] = \cdot|\mathcal{D})\right) \leq \frac{C}{\sqrt{n}}.$$

where $C \in (0, \infty)$ does not depend on n.

Example: in the mean squared error case

$$\mathcal{L}(\theta; \mathcal{D}) = \exp\left(-\sum_{i\in\mathcal{D}} \|h(x_i;\theta) - y_i\|^2\right),$$

the posterior

$$p(G^{(L)}[\mathcal{X}] = \cdot|\mathcal{D}) = g((z_i)_{i\in\mathcal{D}})p(G^{(L)}[\mathcal{X}] = z)$$

is also Gaussian (actually explicitly computable)

$$\Rightarrow p(f^{(L)}[\mathcal{X}] = \cdot|\mathcal{D}) \quad \text{is quantitatively close to a Gaussian.}$$

Plan

Summary

For a deep NN with hidden layer sizes $\min\{n_1, \dots, n_{L-1}\} =: n \to \infty$

$$\text{Gaussian initialization:} \quad \mathcal{W}_2\left(f^{(L)}[\mathcal{X}], G^{(L)}[\mathcal{X}]\right) \leq \frac{C_{\text{prior}}}{\sqrt{n}},$$

$$\text{Bayesian posterior:} \quad \mathcal{W}_1\left(p(f^{(L)}[\mathcal{X}] = \cdot|\mathcal{D}), p(G^{(L)}[\mathcal{X}] = \cdot|\mathcal{D})\right) \leq \frac{C_{\text{posterior}}}{\sqrt{n}}.$$

We kept technicalities at a minimum:

- in the initialization bound $\mathcal{W}_2$ can be replaced with $\mathcal{W}_p$ or other distances
- other network architecture (convolutional, graph NN's)
- non-Gaussian the parameters, e.g. discrete or stable laws

Open questions:

- Sharpness of the bounds
- Properties of the optimal transport map (e.g. w.r.t. hidden layer sizes)
- What happens during/after training (e.g. via SGD)?

Summary

For a deep NN with hidden layer sizes $\min\{n_1, \dots, n_{L-1}\} =: n \to \infty$

Gaussian initialization: $\mathcal{W}_2\left(f^{(L)}[\mathcal{X}], G^{(L)}[\mathcal{X}]\right) \leq \frac{C_{\text{prior}}}{\sqrt{n}},$

Bayesian posterior: $\mathcal{W}_1\left(p(f^{(L)}[\mathcal{X}] = \cdot|\mathcal{D}), p(G^{(L)}[\mathcal{X}] = \cdot|\mathcal{D})\right) \leq \frac{C_{\text{posterior}}}{\sqrt{n}}.$

We kept technicalities at a minimum:

- in the initialization bound $\mathcal{W}_2$ can be replaced with $\mathcal{W}_p$ or other distances
- other network architecture (convolutional, graph NN's)
- non-Gaussian the parameters, e.g. discrete or stable laws

Open questions:

- Sharpness of the bounds
- Properties of the optimal transport map (e.g. w.r.t. hidden layer sizes)
- What happens during/after training (e.g. via SGD)?


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


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


Bordino, A., Favaro S., and Fortini S.
Non-asymptotic approximations of Gaussian neural networks via second-order Poincaré inequalities
arXiv preprint arXiv:2304.04010 (2023).


Favaro, S., Hanin B., Marinucci D., Nourdin I., and Peccati G.
Quantitative clts in deep neural networks
arXiv preprint arXiv:2307.06092 (2023).


Apollonio, N., De Canditiis D., Franzina G., Stolfi P., and Torrisi G.-L.
Normal approximation of random gaussian neural networks
arXiv preprint arXiv:2307.04486 (2023).