Expressivity of Neural Networks for distributions

Ali Ramlaoui ali.ramlaoui@student-cs.fr
Théo Saulus theo.saulus@student-cs.fr
CentraleSupélec, France

February 13, 2024

1 Introduction and contributions

The paper we study in this report is Lee et al. [2017]. The main contribution of the paper is to show that a neural network with a multiple hidden layers can approximate any composition of sufficiently smooth and regular functions, under some conditions. It builds on the work of Barron Barron [1993], who showed that a neural network with a single hidden layer can approximate any continuous function. The paper extends this result for more complex functions under the constrain that the neural network has to be deeper and applies the result to functions applied to probability distributions.

We will focus on the main theorem of the paper and discuss its proof, and provide an experiment to illustrate the result. Our code is available publicly at `https://github.com/Ramlaoui/tdl-express-distributions`.

2 Main result

Notations A Neural Network (NN) is written as a function $g : \mathbb{R}^d \to \mathbb{R}^m$ with $g(x) = W_L\sigma(W_{L-1}\sigma(\dots\sigma(W_1x + b_1)\dots) + b_{L-1}) + b_L$, where $W_i \in \mathbb{R}^{n_i \times n_{i-1}}$ are the weights, $b_i \in \mathbb{R}^{n_i}$ are the biases, and σ is the activation function. We will mainly consider the case where σ is a sigmoidal function, i.e. $\lim_{x\to-\infty}\sigma(x) = 0$ and $\lim_{x\to+\infty}\sigma(x) = 1$. We denote $g_{l:1} = g_l \circ \dots \circ g_1$ to simplify the expressions when needed.

Barron function A function $f : B \subset \mathbb{R}^d \to \mathbb{R}$ is a Barron function if there exists a function g such that the restriction $g|_B = f$ that is Fourier inversible in B, i.e.

$$\forall x \in B, \quad g(x) = g(0) + \int_{\mathbb{R}^d} (e^{i\langle x,\omega\rangle} - 1)\hat{g}(\omega)d\omega,$$

and such that $\omega \mapsto w\hat{g}(w)$ has a finite L^1 norm. We denote the set of functions that are Fourier inversible in B by $\mathcal{F}_B$. The Barron constant of f is defined as

$$C_f = \inf_{g\in\mathcal{F}_B|g|_B=f} \int_{\mathbb{R}^d} \|\omega\|_B|\hat{g}(\omega)|d\omega,$$

where $\|\omega\|_B = \sup_{x\in B}|\langle x,\omega\rangle|$ is the norm of ω in the dual space of B.

Theorem 1. *Let $f : B \subset R^d \mapsto \mathbb{R}$ be a Barron function with Barron constant C_f, and μ be a probability distribution on $\mathbb{R}^d$ with support B. Then, for every k there exists a neural network g with one hidden layer and k neurons such that:*

$$\|f - g\|^2_{L^2(\mu)} := \int_B (f-g)^2 d\mu \leq \frac{(2C_f)^2}{k}.$$

Moreover, we have $g : x \mapsto w_2^\top\sigma(w_1^\top x + b_1) + b_2$ such that $\sum_{i=1}^k |w_2^{(i)}| \leq 2C_f$.

2.1 Approximation of compositions of functions

The main theorem of the paper builds on the previous result to show similar results for compositions of functions but with stronger regularity assumptions. A function $f_i : K^{i-1} \subset \mathbb{R}^{m_{i-1}} \to \mathbb{R}^{m_i}$ with $K^i \subseteq f(K^{i-1})$ is sufficiently smooth for the composition and the theorem if:

1. f_i is Lipschitz continuous with constant $L_i \leq 1$,

2. f_i is a Barron function on $K^{i-1} + sB_{m_{i-1}}$, where $B_{m_{i-1}}$ is the unit ball of $\mathbb{R}^{m_{i-1}}$ and $s > 0$ (definition set with a neighborhood) with Barron constant C_i,

Theorem 2. *For $\varepsilon > 0$, $s > 0$, and $L \geq 1$. Let for every $i \in \{1, \dots, L\}$, $f_i : \mathbb{R}^{m_{i-1}} \to \mathbb{R}^{m_i}$ be functions that are sufficiently smooth as defined above, and μ_0 be a probability distribution on $\mathbb{R}^{m_0}$ with support in K^0. Define $K^i := f_i(K^{i-1})$, and $f := f_L \circ \dots \circ f_1$. Then, there exists a neural network g with L hidden layers and $\left\lceil \frac{4C_i^2 m_i}{\varepsilon^2} \right\rceil$ nodes in each hidden layer i such that:*

$$\|f - g\|^2_{L^2(\mu_0)} \leq (L\varepsilon)^2 \left((2C_L\sqrt{m_L} + D)^2 \frac{L}{3s^2} + 1 \right) \underset{L \text{ or } m_L \to \infty}{=} O\left(\frac{\varepsilon^2 L^3 C_L^2 m_L}{s^2} \right),$$

where D is the diameter of K^L, i.e. $D = \sup_{x,y \in K^L} \|x - y\|$.

This result shows that a neural network can approximate any composition of sufficiently smooth functions with an error that does not grow not exponentially with respect to the space dimensions or the number of functions, and provides a bound on the number of neurons needed to achieve this error. Moreover, Lee et al. [2017] show that the composition of Barron functions is not necessarily a Barron function by constructing a counter-example.

2.2 Sketch of the proof

The idea of the proof is to recursively show that a composition of L functions that verify the smoothness condition of the theorem can be approximated by a neural network on an input set $S^L \subset \mathbb{R}^{m_0}$ that is large enough with a given error and then to extend the result to the whole support of the probability distribution μ_0 which is K^0.

We now provide a step-by-step proof for Theorem 2, intended to be more detailed than the version of the paper. We start by showing that there exists a set $S^L \subset \mathbb{R}^{m_0}$ such that the composition of the functions can be approximated by a neural network on S^L with a given error, as illustrated in Figure 2. More formally, we will show the following theorem first, and then prove the main theorem:

Theorem 3. *Under the same previous assumptions, there exists a NN g with L layers and $S \subset \mathbb{R}^{m_0}$ such that $\mu_0(S) \geq 1 - \frac{\varepsilon^2}{s^2} \sum_{i=1}^{L-1} i^2$ and $\int_S \|f_{L:1} - g\| d\mu_0 \leq L^2\varepsilon^2$.*

Base case $L = 1$. The base case of the proof is to show that a single function $f_1 : \mathbb{R}^{m_0} \mapsto \mathbb{R}^{m_1}$ that verifies the smoothness condition can be approximated by a neural network on an input set $S^1 = \mathbb{R}^{m_0}$ with a given error. Using Theorem 1, this is directly obtained because f_1 is a Barron function:

$$\|f - g\|^2_{L^2(\mu_0)} \leq \frac{4C_1}{\left\lceil \frac{4C_1^2 m_1}{\varepsilon^2} \right\rceil} \leq \frac{\varepsilon^2}{m_1} \leq \varepsilon^2$$

Induction step Let's assume that there exists $S^{L-1} \subset \mathbb{R}^{m_0}$ such that $\mu_0(S^{L-1}) \geq 1 - \frac{\varepsilon^2}{s^2} \sum_{i=1}^{L-2}$ and $g_1, \dots, g_{L-1}$ the $L-1$ layers of the NN $g_{L-1:1} := g_{L-1} \circ \dots \circ g_1$ such that $\int_{S^L} \|f_{L-1:1} - g_{L-1:1}\| d\mu_0 \leq L^2\varepsilon^2$.

By Theorem 1, we have that for all component $j \in \{1, \dots, m_L\}$, for **any** distribution μ on $K'_{L-1} \subseteq \mathbb{R}^{m_{L-1}}$, and for all number of neurons $k \in \mathbb{N}$, there exists a 1-layer NN $g_{L,j}$ with k nodes such that $\int_{\mathbb{R}^{m_{L-1}}} (f_{L,j} - g_{L,j})^2 d\mu \leq \frac{4C'^2}{k}$.

Then, we use this result for the set $S_L := S_{L-1} \cap \{x : g_{L-1:1}(x) \in K_{L-1} + sB_{m_{L-1}}\}$, for the support $K_L + sB_{m_L}$, for $\left\lceil \frac{4C_L^2 m_L}{\varepsilon^2} \right\rceil$ neurons, and for a carefully crafted distribution $\mu' : S \subset \mathbb{R}^{m_{L-1}} \mapsto \mu_0(g^{-1}_{L-1:1}(S) \cap S_L)$ (known as the pushforward of $\mathbf{1}_{S_L}\mu_0$ by $g_{L-1:1}$), all of which can be proven to be suitable to apply the theorem:

$$\int_{\mathbb{R}^{m_{L-1}}} (f_{L,j} - g_{L,j})^2 d\mu' \leq \frac{4C_L^2}{\left\lceil \frac{4C_L^2 m_L}{\varepsilon^2} \right\rceil} \leq \frac{\varepsilon^2}{m_L}$$

Since this is valid for every m_L component, we can sum to obtain:

$$\int_{\mathbb{R}^{m_{L-1}}} \|f_L - g_L\|^2 d\mu' \leq \varepsilon^2$$

We provide in Annex B detailed computatiosn to upper-bound the integral by a given error:

$$\left(\int_{\mathbb{R}^{m_L}} 1_{S_L}\|f_{1:L}-g_{1:L}\|^2 d\mu_0\right)^{\frac{1}{2}} \leq L\varepsilon$$

Now, let's check that $\mu_0(S_L)$ is lower-bounded as expected. We start by bounding the measure of $S_{L-1}\cap\{x : g_{L-1:1}(x)\notin K_{L-1}+sB_{m_{L-1}}\}$ first:

$$\begin{aligned}
&\mu_0(S_{L-1}\cap\{x : g_{L-1:1}(x)\notin K_{L-1}+sB_{m_{L-1}}\})\\
&\quad\leq \mu_0(S_{L-1}\cap\{x : \|f_{L-1:1}(x)-g_{L-1:1}(x)\|\geq s\}) && \text{because } \forall x\in Supp(\mu_0), f_{L-1:1}(x)\in K_{L-1}\\
&\quad\leq \frac{(L-1)^2\varepsilon^2}{s^2} && \text{by Markov inequality, and induction hyp. on } S_{L-1}.
\end{aligned}$$

Thus, we obtain:

$$\begin{aligned}
\mu_0(S_L) &= \mu_0(S_{L-1}\cap\{x : g_{L-1:1}(x)\in K_{L-1}+sB_{m_{L-1}}\})\\
&\geq \mu_0(S_{L-1})\\
&\qquad -\mu_0(S_{L-1}\cap\{x : g_{L-1:1}(x)\notin K_{L-1}+sB_{m_{L-1}}\}) && \text{by the set inclusion } A\backslash(A\cap B^c)\subseteq A\cap B\\
&\geq 1-\frac{(L-1)^2\varepsilon^2}{s^2} && \text{using previous computations}\\
&\geq 1-\sum_{i=1}^{L-1} i^2\frac{\varepsilon^2}{s^2} && \text{by } \sum_{i=1}^{L-1} i^2=(L-1)^2+\ldots+1. \qquad (*)
\end{aligned}$$

This concludes the induction, and proves Theorem 3. Let's conclude the proof of Theorem 2. Our aim is here to remove the dependency on the sets S_l, which is achieved by bounding S_l^c.

First, we bound the diameter of the range of g_L:

$$\begin{aligned}
&\sup_{y,y'\in Im(g_L)}\|y-y'\|\\
&=\sup_{x,x'\in\mathbb{R}^{m_{L-1}}}\|W_L\sigma(x)+b_L-(W_L\sigma(x')+b_L)\|\\
&\leq 2C_L\sqrt{m_L}\sup_{x,x'\in\mathbb{R}^{m_{L-1}}}\|\sigma(x)-\sigma(x')\| && \text{using Th. 1 to obtain } \forall j\leq m_L, \sum_{i=1}^{r}|W_L^{(j,i)}|\leq 2C_L\\
&\leq 2C_L\sqrt{m_L}\cdot 1 && \text{using } Im(\sigma)\subseteq[0,1].
\end{aligned}$$

Now, by bounding successively with trivial bounds, we have for all $x\in K_{L-1}$:

$$\begin{aligned}
\|g_L(x)-f_L(x)\| &\leq \sup_{x'\in K_{L-1}}\left(\|g_L(x)-g_L(x')\|+\|g_L(x')-f_L(x)\|\right)\\
&\leq 2C_L\sqrt{m_L}+\sup_{y,y'\in K_L}\|x-x'\|\\
&\leq 2C_L\sqrt{m_L}+D
\end{aligned}$$

Furthermore, we have:

$$\mu_0(S_L^c)=1-\mu_0(S_L)\leq\left(\sum_{i=1}^{L-1}i^2\right)\frac{\varepsilon^2}{s^2}=\frac{(L-1)L(2(L-1)+1)}{6}\frac{\varepsilon^2}{s^2}\leq\frac{L^3\varepsilon^2}{3s^2} \qquad (**)$$

Finally,

$$\begin{aligned}
&\int_{K_0}||f_{1:L}-g_{1:L}||^2 d\mu_0\\
&\leq\int_{S_l}||f_{1:L}-g_{1:L}||^2 d\mu_0+\int_{S_L^c}||f_{1:L}-g_{1:L}||^2 d\mu_0\\
&\leq L^2\varepsilon^2+(2C\sqrt{m_L}+D)^2\frac{L^3\varepsilon^2}{3s^2}. && \text{using Theorem 3, and previous computations.}
\end{aligned}$$

This concludes the proof of Theorem 2.

2.3 Improvement of the approximation bound

In the previous computations, two steps cought our attention, and led us to propose a better bound. More specifically, the scalar $\sum_{i=1}^{L-1} i^2$ is used two times: once as a trivial upper bound $(L-1)^2$ in (∗), and once as a trivial lower bound for L^3 (∗∗).

Instead, we propose to not use this bound, and stick with $(L-1)^2$. Thus, we obtain at step (∗∗) that $\mu_0(S_L^c) = 1 - \mu_0(S_L) \leq \frac{(L-1)^2\varepsilon^2}{s^2}$. Finally, we can conclude:

$$\int_{K_0} ||f_{1:L} - g_{1:L}||^2 d\mu_0 \leq L^2\varepsilon^2 + (2C\sqrt{m_L} + D)^2 \frac{(L-1)^2\varepsilon^2}{s^2} \leq L^2\varepsilon^2 \left(1 + \frac{(2C\sqrt{m_L} + D)^2}{s^2}\right)$$

This improvement is significant, since the dependency in L is now only quadratic instead of cubic. In the following, we will stick to the paper version, for the sake of global coherence.

2.4 Corollary for approximating distributions

We keep the same notations as in the previous sections. The main theorem of the paper can be applied to the approximation of probability distributions:

Corollary 1. *If $X \sim \mu_0$, then it is possible to quantify the approximation error of the distribution of $g(X)$ to the distribution of $f(X)$ using the Wasserstein distance:*

$$W_2(f(X), g(X)) \triangleq \left(\inf_{\pi \in \Pi(f(X), g(X))} \int_{\mathbb{R}^{m_L} \times \mathbb{R}^{m_L}} \|x - y\|^2 d\pi(x, y)\right)^{\frac{1}{2}} \leq L\varepsilon \sqrt{(2C_L\sqrt{m_L} + D)^2 \frac{L}{3s^2} + 1},$$

where $\Pi(f(X), g(X))$ is the set of probability measures on $\mathbb{R}^{m_0} \times \mathbb{R}^{m_L}$ with marginals the distributions of $f(X)$ and $g(X)$.

3 Experiments

In this section, we try to illustrate the direct corollary of the main theorem of the paper allowing to measure the expressive power of neural networks for approximating distributions.

We consider multiple examples of compositions of Barron functions applied to probability distributions, and we train neural networks to approximate these compositions and measure the Wasserstein distance between the approximated distribution and the true distribution, along with the validation loss.

Since the approximation errors of the theorems are only upper bounds and that the Barron function constants are not known in practice, we try to observe the behavior of the approximation error with respect to the number of neurons in the hidden layers of the neural network and the number of hidden layers.

3.1 Setup

For every experiment, we choose an input distribution μ_0 and we consider a composition of l functions $f_1, \ldots, f_l$ that we apply to $X \sim \mu_0$. We then consider a neural network with L hidden layers and k neurons in each hidden layer, and we train it to approximate the output distribution of the composition of functions. At every training step, we sample N points from the input distribution and we compute the output distribution of the composition of functions, and we train the neural network to minimize the Mean Squared Error (MSE). We then measure the Wasserstein distance between the approximated distribution and the true distribution, and we measure the validation loss. We then repeat the experiment for different values of k and L and we observe the behavior of the approximation error with respect to these parameters.

We experiment with mulitple prior distributions on $\mathbb{R}^d$, mainly a Gaussian distribution, a mixture of gaussians, and uniform distribution on the unit ball or the unit cube. We also consider linear, quadratic, radial basis, trigonometric and sigmoid functions for the compositions. A comprehensive description of the experimental setting is available in Annex C.

3.2 Results

Composing similar functions does not always lead to a very interesting output distribution and the output space sometimes collapses to a point depending on the function being used. Moreover, the composition of simple functions such as linear functions without any non-linearity is still a linear function so it stays a Barron function. In order to get more interesting results, we consider the compositions of different functions, and we observe the behavior of the approximation error with respect to the number of neurons and the number of functions.

Figure 4 shows the loss function obtained by a neural network of depth 4 trained to approximate a composition of l functions that are quadratic, linear or sigmoidal. We notice that until $l = 10$, the neural network is correctly able to approximate the function. However, once 10 or more functions are composed, $L = 4$ is no longer enough and deeper neural networks are needed to improve the results. This shows that the depth needed to approximate the distribution obtained from a composition of Barron functions seems to be correlated to the number of functions.

We present results of multiple experiments for composition of multiple linear, sigmoidal and trigonometric functions and the approximation results in terms of Wasserstein distance and validation loss obtained for different Neural Network widths and depths in Table 1. As expected, both the Wasserstein distance and the validation loss decrease with the number of neurons and the number of layers.

It is difficult to validate the theorem and the bounds provided by the theorem because it is only an existence theorem: there is no guarantee to reach the parameters that verify the theorems. Moreover, deeper neural networks are way more difficult to train and the loss does not always converge because of the large number of parameters to train as can be seen in Figure 5. On top of it, the Barron constants are not known in practice, which prevents us to try an architecture with exactly the right number of neurons.

Nonetheless, figure 6 shows that the neural networks are able to learn correctly with multiple different priors. The batch size used for sampling from the distribution at every epoch seems to be important to correctly learn multimodal distributions. Figure 8 shows the original output distribution and the approximated distribution for different neural network sizes for a composition of 6 linear, sigmoid and trigonometric functions applied to a Gaussian input distribution.

4 Conclusion

In this report, we presented the main result of Lee et al. [2017] and provided a sketch of the proof, and an improvement on the bounds. We also implemented from scratch an experiment to illustrate the result with composition of functions and neural networks of varying widths and depths. Further experiments could be conducted with longer training and more attempts, to try to measure Barron constants and to get a better experimental evaluation of the error bound.

References

Andrew R Barron. Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Transactions on Information theory*, 39(3):930–945, 1993.

Holden Lee, Rong Ge, Tengyu Ma, Andrej Risteski, and Sanjeev Arora. On the ability of neural nets to express distributions. In *Conference on Learning Theory*, pages 1271–1296. PMLR, 2017.

A Illustrations

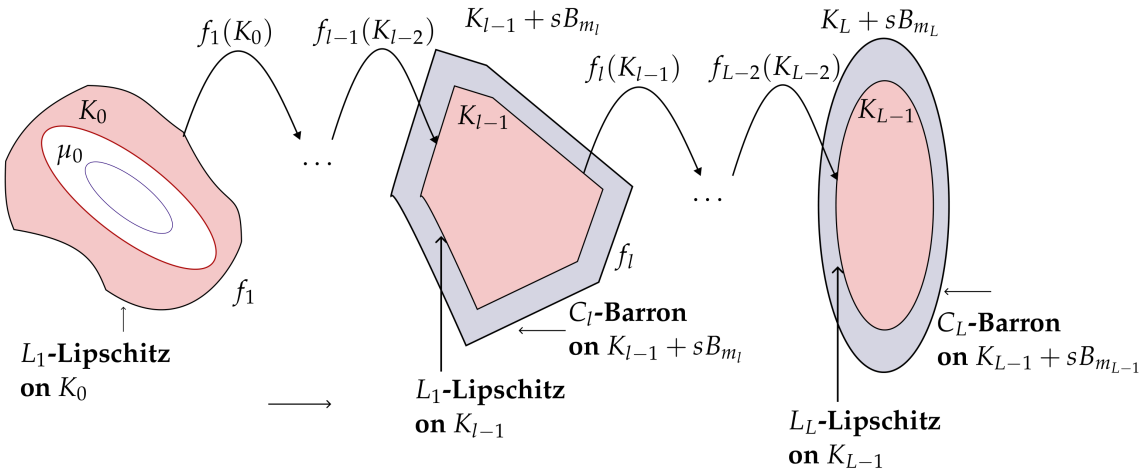

Figure 1: Illustration of the smoothness condition for the composition of functions

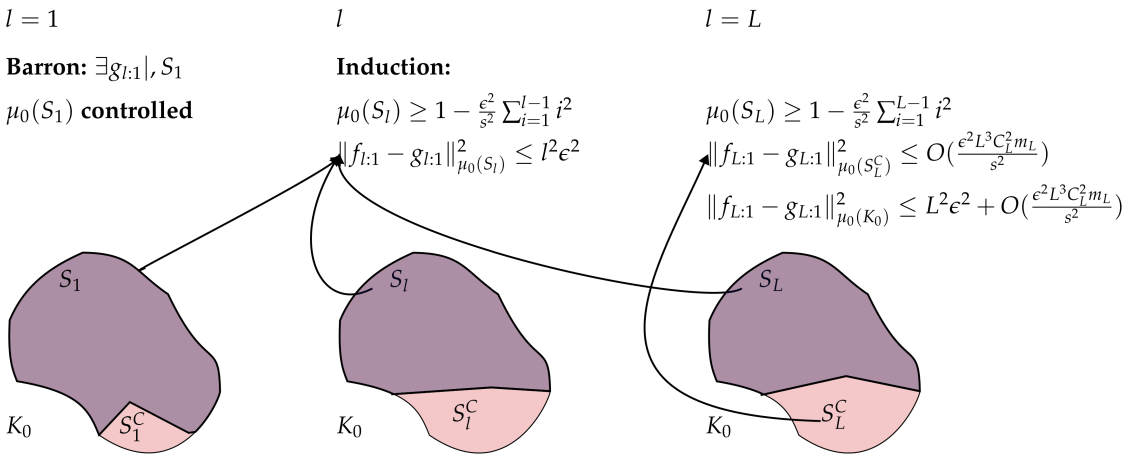

Figure 2: Sketch of the proof

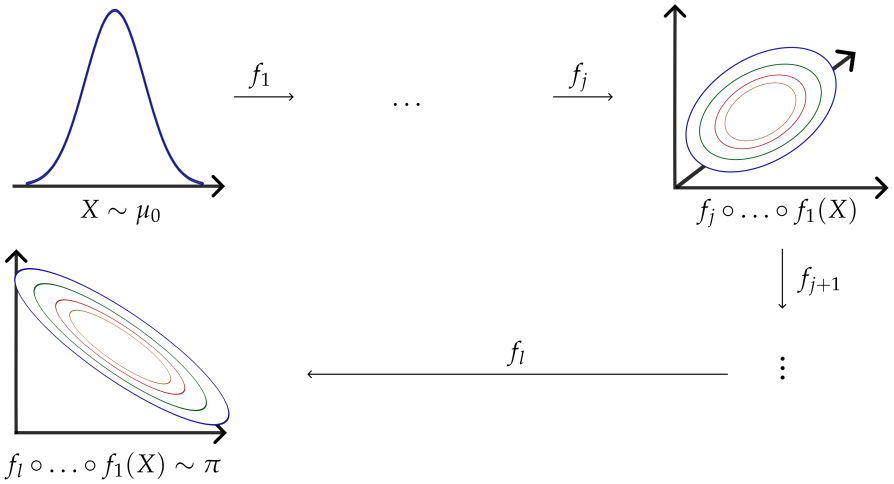

Figure 3: Applications to approximating distributions

B Eluded computation steps

Here is the details of the eluded computations, using previously defined notations:

$$
\begin{aligned}
&\left(\int_{\mathbb{R}^{m_L}} 1_{S_L}\|f_{1:L} - g_{1:L}\|^2 d\mu_0\right)^{\frac{1}{2}} \\
&= \left(\int_{\mathbb{R}^{m_L}} 1_{S_L}\|f_L \circ f_{1:L-1} - g_L \circ g_{1:L-1}\|^2 d\mu_0\right)^{\frac{1}{2}} \\
&\leq \left(\int_{\mathbb{R}^{m_L}} 1_{S_L}\|f_L \circ (f_{1:L-1} - g_{1:L-1})\|^2 d\mu_0\right)^{\frac{1}{2}} && \text{by triangle inequality} \\
&\quad + \left(\int_{\mathbb{R}^{m_L}} 1_{S_L}\|(f_L - g_L) \circ g_{1:L-1}\|^2 d\mu_0\right)^{\frac{1}{2}} \\
&\leq \left(\int_{\mathbb{R}^{m_L}} 1_{S_L}\|f_L \circ (f_{1:L-1} - g_{1:L-1})\|^2 d\mu_0\right)^{\frac{1}{2}} && \text{by construction of } \mu' \\
&\quad + \left(\int_{\mathbb{R}^{m_{L-1}}} \|f_L - g_L\|^2 d\mu'\right)^{\frac{1}{2}} \\
&\leq L_L \left(\int_{\mathbb{R}^m} 1_{S_L}\|f_{1:L-1} - g_{1:L-1}\|^2 d\mu_0\right)^{\frac{1}{2}} + \varepsilon && \text{by lipschitzian property, and induction hyp.} \\
&\leq L_L \left(\int_{\mathbb{R}^m} 1_{S_{L-1}}\|f_{1:L-1} - g_{1:L-1}\|^2 d\mu_0\right)^{\frac{1}{2}} + \varepsilon && \text{because } S_{L-1} \subseteq S_L \\
&\leq 1 \cdot (L-1)\varepsilon + \varepsilon && \text{using that } L_i \leq 1, \text{ and the induction hyp.} \\
&\leq L\varepsilon
\end{aligned}
$$

C Experimental settings details

C.1 Distributions

We consider the following probability distributions:

- A d-dimensional Gaussian distribution $\mathcal{N}(0, I_d)$,
- A d-dimensional Gaussian mixture
- A d-dimensional uniform distribution on the unit ball $B_d = \{x \in \mathbb{R}^d | \|x\|_2 \leq 1\}$,
- A d-dimensional uniform distribution on the unit cube $[0,1]^d$.

C.2 Compositions of functions

We consider the following functions that we will compose in different ways during the experiments:

- A linear function $f(x) = Ax + b$ with $A \in \mathbb{R}^{d \times n}$ and $b \in \mathbb{R}^n$,
- A quadratic function $f(x) = x^T Ax + b^T x + c$ with $A \in \mathbb{R}^{d \times d}$, $b \in \mathbb{R}^d$ and $c \in \mathbb{R}$,
- A trigonometric function $f(x) = \begin{cases} \sin(Ax + b) \\ \cos(Ax + b) \end{cases}$ with $A \in \mathbb{R}^{d \times n}$ and $b \in \mathbb{R}^n$,
- A radial basis function $f(x) = \exp(-\|x - c\|^2/\sigma^2)$ with $c \in \mathbb{R}^d$ and $\sigma > 0$,
- A sigmoidal function such that $\lim_{x \to -\infty} f(x) = 0$ and $\lim_{x \to +\infty}(x) = 1$.

D Comparison of the expressiveness of different neural networks

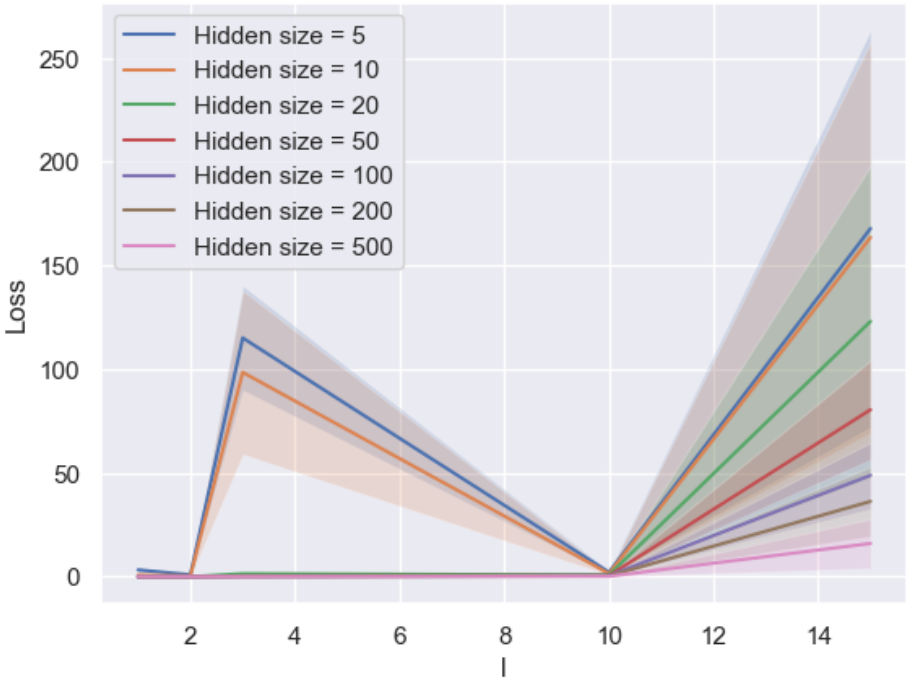

Figure 4: Validation loss for multiple composition of functions. The same function was approximated with the same prior distribution. The functions are picked randomly. After $l = 10$, the output distribution explodes so much that it becomes impossible to approximate by a neural network.

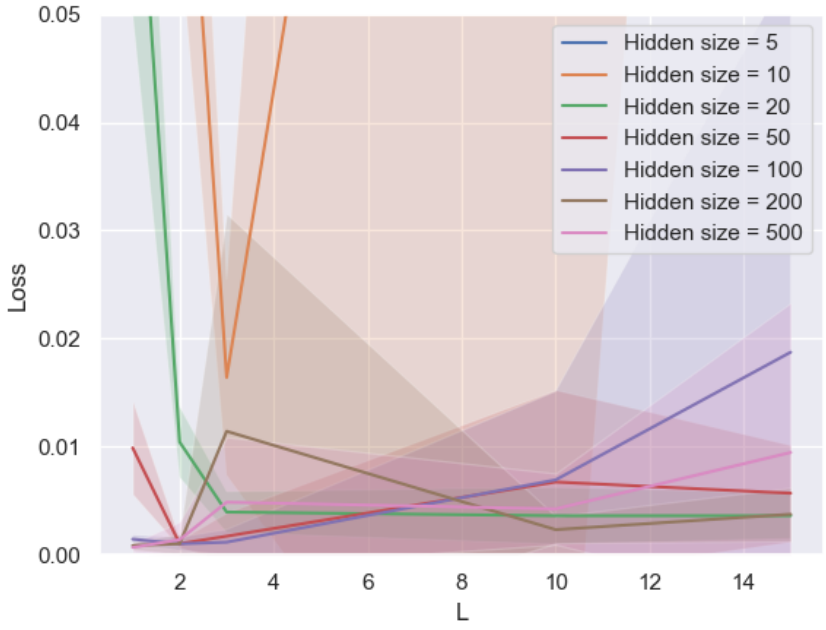

Figure 5: Validation loss for different layer of neural networks trying to approximate the same composition of 5 functions. The same function was approximated with the same prior distribution. The erratic behaviour of the loss highlights the difficulty of training depeer and larger neural networks.

E Comparing W_2 and loss for compositions of random functions

$Functions - l - d - output_{size}$	L	$hidden_{size}$	loss	W_2
linear-rbf-trigonometric-3-5-3	3	10	252.05	151.16
		50	52.43	33.89
		100	30.38	25.69
		500	25.83	23.91
	5	10	60.39	37.80
		50	30.27	25.57
		100	29.62	26.47
		500	16.96	15.64
	7	10	49.54	32.15
		50	30.07	25.96
		100	29.58	26.56
		500	29.30	27.37
linear-rbf-trigonometric-5-5-3	3	10	9090.21	8083.76
		50	380.40	260.46
		100	96.54	48.14
		500	11.48	12.27
	5	10	721.50	510.65
		50	47.88	16.86
		100	17.93	15.67
		500	9.46	11.42
	7	10	551.04	389.95
		50	67.37	26.18
		100	17.03	15.63
		500	9.01	10.78
linear-rbf-trigonometric-7-5-3	3	10	15609.91	14731.33
		50	565.63	510.33
		100	226.81	209.49
		500	6.05	5.55
	5	10	1127.68	875.77
		50	53.60	45.85
		100	15.18	11.79
		500	5.54	5.36
	7	10	855.30	698.30
		50	91.73	81.19
		100	14.25	11.27
		500	5.32	5.07

Table 1: Wasserstein-2 distance and validation loss for different compositions of functions and depths with a Gaussian prior

F Visualizing the approximations

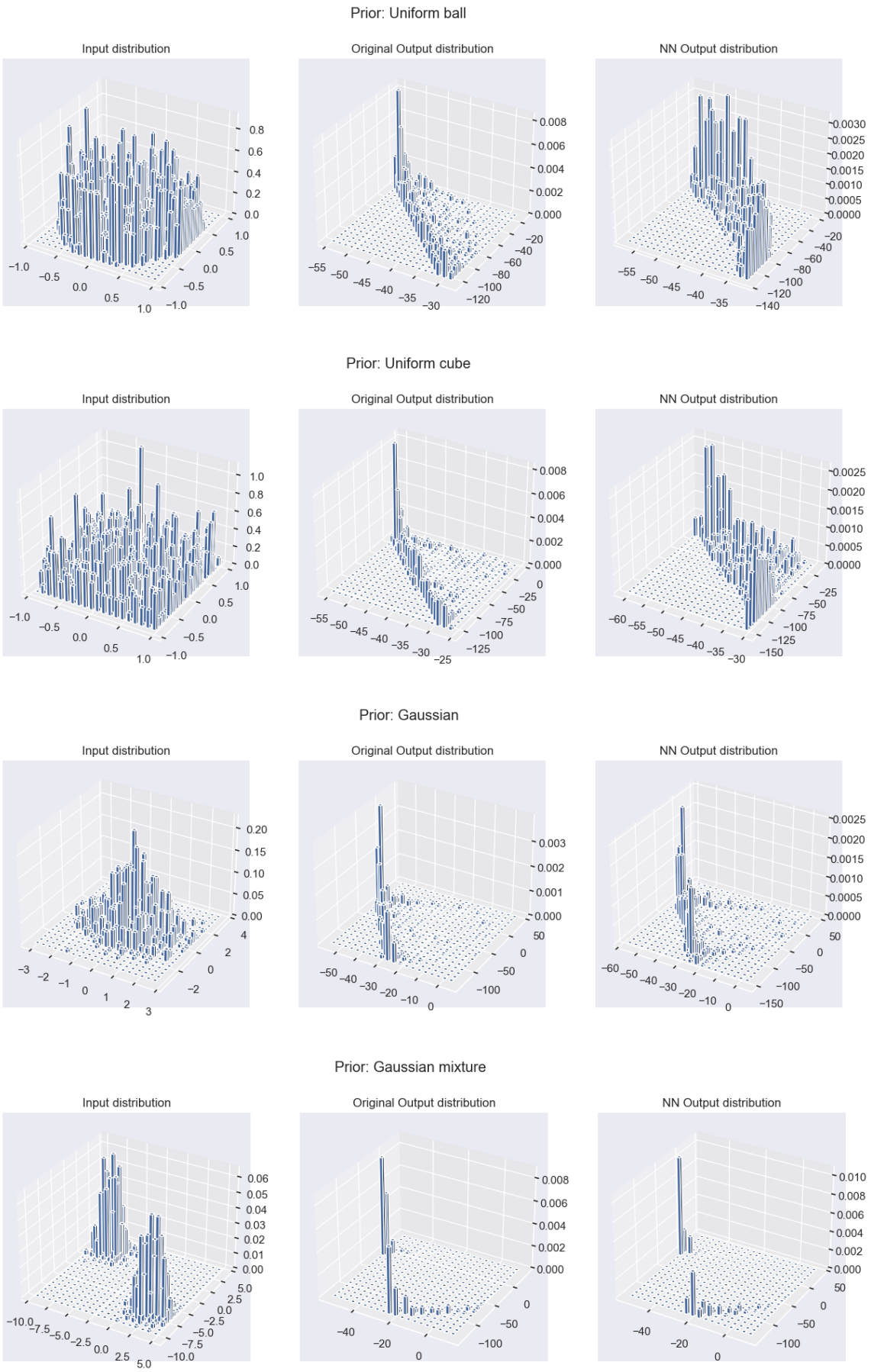

Figure 6: Approximation of a composition of 5 quadratic, linear and sigmoid functions for different priors and a 4-hidden layer neural network with 100 hidden nodes.

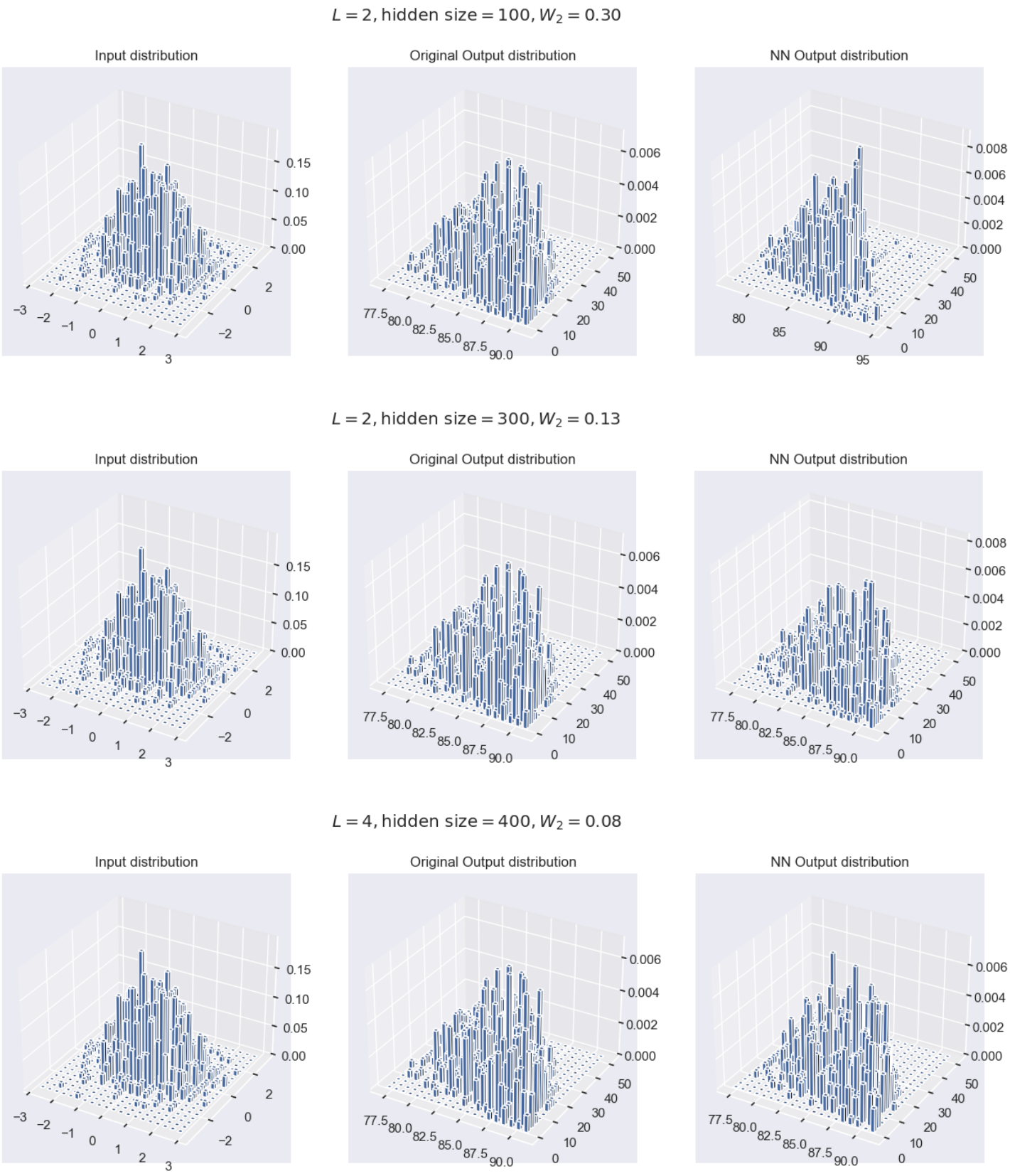

Figure 7: Comparison of the original output distribution with the approximated distribution for different neural network sizes. The input distribution is Gaussian and the function is a composition of 6 linear, sigmoid and trigonometric functions.

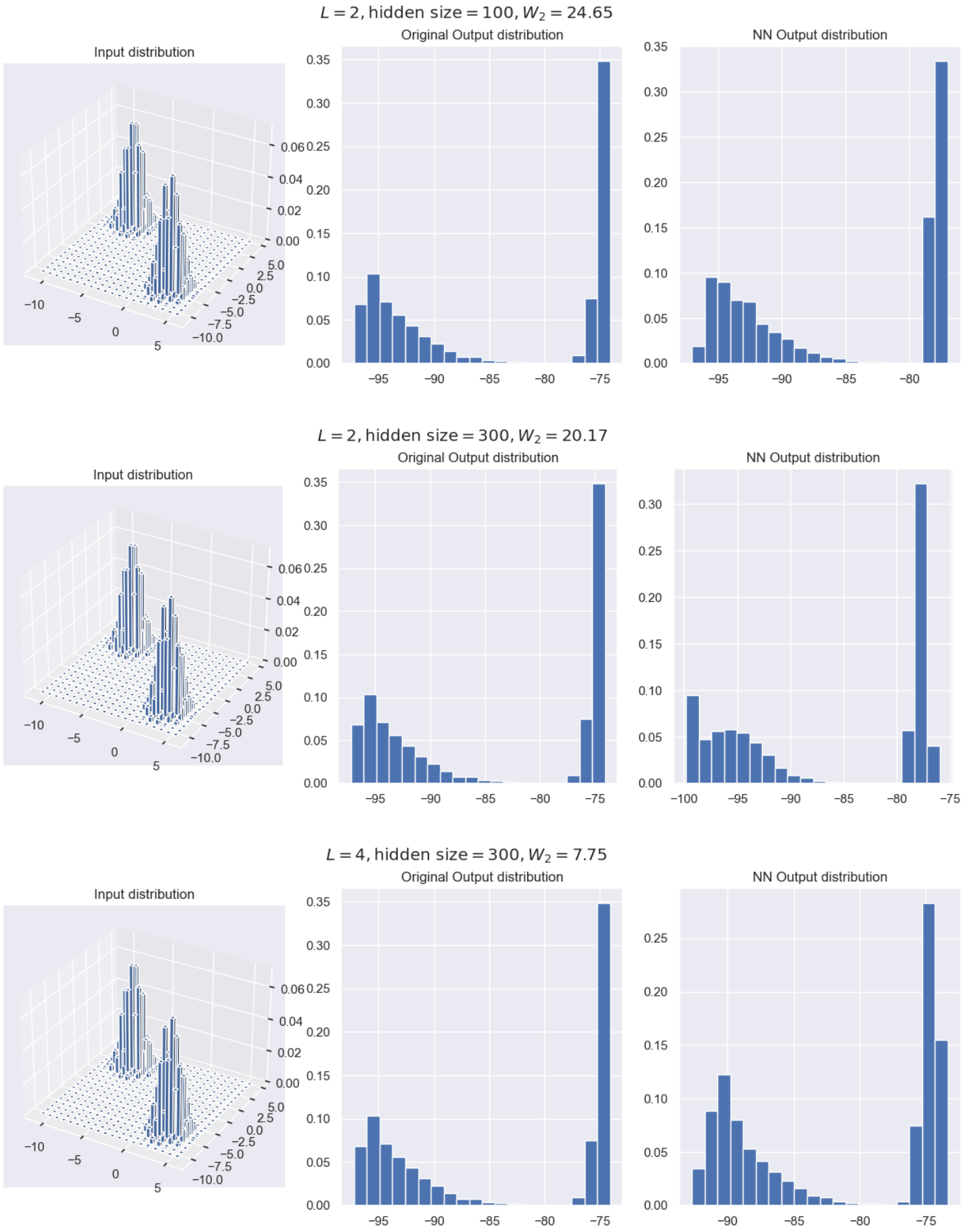

Figure 8: Comparison of the original output distribution with the approximated distribution for different neural network sizes. The input distribution is a mixture of gaussians and the function is a composition of 8 linear, sigmoid and trigonometric functions. The output space is one dimensionnal.