ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Seminar for Applied Mathematics SAM

Neural Networks for Singular Perturbations

J. A. A. Opschoor and Ch. Schwab and C. Xenophontos

Research Report No. 2024-03
January 2024

Seminar für Angewandte Mathematik
Eidgenössische Technische Hochschule
CH-8092 Zürich
Switzerland

Neural Networks for Singular Perturbations

J. A. A. Opschoor[1], Ch. Schwab[1*], C. Xenophontos[2]

[1]Seminar for Applied Mathematics, ETH Zürich, Rämistrasse 101, Zürich, 8092, Switzerland.
[2]Department of Mathematics and Statistics, University of Cyprus, P.O. BOX 20537, Nicosia, 1678, Cyprus.

*Corresponding author(s). E-mail(s): christoph.schwab@sam.math.ethz.ch;
Contributing authors: joost.opschoor@sam.math.ethz.ch; xenophontos.christos@ucy.ac.cy;

Abstract

We prove deep neural network (DNN for short) expressivity rate bounds for solution sets of a model class of singularly perturbed, elliptic two-point boundary value problems, in Sobolev norms, on the bounded interval $(-1,1)$. We assume that the given source term and reaction coefficient are analytic in $[-1,1]$.
We establish expression rate bounds in Sobolev norms in terms of the NN size which are uniform with respect to the singular perturbation parameter for several classes of DNN architectures. In particular, ReLU NNs, spiking NNs, and **tanh**- and sigmoid-activated NNs. The latter activations can represent "exponential boundary layer solution features" explicitly, in the last hidden layer of the DNN, i.e. in a shallow subnetwork, and afford improved robust expression rate bounds in terms of the NN size.
We prove that all DNN architectures allow *robust exponential solution expression* in so-called 'energy' as well as in 'balanced' Sobolev norms, for analytic input data.

Keywords: Singular Perturbations, Exponential Convergence, ReLU Neural Networks, Spiking Neural Networks, Tanh Neural Networks

MSC Classification: 34B08 , 34D15 , 65L11

1 Introduction

Singular perturbations are ubiquitous in engineering and in the sciences. Let us mention only solid mechanics (theory of thin solids, such as beams, plates and shells), fluid mechanics (viscous flows at large Reynolds number), and electromagnetics (eddy current problems in lossy media). In all these applications, PDEs depend on a small parameter $\varepsilon \in (0, 1]$ in physically relevant regimes of input data. Standard numerical approximation methods (Finite Volume, Finite Difference or Finite Element) generally do *not* perform uniformly w.r. to the physical perturbation parameter: in general, a so-called *scale resolution* condition relating the discretization parameters (such as the meshwidth h in Finite Element Methods) and ε needs to hold.

A common trait of singularly perturbed, *elliptic* PDE problems is an additive decomposition of PDE solutions into a regular (typically analytic) part and into singular components. See, e.g., [7] and the references there. The regular solution part u_ε^S may depend on the singular perturbation parameter ε, but derivatives of the smooth part u_ε^S satisfy bounds in Sobolev norms which are uniform in terms of ε. We refer to [7] and the references there for examples.

The singular perturbation part u_ε^{BL}, on the contrary, is not uniformly smooth in terms of ε. Its k-th derivative typically grows as $O(\varepsilon^{-k})$. Due to their exponential decay with respect to the distance to the boundary, they are referred to as *boundary layers*, and denoted herein by u_ε^{BL}.

1.1 Contributions

We prove that exponential boundary layer functions u_ε^{BL} which arise in asymptotic expansions of singularly perturbed elliptic boundary value problems can be expressed at exponential (w.r. to the NN size) rates by various DNN architectures *robustly*, i.e. *uniformly with respect to the perturbation parameter*, in various Sobolev norms. The DNNs considered are strict ReLU NNs (Propositions 5.2 and 5.3), spiking NNs (Theorem 6.6), and tanh-activated NNs (Theorem 7.4).

1.2 Layout

Section 2 introduces a model singularly perturbed reaction-diffusion two-point boundary value problem, specifies the assumptions on the problem data and recaps several results on the analytic regularity and the asymptotic behavior of its solutions.

Section 3 addresses mostly known FE approximation results, in particular featuring so-called *robust exponential convergence rates* for the parametric solutions.

Section 4 introduces assumptions on the architecture of the NNs.

Section 5 shows robust exponential convergence of strict ReLU NNs, i.e. neural networks with only ReLU activations: Proposition 5.2 states that emulation accuracy $\tau > 0$ in a so-called "balanced norm" can be achieved uniformly w.r. to the perturbation parameter $0 < \varepsilon \leq 1$ with a ReLU NN of size $O(|\log(\tau)|^2)$ and of depth $O(|\log(\tau)|(1 + |\log(|\log(\tau)|)|))$, i.e. the constants which are implicit in $O(\ )$ are independent of ε.

Section 6 establishes a corresponding conclusion for spiking NNs using a ReLU NN-to-spiking NN conversion algorithm from [23].

Assuming a constant reaction coefficient function, stronger robust exponential solution expression rate bounds by strict tanh NNs are proved in Section 7. We prove in Theorem 7.4 that to achieve expression error $\tau > 0$ in balanced norms, tanh-activated NNs of depth $O(\log|\log(\tau)|)$ and size $O(|\log(\tau)|)$ are sufficient. A corresponding result for the so-called sigmoid activation is shown in Appendix B.

Appendix A contains statement and proof of expression rate bounds for Chebyšev polynomials by strict tanh NNs which are used in various places and are of independent interest.

The generalization of this Chebyšev polynomial emulation to general smooth activation functions is established in Appendix B.

2 The Model Problem and its Regularity

Consider the following linear, singularly perturbed, reaction-diffusion boundary value problem (BVP): find $u_\varepsilon(x)$ such that

$$-\varepsilon^2 u_\varepsilon''(x) + b(x)u_\varepsilon(x) = f(x)\ ,\ x \in I = (-1,1)\ , \tag{2.1}$$

$$u_\varepsilon(\pm 1) = 0, \tag{2.2}$$

where $\varepsilon \in (0,1]$ is a small parameter that can approach zero, and $b(x), f(x)$ are given analytic functions on $\overline{I} = [-1,1]$, with $b(x) \geq \underline{b} > 0$ on $\overline{I}$ for some constant $\underline{b}$. Moreover, we assume there exist positive constants C_f, K_f, C_b, K_b such that $\forall\ n \in \mathbb{N}_0$, there holds

$$\|f^{(n)}\|_{L^\infty(I)} \leq C_f K_f^n n!\ ,\ \|b^{(n)}\|_{L^\infty(I)} \leq C_b K_b^n n!. \tag{2.3}$$

The above problem was studied in [13] where the following result was established.

Theorem 2.1 ([13, Thm. 1])**.** *For $0 < \varepsilon \leq 1$, there exists a unique solution $u_\varepsilon \in H_0^1(I)$ of (2.1)–(2.2). There exist positive constants C, K, independent of ε, such that*

$$\|u_\varepsilon^{(n)}\|_{L^2(I)} \leq CK^n \max\{n, \varepsilon^{-1}\}^n \ \forall\ n \in \mathbb{N}_0. \tag{2.4}$$

The above corresponds to classical differentiability and it is useful in the case when ε is large. If one uses the method of matched asymptotic expansions, a more refined regularity result can be obtained, as stated below.

Proposition 2.2 ([13])**.** *Let $u_\varepsilon \in H_0^1(I)$ be the solution of (2.1)–(2.2) and assume (2.3) holds. Then, u_ε may be decomposed as*

$$u_\varepsilon = u_\varepsilon^S + u_\varepsilon^{BL} + u_\varepsilon^R = u_\varepsilon^S + u_\varepsilon^+ + u_\varepsilon^- + u_\varepsilon^R, \tag{2.5}$$

where u_ε^S denotes the smooth part, $u_\varepsilon^\pm$ denote the boundary layers at the two end-points, and u_ε^R denotes the remainder. Furthermore, there exist positive constants $C_1, K_1, C_2, K_2, C_3, K_3$ independent of ε, such that

$$\left\| \left(u_\varepsilon^S\right)^{(n)} \right\|_{L^2(I)} \leq C_1 K_1^n n!, \qquad \textit{for all } n \in \mathbb{N}_0, \tag{2.6}$$

$$\left| \left(u_\varepsilon^\pm\right)^{(n)}(x) \right| \leq C_2 K_2^n e^{-\sqrt{\underline{b}}(1\mp x)/\varepsilon} \max\{n, \varepsilon^{-1}\}^n, \qquad \forall x \in \overline{I}, n \in \mathbb{N}_0, \tag{2.7}$$

$$\left\|\left(u_\varepsilon^R\right)^{(n)}\right\|_{L^2(I)} \le C_3\varepsilon^{2-n}e^{-K_3/\varepsilon}, \qquad \text{for all } n \in \{0,1,2\}. \tag{2.8}$$

Proof. This follows from the results in Section 2 of [13]. □

As can be seen from (2.4), the norms of the derivatives of u_ε may grow when $\varepsilon \to 0$. For u_ε and its first derivative, a more precise estimate of this ε-dependence is stated in the following lemma.

Lemma 2.3 ([14]). *Let u_ε be the solution u_ε of (2.1)–(2.2) and assume (2.3) holds. Then, there exists a constant $C > 0$, independent of ε, such that*

$$\|u_\varepsilon\|_{L^2(I)} \le C, \qquad \|u'_\varepsilon\|_{L^2(I)} \le C\varepsilon^{-1/2}, \qquad \|u_\varepsilon\|_{L^\infty(I)} \le C.$$

Proof. From Proposition 2.2, it follows that there exist constants $C_1, K_1 > 0$ independent of ε such that, for every $0 < \varepsilon \le 1$ and for every $n \in \mathbb{N}_0$ holds

$$\|(u_\varepsilon^S)^{(n)}\|_{L^2(I)} \le C_1K_1^n n!\,.$$

This can be combined with the interpolation inequality[1]

$$\|u_\varepsilon^S\|_{L^\infty(I)} \le C\|u_\varepsilon^S\|_{L^2(I)}^{1/2}\|u_\varepsilon^S\|_{H^1(I)}^{1/2}$$

to obtain $\|u_\varepsilon^S\|_{L^\infty(I)} \le C$, for some $C > 0$ independent of ε.

Similarly, $\|u_\varepsilon^R\|_{L^2(I)} \le C_3$ and $\|(u_\varepsilon^R)'\|_{L^2(I)} \le C_3$ imply that $\|u_\varepsilon^R\|_{L^\infty(I)} \le C$, for $C > 0$ independent of ε.

Finally, for the boundary layers we use [14, Equation (2.19)], which is a sharper bound in terms of ε than (2.7) in Proposition 2.2. It states that

$$\varepsilon^{-1/2}\|u_\varepsilon^\pm\|_{L^2(I)} + \varepsilon^{1/2}\|(u_\varepsilon^\pm)'\|_{L^2(I)} + \|u_\varepsilon^\pm\|_{L^\infty(I)} \le C. \tag{2.9}$$

Combining these estimates for the terms in (2.5) finishes the proof. □

Remark 2.4. *In the case of constant coefficients, i.e. $b(x) = b \in \mathbb{R}$, $b > 0$ in (2.1), the boundary layer parts of the solution may be explicitly obtained, as was the case in [22, Theorem 2.1]. Denote again by u_ε the solution of (2.1)–(2.2) and assume (2.3) holds.*

Then, u_ε may be decomposed as

$$\tilde{u}_\varepsilon = u_\varepsilon^S + \tilde{u}_\varepsilon^+ + \tilde{u}_\varepsilon^- + u_\varepsilon^R, \tag{2.10}$$

where u_ε^S and u_ε^R denote the smooth part and the remainder from Proposition 2.2 and

$$\tilde{u}_\varepsilon^\pm(x) = C^\pm e^{-\sqrt{b}(1\mp x)/\varepsilon}, \tag{2.11}$$

where the constants $C^\pm$ are bounded independently of ε (see [22] for more details). These boundary layer functions are related to $u_\varepsilon^\pm$ from Proposition 2.2 through $u_\varepsilon^+ + u_\varepsilon^- = u_\varepsilon^{BL} = \tilde{u}_\varepsilon^+ + \tilde{u}_\varepsilon^-$.

[1] Follows from Cauchy-Schwarz and $(u(x))^2 - (u(y))^2 = \int_y^x (u^2)'(\xi)d\xi$ for $-1 \le y < x \le 1$ and $u \in H^1(I)$.

3 *hp*-Approximation

The approximation to the solution of (2.1)–(2.2) by the Finite Element Method (FEM) was studied in [13], and in [22]. In this section we summarize the relevant results. First, we cast (2.1) – (2.2) into an equivalent weak formulation that reads: find $u_\varepsilon \in H_0^1(I)$ such that for all $v \in H_0^1(I)$ there holds

$$\int_0^1 \left\{\varepsilon^2 u_\varepsilon' v' + b u_\varepsilon v\right\} dx = \int_0^1 f v dx. \tag{3.1}$$

In order to define the discrete version of (3.1), for $N \in \mathbb{N}$, let $\Delta = \{x_j\}_{j=0}^N$ be an arbitrary partition of I and set $I_j = (x_{j-1}, x_j)$, $h_j = x_j - x_{j-1}, j = 1, \ldots, N$. With $\mathcal{P}_p(I)$ the space of polynomials of degree at most p on I, we define the spaces

$$\mathcal{S}^p(\Delta) := \{w \in H^1(I) : w|_{I_j} \in \mathcal{P}_p(I_j), j = 1, \ldots, N\}, \tag{3.2}$$

$$\mathcal{S}_0^p(\Delta) := \mathcal{S}^p(\Delta) \cap H_0^1(I). \tag{3.3}$$

The discrete version of (3.1), then reads: find $u_\varepsilon^{FEM} \in \mathcal{S}_0^p(\Delta)$ such that for all $v \in \mathcal{S}_0^p(\Delta)$, there holds

$$\int_0^1 \left\{\varepsilon^2 \left(u_\varepsilon^{FEM}\right)' v' + b u_\varepsilon^{FEM} v\right\} dx = \int_0^1 f v dx. \tag{3.4}$$

Associated with the above problem, we have the so-called *energy norm*:

$$\|w\|_\varepsilon^2 := \int_0^1 \left\{\varepsilon^2 (w')^2 + b w^2\right\} dx, \; w \in H_0^1(I), \tag{3.5}$$

and the usual best approximation property holds:

$$\|u_\varepsilon - u_\varepsilon^{FEM}\|_\varepsilon \le \|u_\varepsilon - v\|_\varepsilon \; \forall \, v \in \mathcal{S}_0^p(\Delta). \tag{3.6}$$

The following *spectral boundary layer mesh* is the minimal one which yields *exponential* convergence rates in terms of the number of degrees of freedom (i.e. the number of "Finite-Element features") as the polynomial degree p is increased.

Definition 3.1 (Spectral Boundary Layer mesh, [13, Definitions 13 and 14])**.** *For* $\kappa > 0$, $p \in \mathbb{N}$ *and* $0 < \varepsilon \le 1$, *the* Spectral Boundary Layer mesh $\Delta_{BL}(\kappa, p)$ *is defined as*

$$\Delta_{BL}(\kappa, p) := \begin{cases} \{-1, -1 + \kappa p \varepsilon, 1 - \kappa p \varepsilon, 1\} & \text{if } \kappa p \varepsilon < 1/2 \\ \{-1, 1\} & \text{if } \kappa p \varepsilon \ge 1/2. \end{cases}$$

Furthermore, let us define the spaces $V^p(\kappa)$ *and* $V_0^p(\kappa)$ *of piecewise polynomials of degree at most* p *via*

$$V^p(\kappa) := \mathcal{S}^p(\Delta_{BL}(\kappa, p)), \; V_0^p(\kappa) := \mathcal{S}_0^p(\Delta_{BL}(\kappa, p)) = V^p(\kappa) \cap H_0^1(I).$$

Using the above mesh, the following was shown in [13].

Proposition 3.2 ([13, Thm. 16]). *Assume that (2.3) holds and let* u_ε *be the solution of (3.1). Then, there exists* $\kappa_0 > 0$ *(depending only on* b *and* f*) such that for every* $\kappa \in (0, \kappa_0)$ *and* $p \in \mathbb{N}$ *there exist positive constants* C*,* β*, independent of* ε *and* p*, such that*

$$\inf_{v \in V_0^p(\kappa)} \|u_\varepsilon - v\|_\varepsilon \leq Ce^{-\beta p}. \tag{3.7}$$

For the ensuing deep NN approximation constructions, it is important to note that the proof of the above result is constructive, in that $v \in V_0^p(\kappa)$ can be taken to be the element-wise Gauß-Lobatto interpolant of u_ε. Hence, knowledge of the values of u_ε in the Gauß-Lobatto points in each (sub)interval of $\Delta_{BL}(\kappa, p)$ is the only required information for constructing v.

It is well known (see, e.g. [14] and the references therein) that the energy norm $\| \circ \|_\varepsilon$ defined in (3.5) is deficient in the sense that it does not "see the layers": as $\varepsilon \to 0$, it holds that

$$\|u_\varepsilon^S\|_\varepsilon = O(1) \text{ while } \|u_\varepsilon^\pm\|_\varepsilon = O(\varepsilon^{1/2}).$$

A correctly *balanced* norm should yield $\|u_\varepsilon^S\|_B = O(1) = \|u_\varepsilon^\pm\|_B$. The so-called *balanced norm* $\| \circ \|_B$ defined in the following expression is such a norm:

$$\|w\|_B^2 := \varepsilon \|w'\|_{L^2(I)}^2 + \|w\|_{L^2(I)}^2. \tag{3.8}$$

Unfortunately, the bilinear form associated with the weak formulation (3.1) is *not* coercive with respect to this norm, and standard numerical analysis techniques fail in proving exponential convergence with respect to this norm. In [14] this was by-passed through an alternative analysis (see [14] for details) and the following was shown.

Proposition 3.3 ([14, Thm. 2.6 and Cor. 2.7]). *Assume that (2.3) holds and let* u_ε *be the solution of (3.1).*

Then, there exists $\tilde{\kappa}_0 > 0$ *(depending only on* b *and* f*) such that for every* $\kappa \in (0, \tilde{\kappa}_0)$ *and every* $p \in \mathbb{N}$*, the following holds.*

Denoting by $u_\varepsilon^{FEM} \in V_0^p(\kappa)$ *the Galerkin Finite-Element solution of (3.4), there exist positive constants* C*,* β *that are independent of* ε *and* p*, such that*

$$\left\{ \varepsilon^{1/2} \|u_\varepsilon' - (u_\varepsilon^{FEM})'\|_{L^2(I)} + \|u_\varepsilon - u_\varepsilon^{FEM}\|_{L^2(I)} \right\} \leq Ce^{-\beta p}, \tag{3.9}$$

$$\|u_\varepsilon - u_\varepsilon^{FEM}\|_{L^\infty(I)} \leq Ce^{-\beta p}. \tag{3.10}$$

The proof of the above proposition is again constructive, as it is based on the proof of Proposition 3.2 (see [14]).

Finally, for the approximation of the explicit boundary layer expressions from Remark 2.4 we recall the following result from [22]. We state our result for the boundary layer function $\tilde{u}_\varepsilon^-(x) = \exp((1 + x)/e)$ for $x \in (-1, 1)$, corresponding to the left boundary point, and corresponding to $b = 1$ in Remark 2.4.

Proposition 3.4 ([22, Thm. 5.1, Cor. 5.1], [21, Thm. 3.74, Cor. 3.77])**.** *For* $\varepsilon \in (0,1]$ *and* $p \in \mathbb{N}$*, let the mesh* Δ *be as follows:*

$$\Delta = \begin{cases} \{-1, -1+\kappa\tilde{p}\varepsilon, 1\}, & \text{if } \kappa\tilde{p}\varepsilon < 2, \\ \{-1, 1\}, & \text{if } \kappa\tilde{p}\varepsilon \geq 2, \end{cases} \tag{3.11}$$

for $\tilde{p} := p + \frac{1}{2}$ *and constants* $0 < \kappa_1$ *and* $\kappa_1 \leq \kappa < 4/e =: \kappa_0$ *which are independent of* p *and* ε*.*

Then, with $\tilde{u}^-_\varepsilon(x) = \exp((1+x)/e)$ *for* $x \in (-1,1)$ *as defined in Remark 2.4 with* $b = 1$*, there exists* $v \in \mathcal{S}^p(\Delta)$ *with* $v(\pm 1) = \tilde{u}^-_\varepsilon(\pm 1)$ *and*

$$\varepsilon^{1/2} \left\| (\tilde{u}^-_\varepsilon)' - v' \right\|_{L^2(I)} + \varepsilon^{-1/2} \left\| \tilde{u}^-_\varepsilon - v \right\|_{L^2(I)} + \left\| \tilde{u}^-_\varepsilon - v \right\|_{L^\infty(I)} \leq C \exp(-\beta p), \tag{3.12}$$

for constants $C, \beta > 0$ *independent of* p *and* ε*.*

Remark 3.5. *In [21, 22], for the case* $\kappa\tilde{p}\varepsilon < 2$*, it is shown to be sufficient to use polynomial degree* 1 *in the element* $(-1+\kappa\tilde{p}\varepsilon, 1)$*. Also,* β *is specified explicitly. For the case* $\kappa\tilde{p}\varepsilon \geq 2$*, it is shown that the error converges faster than exponentially. There exists* $C > 0$ *such that the error bounds also hold if we replace* $C\exp(-\beta p)$ *by* $C\exp(-\tilde{p}\log(2\tilde{p}\varepsilon/e))$*.*

4 Neural Network Definitions

As usual (e.g. [16–18]), we define a neural network (NN) in terms of its weight matrices and bias vectors. We distinguish between a neural network and the function it realizes, called *realization* of the NN, which is the composition of parameter-dependent affine transformations and nonlinear activations. We recall some NN formalism in the notation of [18, Section 2].

Definition 4.1 ([18, Definition 2.1])**.** *For* $d, L \in \mathbb{N}$*, a* neural network Φ *with input dimension* $d \geq 1$ *and number of layers* $L \geq 1$*, comprises a finite sequence of matrix-vector tuples, i.e.*

$$\Phi = ((A_1, b_1), (A_2, b_2), \ldots, (A_L, b_L)).$$

For $N_0 := d$ *and* numbers of neurons $N_1, \ldots, N_L \in \mathbb{N}$ per layer*, for all* $\ell = 1, \ldots, L$ *it holds that* $A_\ell \in \mathbb{R}^{N_\ell \times N_{\ell-1}}$ *and* $b_\ell \in \mathbb{R}^{N_\ell}$*.*

For a NN Φ *and an activation function* $\varrho : \mathbb{R} \to \mathbb{R}$*, we define the associated* realization *of* Φ *as the function*

$$\mathrm{R}(\Phi) : \mathbb{R}^d \to \mathbb{R}^{N_L} : x \to x_L,$$

where

$$\begin{aligned} x_0 &:= x, \\ x_\ell &:= \varrho(A_\ell x_{\ell-1} + b_\ell), \qquad \text{for } \ell = 1, \ldots, L-1, \end{aligned}$$

$$x_L := A_L x_{L-1} + b_L.$$

Here ϱ acts componentwise on vector-valued inputs, $\varrho(y) = (\varrho(y_1), \dots, \varrho(y_m))$ for all $y = (y_1, \dots, y_m) \in \mathbb{R}^m$. We call the layers indexed by $\ell = 1, \dots, L-1$ hidden layers, *in those layers the activation function is applied. No activation is applied in the last layer of the NN.*

We refer to $L(\Phi) := L$ as the depth *of Φ and call $M(\Phi) := \sum_{\ell=1}^{L} \|A_\ell\|_0 + \|b_\ell\|_0$ the* size *of Φ, which is the number of nonzero components in the weight matrices A_ℓ and the bias vectors b_ℓ. Furthermore, we call d and N_L the* input dimension *and the* output dimension.

Some related works, e.g. [4], use the width as a measure for the complexity of a NN, which is defined as $\max_{\ell=0}^{L} N_\ell$. Note that in each layer of a fully connected NN the number of nonzero weights can be as large as the width squared.

We will refer to NNs with only activation function ϱ as strict ϱ-NNs, or simply as ϱ-NNs. This includes NNs of depth 1, which do not have hidden layers and which exactly realize affine transformations.

5 ReLU Neural Network Approximations

In this section, we consider the approximation of univariate functions on bounded intervals by neural networks with the ReLU activation function $\rho : \mathbb{R} \to \mathbb{R} : x \mapsto \max\{0, x\}$. In Proposition 5.1 below, we recall the ReLU NN approximation of continuous, piecewise polynomial functions from [17, Proposition 3.11].[2] It allows us to transfer the finite element approximation results from Section 3 and obtain approximation rate bounds for ReLU NNs in Propositions 5.2 and 5.3.

Proposition 5.1 ([17, Proposition 3.11]). *For $-\infty < a < b < \infty$, let $I := (a, b)$. For all $N \in \mathbb{N}$, all $p \in \mathbb{N}$, all partitions $\Delta = \{x_j\}_{j=0}^{N}$ of I into N open, disjoint, connected subintervals $I_j = (x_{j-1}, x_j)$ of length $h_j = x_j - x_{j-1}$, $j = 1, \dots, N$, $h = \max_{j=1}^{N} h_j$, and for all $v \in \mathcal{S}^p(\Delta)$,* [3] *for all relative tolerances $\tau \in (0, 1)$ there exists a ReLU NN $\Phi_\tau^{v,\Delta,p}$ such that for all $1 \le r, r' \le \infty$ there holds*

$$(2/h_i)^{1-t} \left| v - \mathrm{R}\left(\Phi_\tau^{v,\Delta,p}\right) \right|_{W^{t,r}(I_i)} \le \tau \tfrac{1}{2} (2/h_i)^{1+1/r'-1/r} \min_{\substack{u \in \mathcal{P}_p: \\ u'' = v''|_{I_i}}} \|u\|_{L^{r'}(I_i)}, \tag{5.1}$$

for all $i = 1, \dots, N$ and $t = 0, 1$,

$$\tfrac{1}{h} \left\| v - \mathrm{R}\left(\Phi_\tau^{v,\Delta,p}\right) \right\|_{L^r(I)} \le \left| v - \mathrm{R}\left(\Phi_\tau^{v,\Delta,p}\right) \right|_{W^{1,r}(I)} \le \tfrac{1}{2} \tau \left| v \right|_{W^{1,r}(I)}, \tag{5.2}$$

$$\begin{aligned} L\left(\Phi_\tau^{v,\Delta,p}\right) &\le C(1 + \log_2(p)) \log_2(1/\tau) + C(1 + \log_2(p))^3, \\ M\left(\Phi_\tau^{v,\Delta,p}\right) &\le CNp\big(1 + \log_2(1/\tau) + \log_2(p)\big), \end{aligned}$$

[2] The result in [17, Proposition 3.11] is stated for different polynomial degrees $p_1, \dots, p_N \in \mathbb{N}$ in the elements $I_1, \dots, I_N$ of the partition. Here, we only state that result for the special case that $p_1 = \dots = p_N = p \in \mathbb{N}$.

[3] The definition of $\mathcal{S}^p(\Delta)$ in (3.2) also applies to general intervals $I = (a, b)$ instead of $I = (-1, 1)$.

for a constant $C > 0$ which is independent of I, N, p, Δ, τ and v.

In addition, it holds that $\mathrm{R}\left(\Phi_\tau^{v,\Delta,p}\right)(x_j) = v(x_j)$ for all $j \in \{0,\dots,N\}$. The weights and biases in the hidden layers are independent of v. The weights and biases in the output layer are linear combinations of the function values of v in the Clenshaw–Curtis points in I_i for $i = 1,\dots,N$.

As a direct corollary of Propositions 3.3 and 5.1 we obtain:

Proposition 5.2. *Assume that (2.3) holds. For $\varepsilon \in (0,1]$, let u_ε be the solution of (3.1).*

Then, there exists $\tilde{\kappa}_0 > 0$ (depending only on b and f) such that for every $\kappa \in (0,\tilde{\kappa}_0)$ and $p \in \mathbb{N}$ there exists a ReLU NN $\Phi_\varepsilon^{FEM,\kappa,p}$ such that, with positive constants C, β, independent of ε and p, it holds that

$$\left\{\varepsilon^{1/2}\|u_\varepsilon' - \mathrm{R}(\Phi_\varepsilon^{FEM,\kappa,p})'\|_{L^2(I)} + \|u_\varepsilon - \mathrm{R}(\Phi_\varepsilon^{FEM,\kappa,p})\|_{L^2(I)}\right\} \le Ce^{-\beta p}, \tag{5.3}$$

$$\|u_\varepsilon - \mathrm{R}(\Phi_\varepsilon^{FEM,\kappa,p})\|_{L^\infty(I)} \le Ce^{-\beta p}, \tag{5.4}$$

and $\mathrm{R}(\Phi_\varepsilon^{FEM,\kappa,p})(\pm 1) = 0$.

For a constant $\tilde{C} = \tilde{C}(\beta) > 0$ depending only on β, the network depth and size are bounded as follows:

$$L\left(\Phi_\varepsilon^{FEM,\kappa,p}\right) \le \tilde{C}p(1+\log_2(p)), \qquad M\left(\Phi_\varepsilon^{FEM,\kappa,p}\right) \le \tilde{C}p^2. \tag{5.5}$$

The weights and biases in the hidden layers are independent of u_ε and depend only on κ, p, ε and β.

Proof. We apply Proposition 5.1 to $u_\varepsilon^{FEM} \in V_0^p(\kappa)$ from Proposition 3.3, with accuracy parameter $\tau = e^{-\beta p}$ for β given in Proposition 3.3 and the Spectral Boundary Layer mesh $\Delta := \Delta_{BL}(\kappa,p)$ from Definition 3.1, i.e. if $\kappa p \varepsilon < 1/2$, then the number of elements is $N = 3$, whereas if $\kappa p\varepsilon \ge 1/2$, then $N = 1$. We define $\Phi_\varepsilon^{FEM,\kappa,p} := \Phi_\tau^{u_\varepsilon^{FEM},\Delta,p}$. We obtain from (5.1), with $t = 0$, $r = r' \in \{2,\infty\}$ and $u = v$, that on all elements $I_j \in \Delta_{BL}(\kappa,p)$, $j = 1,\dots,N$, it holds that $\left\|u_\varepsilon^{FEM} - \mathrm{R}(\Phi_\varepsilon^{FEM,\kappa,p})\right\|_{L^r(I_j)} \le \frac{1}{2}\tau\left\|u_\varepsilon^{FEM}\right\|_{L^r(I_j)}$, and thus

$$\begin{aligned}\left\|u_\varepsilon^{FEM} - \mathrm{R}(\Phi_\varepsilon^{FEM,\kappa,p})\right\|_{L^2(I)} &\le \tfrac{1}{2}\tau\left\|u_\varepsilon^{FEM}\right\|_{L^2(I)} \\ &\le \tfrac{1}{2}\tau\left(\|u_\varepsilon\|_{L^2(I)} + \left\|u_\varepsilon - u_\varepsilon^{FEM}\right\|_{L^2(I)}\right) \\ &\le \tfrac{1}{2}e^{-\beta p}\left(C + Ce^{-\beta p}\right) \le Ce^{-\beta p},\end{aligned}$$

where we used Lemma 2.3 and (3.9) in the third step. Here, and in the remainder of the proof, β is as in Proposition 3.3 and C denotes a generic positive constant which is independent of ε and p, but may be different at each appearance. We have the same result as above, also in the maximum norm:

$$\left\|u_\varepsilon^{FEM} - \mathrm{R}(\Phi_\varepsilon^{FEM,\kappa,p})\right\|_{L^\infty(I)} \le \tfrac{1}{2}e^{-\beta p}\left(\|u_\varepsilon\|_{L^\infty(I)} + Ce^{-\beta p}\right) \le Ce^{-\beta p}.$$

From (5.2) we obtain

$$\left\|(u_\varepsilon^{FEM})' - \mathrm{R}(\Phi_\varepsilon^{FEM,\kappa,p})'\right\|_{L^2(I)} \leq \tfrac{1}{2}\tau \left\|(u_\varepsilon^{FEM})'\right\|_{L^2(I)}.$$

Combined with Lemma 2.3 and (3.9), this gives

$$\begin{aligned}\left\|(u_\varepsilon^{FEM})' - \mathrm{R}(\Phi_\varepsilon^{FEM,\kappa,p})'\right\|_{L^2(I)} &\leq \tfrac{1}{2}\tau \left\|(u_\varepsilon^{FEM})'\right\|_{L^2(I)} \\ &\leq \tfrac{1}{2}\tau \left(\|u_\varepsilon'\|_{L^2(I)} + \left\|u_\varepsilon' - (u_\varepsilon^{FEM})'\right\|_{L^2(I)}\right) \\ &\leq \tfrac{1}{2}e^{-\beta p}\left(C\varepsilon^{-1/2} + C\varepsilon^{-1/2}e^{-\beta p}\right) \\ &\leq C\varepsilon^{-1/2}e^{-\beta p}.\end{aligned}$$

Using the triangle inequality to combine these estimates with Equations (3.9)–(3.10) finishes the proof of Equations (5.3)–(5.4). By Proposition 5.1, it also holds that $\mathrm{R}(\Phi_\varepsilon^{FEM,\kappa,p})(\pm 1) = u_\varepsilon^{FEM}(\pm 1) = 0 = u_\varepsilon(\pm 1)$.

As upper bounds on the network depth and size, we obtain from Proposition 5.1

$$\begin{aligned} L\left(\Phi_\varepsilon^{FEM,\kappa,p}\right) &\leq C(1+\log_2(p))\log_2(1/\tau) + C(1+\log_2(p))^3 \\ &\leq \tilde{C}p(1+\log_2(p)), \\ M\left(\Phi_\varepsilon^{FEM,\kappa,p}\right) &\leq CNp\big(1+\log_2(1/\tau)+\log_2(p)\big) \\ &\leq \tilde{C}p^2, \end{aligned}$$

for $\tilde{C} = \tilde{C}(\beta) > 0$ depending only on β. In the last step, we used that $N \leq 3$. ☐

By the same arguments as in the proof of Proposition 5.2, we obtain from Propositions 3.4 and 5.1 the following result on the approximation of exponential boundary layer functions.

Proposition 5.3. *There exists $\tilde{\kappa}_1 > 0$ such that for every $\kappa \in (\tilde{\kappa}_1, 4/e)$ and $p \in \mathbb{N}$ there exists a ReLU NN $\Phi_\varepsilon^{\exp,\kappa,p}$ such that, with positive constants C, β, independent of ε and p, it holds that*

$$\varepsilon^{1/2}\| -\exp(-\cdot/\varepsilon)/\varepsilon - \mathrm{R}(\Phi_\varepsilon^{\exp,\kappa,p})'\|_{L^2((0,1))} \leq Ce^{-\beta p}, \tag{5.6}$$

$$\|\exp(-\cdot/\varepsilon) - \mathrm{R}(\Phi_\varepsilon^{\exp,\kappa,p})\|_{L^2((0,1))} \leq Ce^{-\beta p}, \tag{5.7}$$

$$\|\exp(-\cdot/\varepsilon) - \mathrm{R}(\Phi_\varepsilon^{\exp,\kappa,p})\|_{L^\infty((0,1))} \leq Ce^{-\beta p}. \tag{5.8}$$

For a constant $\tilde{C} = \tilde{C}(\beta) > 0$ depending only on β, the NN depth and size are bounded as follows:

$$L\left(\Phi_\varepsilon^{\exp,\kappa,p}\right) \leq \tilde{C}p(1+\log_2(p)), \qquad M\left(\Phi_\varepsilon^{\exp,\kappa,p}\right) \leq \tilde{C}p^2. \tag{5.9}$$

Proof. Let $\tilde{u}_{2\varepsilon}^-$ and v be as in Proposition 3.4, with 2ε in place of ε. Composing both $\tilde{u}_{2\varepsilon}^-$ and v with the affine transformation $P : [0,1] \to [-1,1] : x \mapsto 2x-1$ gives

$\exp(-x/\varepsilon) = \tilde{u}^-_{2\varepsilon} \circ P(x)$ for all $x \in (0,1)$ and, for $r = 2, \infty$,

$$\begin{aligned} \|\exp(-x/\varepsilon) - v \circ P\|_{L^r((0,1))} &\le \|\tilde{u}^-_{2\varepsilon} - v\|_{L^r((-1,1))} \le C\exp(-\beta p), \\ \varepsilon^{1/2}\| - \tfrac{1}{\varepsilon}\exp(-x/\varepsilon) - (v \circ P)'\|_{L^2((0,1))} &\le \varepsilon^{1/2}\|(\tilde{u}^-_{2\varepsilon})' - v'\|_{L^2((-1,1))}\|P'\|_{L^\infty((0,1))} \\ &\le C\exp(-\beta p) \cdot 2 = C\exp(-\beta p). \end{aligned}$$

Now, we can apply Proposition 5.1 to $v \circ P$ and $\tilde{\Delta} = \{0, \kappa\tilde{p}\varepsilon/2, 1\}$ with accuracy $\tau = \exp(-\beta p)$ to obtain the existence of a ReLU NN $\Phi^{\exp,\kappa,p}_\varepsilon := \Phi^{v\circ P, \tilde{\Delta}, p}_\tau$ which satisfies for $r = 2, \infty$

$$\begin{aligned} \|v \circ P - \mathrm{R}(\Phi^{\exp,\kappa,p}_\varepsilon)\|_{L^r((0,1))} &\le \tfrac{1}{2}\exp(-\beta p)\|v \circ P\|_{L^r((0,1))}, \\ \|(v \circ P)' - \mathrm{R}(\Phi^{\exp,\kappa,p}_\varepsilon)'\|_{L^2((0,1))} &\le \tfrac{1}{2}\exp(-\beta p)\|(v \circ P)'\|_{L^2((0,1))}, \end{aligned}$$

from which we obtain the desired error bounds using the same arguments as in the proof of Proposition 5.2, using (2.9).

We also find the bounds on the network depth and size as in Proposition 5.2:

$$\begin{aligned} L\left(\Phi^{\exp,\kappa,p}_\varepsilon\right) &\le C(1 + \log_2(p))\log_2(1/\tau) + C(1 + \log_2(p))^3 \le \tilde{C}p(1 + \log_2(p)), \\ M\left(\Phi^{\exp,\kappa,p}_\varepsilon\right) &\le CNp\big(1 + \log_2(1/\tau) + \log_2(p)\big) \le \tilde{C}p^2, \end{aligned}$$

for $\tilde{C} = \tilde{C}(\beta) > 0$ depending only on β. □

6 Spiking Neural Network Approximation

So far, we obtained expression rate bounds of strict ReLU NNs where all activations are ReLUs, for solutions of the singularly perturbed, two-point boundary value problem (2.1)–(2.2). Approximation rates for ReLU NNs transfer to so-called *spiking neural networks* (SNNs) which are at the core of some models of so-called *neuromorphic computing* (see, e.g., [23] and the references there). Results in this direction go back several decades, see e.g. [9, 10] and the references there. More recently, Algorithms 1 and 2 in [23] produce for every strict ReLU NN Φ a SNN $\mathrm{S}(\Phi)$ whose realization $\mathrm{R}(\mathrm{S}(\Phi))$ is identical to the input-output map $x \mapsto \mathrm{R}(\Phi)(x)$ of the ReLU NN Φ up to an affine transformation of the input, see Proposition 6.5 for details.

We use [23, Alg. 1 and 2] to deduce from the approximation rate bounds in Section 5 corresponding results for SNNs in terms of the number of nonzero weights in the SNN. We proceed as follows. After defining SNNs, we recall the exact mapping from ReLU NNs to SNNs in Algorithms 6.1 and 6.2 below. In Proposition 6.5 we estimate the size of the resulting SNN in terms of the size of the ReLU NN. SNN approximation of solutions to the singularly perturbed model problem in Section 2 is the topic of Theorem 6.6.

6.1 Spiking Neural Network Definitions

As in [23], we consider SNNs with *integrate-and-fire neurons* in the hidden layers, in which each hidden layer neuron fires exactly once during each evaluation of the network. The output of a hidden layer neuron i in layer ℓ is the spiking time $(t_\ell)_i \in [t_\ell^{\min}, t_\ell^{\max}]$. The spiking time is defined (and computed) as the first time $t \geq t_\ell^{\min}$ at which the *voltage trajectory* $(V_\ell)_i(t)$ of neuron i in layer ℓ attains the threshold value $(\vartheta_\ell)_i$, as detailed in the following definition. The output layer consists of *integration neurons*, which do not fire. The output of each such neuron i in the last layer is the voltage at the final time $(V_L)_i(t_L^{\max})$.

Definition 6.1 (Spiking neural network (SNN))**.** *([23, Section 2.1]) For $d, L \in \mathbb{N}$, a* spiking neural network Φ *with input dimension $d \geq 1$ and number of layers $L \geq 1$, is given by a finite sequence of matrix-vector-vector-number-number tuples, i.e.*

$$\Phi = \big((J_1, \vartheta_1, \alpha_1, t_1^{\min}, t_1^{\max}), \ldots, (J_{L-1}, \vartheta_{L-1}, \alpha_{L-1}, t_{L-1}^{\min}, t_{L-1}^{\max}),\\ (J_L, \alpha_L, t_L^{\min}, t_L^{\max})\big),$$

where in the last tuple, the vector ϑ_L is omitted. For $N_0 := d$ and numbers of neurons $N_1, \ldots, N_L \in \mathbb{N}$ per layer, *for all $\ell = 1, \ldots, L$ it holds that $J_\ell \in \mathbb{R}^{N_\ell \times N_{\ell-1}}$, $\vartheta_\ell, \alpha_\ell \in \mathbb{R}^{N_\ell}$ and $t_\ell^{\min}, t_\ell^{\max} \in \mathbb{R}$, with the exception that we do not consider ϑ_L. In addition, we require that $t_{\ell-1}^{\max} = t_\ell^{\min}$ for all $\ell = 1, \ldots, L$ and that*

$$0 = t_0^{\min} < t_1^{\min} = 1 < \cdots < t_L^{\min} \quad \textit{and} \quad t_0^{\max} = 1 < t_1^{\max} < \cdots < t_{L-1}^{\max} = t_L^{\max}.$$

The input of Φ comprises the firing times $t_0 \in [t_0^{\min}, t_0^{\max}]^d$ of the neurons in the input layer. For all $\ell = 1, \ldots, L-1$ and $i = 1, \ldots, N_\ell$, the spiking time $(t_\ell)_i \in [t_\ell^{\min}, t_\ell^{\max}]$ of neuron i in layer ℓ is defined as the first time $t \geq t_\ell^{\min}$ at which the voltage trajectory $(V_\ell)_i(t)$ *attains or exceeds the* threshold $(\vartheta_\ell)_i \in \mathbb{R}$, *where $(V_\ell)_i(t)$ is defined by $(V_\ell)_i(t_{\ell-1}^{\min}) = 0$ and the following ODE, which holds for all $t \in (t_{\ell-1}^{\min}, t_\ell^{\max})$:*

$$\tfrac{\mathrm{d}}{\mathrm{d}t}(V_\ell)_i(t) = (\alpha_\ell)_i H(t - t_{\ell-1}^{\min}) + \sum_{j=1}^{N_{\ell-1}} (J_\ell)_{ij} H(t - (t_{\ell-1})_j) + (I_\ell)_i(t). \tag{6.1}$$

Here, $H : \mathbb{R} \to \mathbb{R}$ denotes the Heaviside function, defined by $H(x) = 1$ for $x > 0$ and $H(x) = 0$ else. The values $(J_\ell)_{ij}$ are called weights *and $(\alpha_\ell)_i$ is the* slope parameter*. In layers $\ell = 1, \ldots, L-1$, a nonnegative short* pulse *$(I_\ell)_i(t)$ is used to force the neuron to spike at the latest at $t_\ell^{\max}$. In the output layer $\ell = L$, the voltage trajectory is also defined by (6.1), with $I_L \equiv 0$. The output of Φ comprises the voltages of the neurons in the output layer at time $t_L^{\max} = t_{L-1}^{\max}$, which we denote by $\mathrm{R}(\Phi)(t_0) := (V_L)(t_L^{\max}) \in \mathbb{R}^{N_L}$.*

We refer to $L(\Phi) := L$ as the depth *of Φ and call $M(\Phi) := \sum_{\ell=1}^{L} \|J_\ell\|_0$ the* size *of Φ, which is the number of nonzero components in the weight matrices J_ℓ. Furthermore, we call d and N_L the* input dimension *and the* output dimension*.*

In [23], for neuron $i = 1, \ldots, N_\ell$ in hidden layer $\ell = 1, \ldots, L-1$, the pulse is defined in terms of the Dirac delta distribution as $(I_\ell)_i(t) = R\delta(t - t_\ell^{\max})$ for some sufficiently large $R > 0$. Denoting by $(\tilde{V}_\ell)_i$ the voltage trajectory in case $(I_\ell)_i \equiv 0$, it is sufficient to set $R = (\vartheta_\ell)_i - (\tilde{V}_\ell)_i(t_\ell^{\max})$.

Remark 6.2. *For any $0 < \eta \le t_\ell^{\max} - t_\ell^{\min}$ we could equivalently consider the current pulse*

$$(I_\ell)_i(t) = \begin{cases} 0 & \text{for } t \in (t_{\ell-1}^{\min}, t_\ell^{\max} - \eta), \\ \big((\vartheta_\ell)_i - (\tilde{V}_\ell)_i(t_\ell^{\max})\big)/\eta & \text{for } t \in (t_\ell^{\max} - \eta, t_\ell^{\max}), \end{cases}$$

such that $(V_\ell)_i(t_\ell^{\max}) = (\vartheta_\ell)_i$. To ensure that the SNN does not fire earlier than at time $t_\ell^{\max}$, it suffices to choose η small enough, e.g. such that $\big((\vartheta_\ell)_i - (\tilde{V}_\ell)_i(t_\ell^{\max})\big)/\eta > \max_{t \in [t_\ell^{\max} - \eta, t_\ell^{\max}]} |(\tilde{V}_\ell)_i'(t)|$. From this we obtain that $(V_\ell)_i'(t) > 0$ for all $t \in (t_\ell^{\max} - \eta, t_\ell^{\max})$, and thus that $(V_\ell)_i(t) < (V_\ell)_i(t_\ell^{\max})$ for all such t. For all $t < t_\ell^{\max} - \eta$, the pulse does not affect $(\tilde{V}_\ell)_i(t)$, hence also for such t the SNN does not spike.

Remark 6.3. *Imposing $(t_\ell)_i \ge t_\ell^{\min}$ is important. Although the voltage trajectory $(V_\ell)_i(t)$ may attain or exceed the threshold value $(\vartheta_\ell)_i$ at an earlier time, we do not want the neuron to fire earlier than $t_\ell^{\min}$. In [23], this is interpreted as using a time-dependent threshold, which equals the previously specified value for $t \ge t_\ell^{\min}$, and a very large value for $t < t_\ell^{\min}$.*

6.2 ReLU to Spiking Neural Network Conversion

Next, we state a version of [23, Algorithms 1 and 2] for transforming feedforward ReLU networks. For each ReLU NN, the SNN produced by these algorithms has the same input dimension, output dimension, depth and the same layer dimensions, see Proposition 6.5. A large output value of a neuron from the ReLU NN corresponds to early spiking of the corresponding SNN neuron.

We have slightly modified line 15 from [23, Algorithm 1] to define an exact mapping from ReLU NNs to spiking NNs without making use of a training data set. See Remark 6.4 below.

Remark 6.4. *In Line 4 of Algorithm 6.2, we slightly deviate from Line 15 in [23, Algorithm 1]. Because the ReLU NN $((\overline{A}_1, \overline{b}_1), \ldots, (\overline{A}_\ell, \overline{b}_\ell), (I_{N_\ell \times N_\ell}, 0_{N_\ell}))$ realizes a continuous function and we only consider inputs $\overline{x}$ from the compact set $[0,1]^d$, the maximum in Line 4 exists and is finite. We will use this theoretical value of X_ℓ. We note that in [23], it is argued that computing the maximum over (a statistically representative subset of) the training data is sufficient in practice. See part (iv) of [23, Section 4.1]. By defining X_ℓ to be the theoretical maximum, rather than an empirical maximum, it is not necessary anymore to multiply it with a factor $(1+\zeta)$ for $\zeta > 0$ to obtain an upper bound that also holds for (practically) all inputs $\overline{x} \in [0,1]^d$. This multiplicative factor was used in part (iii) of [23, Section 4.2], we do not use it here.*

Another difference and simplification with respect to [23] is that we are only interested in transforming feedforward neural networks without convolutional layers, batch normalization and max pooling. See [23] for the transformation of such features.

Proposition 6.5 ([23, Theorem and Corollary in Section 2.1])**.** *Let $d, L \in \mathbb{N}$, $N_1, \ldots, N_L \in \mathbb{N}$, $x^{\min}, x^{\max} \in \mathbb{R}$, $x^{\min} < x^{\max}$, $\delta \in (0,1)$, $B > 0$ and let $\Phi = ((A_1, b_1), \ldots, (A_L, b_L))$ be a ReLU NN.*

Then, the SNN $\mathrm{S}(\Phi)$ which is the output of Algorithm 6.2 has input dimension d, depth L and layer dimensions $N_1, \ldots, N_L$ and satisfies, for all inputs $x \in [x^{\min}, x^{\max}]^d$, with $\overline{x} = \frac{1}{x^{\max} - x^{\min}}(x - x^{\min}(1, \ldots, 1)^\top)$ and $t_0 = (1, \ldots, 1)^\top - \overline{x}$, that $\mathrm{R}(\Phi)(x) = \mathrm{R}(\mathrm{S}(\Phi))(t_0)$. In addition, $M(\mathrm{S}(\Phi)) \le M(\Phi)$.

Algorithm 6.1 [23, Algorithm 1] The inputs are $d, L \in \mathbb{N}$, $N_1, \dots, N_L \in \mathbb{N}$, constants $\delta \in (0,1)$ and $B > 0$ and a ReLU NN Φ which takes inputs from $[x^{\min}, x^{\max}]^d$. The output is a ReLU NN $\overline{\Phi} = ((\overline{A}_1, \overline{b}_1), \dots, (\overline{A}_L, \overline{b}_L))$ with the same input dimension d, depth L and layer dimensions $N_1, \dots, N_L$, which takes inputs from $[0,1]^d$ such that $\mathrm{R}(\Phi)(x) = \mathrm{R}(\overline{\Phi})(\overline{x})$ for all $x \in [x^{\min}, x^{\max}]^d$ and $\overline{x} = \frac{1}{x^{\max} - x^{\min}}(x - x^{\min}(1, \dots, 1)^\top)$, and such that for all $\ell = 1, \dots, L-1$ and $i = 1, \dots, N_\ell$ holds $\sum_{j=1}^{N_{\ell-1}} \overline{A}_{ij} \in [-B, \delta]$. To rescale the weights and biases of the given ReLU NN Φ without changing the NN output, the algorithm exploits the positive homogeneity of the ReLU activation function $\rho(\lambda x) = \lambda \rho(x)$ for all $\lambda > 0$ and $x \in \mathbb{R}$.

Input: $d, L \in \mathbb{N}$, $N_1, \dots, N_L \in \mathbb{N}$, $x^{\min}, x^{\max} \in \mathbb{R}$, $x^{\min} < x^{\max}$, $\delta \in (0,1)$, $B > 0$ and a ReLU NN $\Phi = ((A_1, b_1), \dots, (A_L, b_L))$

Output: A ReLU NN $\overline{\Phi} = ((\overline{A}_1, \overline{b}_1), \dots, (\overline{A}_L, \overline{b}_L))$

```
1: for i = 1, ..., N_1 do
2:     For j = 1, ..., N_0: (Ā_1)_ij ← (x^max − x^min)(A_1)_ij
3:     (b̄_1)_i ← (b_1)_i + x^min ∑_{j=1}^{d} (A_1)_ij
4: end for
5: For ℓ = 2, ..., L: Ā_ℓ ← A_ℓ, b̄_ℓ ← b_ℓ,
6: for ℓ = 1, ..., L − 1 do
7:     for i = 1, ..., N_ℓ do
8:         (c_ℓ)_i ← ∑_{j=1}^{N_{ℓ−1}} Ā_ij
9:         if (c_ℓ)_i > 1 − δ then
10:            For j = 1, ..., N_{ℓ−1}: (Ā_ℓ)_ij ← (1−δ)/(c_ℓ)_i (Ā_ℓ)_ij
11:            (b̄_ℓ)_i ← (1−δ)/(c_ℓ)_i (b̄_ℓ)_i
12:            For k = 1, ..., N_{ℓ+1}: (Ā_{ℓ+1})_ki ← (c_ℓ)_i/(1−δ) (Ā_{ℓ+1})_ki
13:        else if (c_ℓ)_i < −B then
14:            For j = 1, ..., N_{ℓ−1}: (Ā_ℓ)_ij ← B/|(c_ℓ)_i| (Ā_ℓ)_ij
15:            (b̄_ℓ)_i ← B/|(c_ℓ)_i| (b̄_ℓ)_i
16:            For k = 1, ..., N_{ℓ+1}: (Ā_{ℓ+1})_ki ← |(c_ℓ)_i|/B (Ā_{ℓ+1})_ki
17:        end if
18:    end for
19: end for
20: return Φ̄ ← ((Ā_1, b̄_1), ..., (Ā_L, b̄_L))
```

Proof. The formula for the realization was proved in [23].

The fact that the ReLU NN $\overline{\Phi}$, which is the output of Algorithm 6.1 applied to a ReLU NN Φ, has the same input dimension and layer dimensions as Φ can be observed from the lines in the algorithm in which the weight matrices are initialized. These are Lines 2 and 5. Other lines of the algorithm do not change the sizes of the weight matrices. From Line 20 we see that the network $\overline{\Phi}$ returned by the algorithm has the same number of layers L as the input network Φ. The same ideas apply to Algorithm 6.2, where we see from Lines 7 and 11, where the weight matrices are computed, that the input dimension and the layer dimensions of $\mathrm{S}(\Phi)$ equal those of the output $\overline{\Phi}$

Algorithm 6.2 [23, Algorithm 2] The inputs are $d, L \in \mathbb{N}$, $N_1, \dots, N_L \in \mathbb{N}$, constants $\delta \in (0,1)$, $B > 0$ and a ReLU NN Φ which takes inputs from $[x^{\min}, x^{\max}]^d$. The output is a spiking neural network $\mathrm{S}(\Phi)$ with the same input dimension d, depth L and layer dimensions $N_1, \dots, N_L$. First, the neural network weights are rescaled using Algorithm 6.1. Then, a spiking neural network is defined such that for all $x \in [x_{\min}, x_{\max}]^d$, with $\overline{x} = \frac{1}{x^{\max} - x^{\min}}(x - x^{\min}(1, \dots, 1)^\top)$ and $t_0 = (1, \dots, 1)^\top - \overline{x}$, for all $\ell = 1, \dots, L-1$, the output $(\overline{x}_\ell)_i := \mathrm{R}(((\overline{A}_1, \overline{b}_1), \dots, (\overline{A}_\ell, \overline{b}_\ell), (I_{N_\ell \times N_\ell}, 0_{N_\ell})))(\overline{x})$ of neuron i in layer ℓ of the rescaled ReLU NN after applying ReLU activation corresponds to a spiking time $(t_\ell)_i = t_\ell^{\max} - (\overline{x}_\ell)_i$, and such that $\mathrm{R}(\Phi)(x) = \mathrm{R}(\mathrm{S}(\Phi))(t_0)$.

Input: $d, L \in \mathbb{N}$, $N_1, \dots, N_L \in \mathbb{N}$, $x^{\min}, x^{\max} \in \mathbb{R}$, $x^{\min} < x^{\max}$, $\delta \in (0,1)$, $B > 0$ and a ReLU NN $\Phi = ((A_1, b_1), \dots, (A_L, b_L))$
Output: An SNN $\mathrm{S}(\Phi) = ((J_1, \vartheta_1, \alpha_1, t_1^{\min}, t_1^{\max}), \dots, (J_L, \alpha_L, t_L^{\min}, t_L^{\max}))$
1: Compute $\overline{\Phi} \leftarrow ((\overline{A}_1, \overline{b}_1), \dots, (\overline{A}_L, \overline{b}_L))$ with Algorithm 6.1
2: $t_0^{\min} \leftarrow 0$, $t_0^{\max} \leftarrow 1$
3: **for** $\ell = 1, \dots, L-1$ **do**
4: $X_\ell \leftarrow \max_{\overline{x} \in [0,1]^d} \|\mathrm{R}(((\overline{A}_1, \overline{b}_1), \dots, (\overline{A}_\ell, \overline{b}_\ell), (I_{N_\ell \times N_\ell}, 0_{N_\ell})))(\overline{x})\|_\infty$, where $I_{N_\ell \times N_\ell} \in \mathbb{R}^{N_\ell \times N_\ell}$ denotes the identity matrix, and $0_{N_\ell} \in \mathbb{R}^{N_\ell}$ the zero vector.
5: $t_\ell^{\min} \leftarrow t_{\ell-1}^{\max}$, $t_\ell^{\max} \leftarrow t_{\ell-1}^{\max} + X_\ell$, $\alpha_\ell \leftarrow (1, \dots, 1)^\top \in \mathbb{R}^{N_\ell}$
6: **for** $i = 1, \dots, N_\ell$ **do**
7: For $j = 1, \dots, N_{\ell-1}$: $(J_\ell)_{ij} \leftarrow (\alpha_\ell)_i (\overline{A}_\ell)_{ij} / \Big(1 - \sum_{j=1}^{N_{\ell-1}} (\overline{A}_\ell)_{ij}\Big)$
8: $(\vartheta_\ell)_i \leftarrow (\alpha_\ell)_i (t_\ell^{\max} - t_{\ell-1}^{\min}) + \sum_{j=1}^{N_{\ell-1}} (J_\ell)_{ij} (t_\ell^{\max} - t_\ell^{\min}) - \Big((\alpha_\ell)_i + \sum_{j=1}^{N_{\ell-1}} (J_\ell)_{ij}\Big)(\overline{b}_\ell)_i$
9: **end for**
10: **end for**
11: $t_L^{\min} \leftarrow t_{L-1}^{\max}$, $t_L^{\max} \leftarrow t_{L-1}^{\max}$, $J_L \leftarrow \overline{A}_L$, $\alpha_L \leftarrow \overline{b}_L / (t_{L-1}^{\max} - t_{L-1}^{\min})$
12: **return** $\mathrm{S}(\Phi) \leftarrow ((J_1, \vartheta_1, \alpha_1, t_1^{\min}, t_1^{\max}), \dots, (J_L, \alpha_L, t_L^{\min}, t_L^{\max}))$

of Algorithm 6.1, and thus those of Φ. From Line 12, we observe that the number of layers of $\mathrm{S}(\Phi)$ is L, which is the same as that of $\overline{\Phi}$ and that of Φ.

To prove the bound on the network size, we first observe that in all lines which affect the weights, which are Lines 2, 5, 10, 12, 14 and 16 of Algorithm 6.1 and Lines 7 and 11 of Algorithm 6.2, the sign of the weights is not changed. In particular, the number of nonzero weights of the SNN equals that of the ReLU NN, which implies the desired neural network size bound. □

6.3 Spiking Neural Network Solution Approximation

As a direct consequence of Propositions 5.2 and 6.5 we obtain the expression rate bounds for solutions of (2.1)–(2.2) with spiking NNs.

Theorem 6.6. *Assume that (2.3) holds. For* $\varepsilon \in (0,1]$ *let* u_ε *be the solution of (3.1). Recall from Proposition 5.2 the constant* $\tilde{\kappa}_0 > 0$ *(depending only on* b *and* f*) and for all* $\kappa \in (0, \tilde{\kappa}_0)$ *and* $p \in \mathbb{N}$ *the ReLU NN* $\Phi_\varepsilon^{FEM,\kappa,p}$.

Then, with the positive constants C and β from Proposition 5.2, independent of ε and p, the SNN $\mathrm{S}(\Phi_\varepsilon^{FEM,\kappa,p})$ *constructed by Algorithm 6.2 satisfies*

$$\left\{\varepsilon^{1/2}\|u'_\varepsilon - \mathrm{R}(\mathrm{S}(\Phi_\varepsilon^{FEM,\kappa,p}))'\|_{L^2(I)} + \|u_\varepsilon - \mathrm{R}(\mathrm{S}(\Phi_\varepsilon^{FEM,\kappa,p}))\|_{L^2(I)}\right\} \leq Ce^{-\beta p}, \quad (6.2)$$

$$\|u_\varepsilon - \mathrm{R}(\mathrm{S}(\Phi_\varepsilon^{FEM,\kappa,p}))\|_{L^\infty(I)} \leq Ce^{-\beta p}, \quad (6.3)$$

and $\mathrm{R}(\mathrm{S}(\Phi_\varepsilon^{FEM,\kappa,p}))(\pm 1) = 0$.

For a constant $\tilde{C} = \tilde{C}(\beta) > 0$ depending only on β, the SNN depth and size are bounded as follows:

$$L\left(\mathrm{S}(\Phi_\varepsilon^{FEM,\kappa,p})\right) \leq \tilde{C}p(1 + \log_2(p)), \qquad M\left(\mathrm{S}(\Phi_\varepsilon^{FEM,\kappa,p})\right) \leq \tilde{C}p^2. \quad (6.4)$$

The weights in the hidden layers are independent of u_ε and depend only on κ, p, ε and β.

Remark 6.7. *The presently used reasoning to infer expression rate bounds for spiking NN architectures from rates proved for ReLU NNs naturally also applies to other results, e.g. those in [11, 15, 17] and also for so-called* operator networks *of strict ReLU type in [12].*

7 tanh Neural Network Approximations

In Section 2, we have seen in Remark 2.4 that when the reaction coefficient function $b(x)$ is constant, then the boundary layer functions are known explicitly and are given by (2.11). Particularly simple NN approximations of these boundary layer functions can be obtained with NNs which have one hidden layer and use as activation function

$$\tanh(x) = \tfrac{1-\exp(-2x)}{1+\exp(-2x)}, \qquad x \in \mathbb{R}.$$

This is the topic of Section 7.1, and is of independent interest. In Section 7.2, we state the principal result of this section: exponential DNN expression rate bounds in Sobolev norms on the set of solutions to (2.1)–(2.2) which are uniform in the singular perturbation parameter $\varepsilon \in (0, 1]$. Based on expression rate bounds in Sections 7.3–7.4 and Appendix A, in Section 7.5 we construct deep tanh-activated NN approximations of analytic functions, and in particular of the smooth term in (2.11), sharpening previous results in [4].

In Section 7.6, we prove the main result of Section 7 by combining the tanh-NN approximation of the smooth term with a tanh-NN approximation of the boundary layer components of the solution u^ε developed in Section 7.1.

Throughout Section 7, we will use the convention that for a function $F \in W^{1,\infty}(D)$ for a domain $D \subset \mathbb{R}^d$, $d \in \mathbb{N}$, the $W^{1,\infty}(D)$-norm is defined as $\|F\|_{W^{1,\infty}(D)} = \max\{\|F\|_{L^\infty(D)}, \max_{j=1}^d \|\frac{\partial}{\partial x_j}F\|_{L^\infty(D)}\}$.

7.1 tanh Emulation of the Exponential Function

We analyze tanh NN approximations of the exponential function in Lemma 7.2 below, based on the following observation.

Lemma 7.1. *For all* $x_0 \geq 0$ *and all* $x \geq 0$ *there holds*

$$\begin{aligned} E(x) &:= \left|\exp(-x) - \exp(x_0)\tfrac{1}{2}(1 - \tanh(\tfrac{1}{2}x + \tfrac{1}{2}x_0))\right| \leq \exp(-x_0), \\ |E'(x)| &:= \left|\tfrac{\partial E}{\partial x}(x)\right| \leq 2\exp(-x_0). \end{aligned}$$

Proof. We start by noting that for all $x \in \mathbb{R}$

$$\begin{aligned} (1 - \tanh(x)) = \tfrac{2\exp(-2x)}{1+\exp(-2x)}, \qquad \tfrac{1}{2}(1 - \tanh(\tfrac{1}{2}x)) = \tfrac{\exp(-x)}{1+\exp(-x)}, \\ \left|\exp(-x) - \tfrac{1}{2}(1 - \tanh(\tfrac{1}{2}x))\right| = \left|\exp(-x) - \tfrac{\exp(-x)}{1+\exp(-x)}\right| = \tfrac{1}{1+\exp(-x)}\exp(-2x). \end{aligned}$$

Using this result in the fourth step below, we obtain that for all $x_0 \geq 0$ and all $x \geq 0$

$$\begin{aligned} E(x) &:= \left|\exp(-x) - \exp(x_0)\tfrac{1}{2}(1 - \tanh(\tfrac{1}{2}x + \tfrac{1}{2}x_0))\right| \\ &= \left|\exp(-x) - \exp(x_0)\tfrac{\exp(-x-x_0)}{1+\exp(-x-x_0)}\right| \\ &= \exp(x_0)\left|\exp(-x - x_0) - \tfrac{\exp(-x-x_0)}{1+\exp(-x-x_0)}\right| \\ &= \exp(x_0)\tfrac{1}{1+\exp(-x-x_0)}\exp(-2x - 2x_0) \\ &= \tfrac{1}{1+\exp(-x-x_0)}\exp(-2x - x_0) \\ &\leq \exp(-2x - x_0) \leq \exp(-x_0). \end{aligned}$$

In addition, we obtain

$$\begin{aligned} E'(x) &= \tfrac{1}{1+\exp(-x-x_0)} \cdot -2\exp(-2x - x_0) \\ &\quad + \tfrac{-1}{(1+\exp(-x-x_0))^2} \cdot -\exp(-x - x_0)\exp(-2x - x_0) \\ &= \Big(-\tfrac{2}{1+\exp(-x-x_0)} + \tfrac{1}{(1+\exp(-x-x_0))^2}\exp(-x - x_0)\Big)\exp(-2x - x_0), \\ |E'(x)| &\leq 2\exp(-2x - x_0) \leq 2\exp(-x_0). \end{aligned}$$

We used that the absolute value of the negative term is larger than that of the positive term, hence $|E'(x)|$ is bounded from above by the absolute value of the negative term. □

As a result, we have the following *shallow* tanh *NN approximation rate bound of the exponential function* $\exp(-\cdot)$. This result is of independent interest, as exponential boundary layer functions appear in a wide range of multivariate, singular perturbation problems (see e.g. [2] and the references there).

Lemma 7.2. *For all $\tau \in (0,1]$ there exists a* tanh *NN $\Phi_\tau^{\exp}$ such that for all $x \geq 0$ there holds*

$$|\exp(-x) - \mathrm{R}(\Phi_\tau^{\exp})(x)| \leq \exp(-\tau), \tag{7.1a}$$
$$|-\exp(-x) - \mathrm{R}(\Phi_\tau^{\exp})'(x)| \leq \exp(-\tau), \tag{7.1b}$$

and such that $L(\Phi_\tau^{\exp}) = 2$ and $M(\Phi_\tau^{\exp}) = 4$.

Proof. We set $x_0 = \log(2/\tau)$, such that $\exp(x_0) = \frac{2}{\tau}$, and define the tanh NN

$$\begin{aligned}\Phi_\tau^{\exp} &:= \left(\left(\tfrac{1}{2}, \tfrac{1}{2}x_0\right), \left(-\tfrac{1}{2}\exp(x_0), \tfrac{1}{2}\exp(x_0)\right)\right)\\ &= \left(\left(\tfrac{1}{2}, \tfrac{1}{2}\log(2/\tau)\right), \left(-\tfrac{1}{\tau}, \tfrac{1}{\tau}\right)\right),\end{aligned}$$

which implies that

$$\mathrm{R}(\Phi_\tau^{\exp})(x) = \exp(x_0)\tfrac{1}{2}(1 - \tanh(\tfrac{1}{2}x + \tfrac{1}{2}x_0)), \qquad \text{for all } x \geq 0.$$

Using Lemma 7.1, we obtain the error bounds

$$\begin{aligned}|\exp(-x) - \mathrm{R}(\Phi_\tau^{\exp})(x)| &\leq \exp(-x_0) = \tfrac{\tau}{2}, &&\text{for all } x \geq 0,\\ |-\exp(-x) - \mathrm{R}(\Phi_\tau^{\exp})'(x)| &\leq 2\exp(-x_0) = \tau, &&\text{for all } x \geq 0.\end{aligned}$$

From the definition of $\Phi_\tau^{\exp}$, we observe that $L(\Phi_\tau^{\exp}) = 2$ and $M(\Phi_\tau^{\exp}) = 4$. □

Remark 7.3. *We can apply the above analysis also to the sigmoid activation*

$$\sigma(x) = \tfrac{\exp(x)}{1+\exp(x)} = \tfrac{1}{1+\exp(-x)}, \qquad x \in \mathbb{R}.$$

We observe that for all $x \in \mathbb{R}$ holds $\sigma(-x) = \frac{\exp(-x)}{1+\exp(-x)} = \frac{1}{2}(1 - \tanh(\frac{1}{2}x))$. Thus, the σ-NN

$$\Phi_\tau^{\sigma,\exp} := \big((-1, -x_0), \big(\exp(x_0), 0\big)\big) = \big((-1, -\log(2/\tau)), \big(\tfrac{2}{\tau}, 0\big)\big)$$

satisfies

$$L(\Phi_\tau^{\sigma,\exp}) = 2, \;\; M(\Phi_\tau^{\sigma,\exp}) = 3, \;\; \textit{and}\;\; \mathrm{R}(\Phi_\tau^{\sigma,\exp}) = \exp(x_0)\sigma(-\cdot - x_0) = \mathrm{R}(\Phi_\tau^{\exp}).$$

Here $\Phi_\tau^{\exp}$ is the tanh *NN from Lemma 7.2. In particular, $\Phi_\tau^{\sigma,\exp}$ satisfies* (7.1)*.*

7.2 tanh NN Solution Approximation

We state the main result of this section, namely *ε-robust approximation rates for strict* tanh*-activated deep NN approximations of solution families $\{u^\varepsilon : 0 < \varepsilon \leq 1\} \subset H_0^1(I)$ to the singularly perturbed, reaction-diffusion BVP* (2.1)–(2.2). To leverage the Lemma 7.2, we consider again (2.1)–(2.2) in the special case that the reaction coefficient $b(x)$ is constant, and equals $b \in \mathbb{R}$. Without loss of generality, we assume that $b = 1$. The solutions for general $b > 0$ can be found by solving the BVP with ε^2 replaced by ε^2/b and f replaced by f/b.

Theorem 7.4. *Assume that (2.3) holds and that the reaction coefficient function is constant and satisfies $b(x) = 1$ for all $x \in I = (-1, 1)$. For $\varepsilon \in (0, 1]$, let u_ε be the solution of (3.1).*

Then, for all $p \in \mathbb{N}$ there exists a tanh *NN $\Phi^{u_\varepsilon,p}_\varepsilon$ such that, with positive constants C, β, independent of $\varepsilon \in (0, 1]$ and of $p \geq 1$, it holds that $\|u_\varepsilon - \mathrm{R}(\Phi^{u_\varepsilon,p}_\varepsilon)\|_{W^{1,\infty}(I)} \leq Ce^{-\beta p}$, which implies that*

$$\left\{\varepsilon^{1/2}\|u'_\varepsilon - \mathrm{R}(\Phi^{u_\varepsilon,p}_\varepsilon)'\|_{L^2(I)} + \|u_\varepsilon - \mathrm{R}(\Phi^{u_\varepsilon,p}_\varepsilon)\|_{L^2(I)}\right\} \leq Ce^{-\beta p}, \tag{7.2}$$

$$\|u_\varepsilon - \mathrm{R}(\Phi^{u_\varepsilon,p}_\varepsilon)\|_{L^\infty(I)} \leq Ce^{-\beta p}. \tag{7.3}$$

For a constant $\tilde{C} > 0$ independent of f, p, ε, C and β, the network depth and size are bounded as follows:

$$L(\Phi^{u_\varepsilon,p}_\varepsilon) = \lceil \log_2(p) \rceil + 1, \qquad M(\Phi^{u_\varepsilon,p}_\varepsilon) \leq \tilde{C}p. \tag{7.4}$$

The weights and biases in the hidden layers are independent of u_ε and depend only on p, ε and β.

A result corresponding to this theorem also holds for sigmoid-activated NNs. We show this in Appendix B.

The rest of this section is devoted to the proof of Theorem 7.4. In Section 7.3, we review results (in principle known) on a calculus of tanh-activated deep NNs, in particular in Section 7.4 tanh-activated deep NN emulation of the identity and of products of real numbers. Section 7.5 addresses the tanh-emulation of analytic functions, which are based on the novel emulations bounds of Chebyšev polynomials by deep tanh-activated NNs, which are proved in Appendix A.

7.3 Calculus of NNs

In the following sections, we will construct NNs from smaller networks using a *calculus of NNs*, which we now recall from [18]. The results cited from [18] were derived for NNs which only use the ReLU activation function, but they also hold for networks with different activation functions without modification.

Proposition 7.5 (Parallelization of NNs [18, Definition 2.7])**.** *For $d, L \in \mathbb{N}$ let $\Phi^1 = \left((A^{(1)}_1, b^{(1)}_1), \ldots, (A^{(1)}_L, b^{(1)}_L)\right)$ and $\Phi^2 = \left((A^{(2)}_1, b^{(2)}_1), \ldots, (A^{(2)}_L, b^{(2)}_L)\right)$ be two NNs with input dimension d and depth L. Let the* parallelization *$\mathrm{P}(\Phi^1, \Phi^2)$ of Φ^1 and Φ^2 be defined by*

$$\begin{aligned}
\mathrm{P}(\Phi^1, \Phi^2) &:= ((A_1, b_1), \ldots, (A_L, b_L)), \\
A_1 &= \begin{pmatrix} A^{(1)}_1 \\ A^{(2)}_1 \end{pmatrix}, \quad A_\ell = \begin{pmatrix} A^{(1)}_\ell & 0 \\ 0 & A^{(2)}_\ell \end{pmatrix}, \qquad && \text{for } \ell = 2, \ldots L, \\
b_\ell &= \begin{pmatrix} b^{(1)}_\ell \\ b^{(2)}_\ell \end{pmatrix}, && \text{for } \ell = 1, \ldots L.
\end{aligned}$$

Then,

$$\mathrm{R}(\mathrm{P}(\Phi^1,\Phi^2))(x) = (\mathrm{R}(\Phi^1)(x), \mathrm{R}(\Phi^2)(x)), \quad \textit{for all } x \in \mathbb{R}^d,$$
$$L(\mathrm{P}(\Phi^1,\Phi^2)) = L, \qquad M(\mathrm{P}(\Phi^1,\Phi^2)) = M(\Phi^1) + M(\Phi^2).$$

The parallelization of more than two NNs is handled by repeated application of Proposition 7.5.

Similarly, we can also construct a NN emulating the sum of the realizations of two NNs.

Proposition 7.6 (Sum of NNs)**.** *For $d, N, L \in \mathbb{N}$ let $\Phi^1 = \Big((A_1^{(1)}, b_1^{(1)}), \ldots, (A_L^{(1)}, b_L^{(1)})\Big)$ and $\Phi^2 = \Big((A_1^{(2)}, b_1^{(2)}), \ldots, (A_L^{(2)}, b_L^{(2)})\Big)$ be two NNs with input dimension d, output dimension N and depth L. Let the* sum $\Phi^1 + \Phi^2$ *of Φ^1 and Φ^2 be defined by*

$$\begin{aligned}
\Phi^1 + \Phi^2 &:= ((A_1, b_1), \ldots, (A_L, b_L)),\\
A_1 &= \begin{pmatrix} A_1^{(1)} \\ A_1^{(2)} \end{pmatrix}, \quad b_1 = \begin{pmatrix} b_1^{(1)} \\ b_1^{(2)} \end{pmatrix},\\
A_\ell &= \begin{pmatrix} A_\ell^{(1)} & 0 \\ 0 & A_\ell^{(2)} \end{pmatrix}, \quad b_\ell = \begin{pmatrix} b_\ell^{(1)} \\ b_\ell^{(2)} \end{pmatrix}, \qquad \textit{for } \ell = 2, \ldots L-1.\\
A_L &= \Big(A_L^{(1)} \; A_L^{(2)}\Big), \quad b_L = b_L^{(1)} + b_L^{(2)}.
\end{aligned}$$

Then,

$$\mathrm{R}(\Phi^1 + \Phi^2)(x) = \mathrm{R}(\Phi^1)(x) + \mathrm{R}(\Phi^2)(x), \quad \textit{for all } x \in \mathbb{R}^d,$$
$$L(\Phi^1 + \Phi^2) = L, \qquad M(\Phi^1 + \Phi^2) \leq M(\Phi^1) + M(\Phi^2).$$

We will sometimes use the parallelization of networks which do not have the same inputs.

Proposition 7.7 (Full parallelization of NNs [5, Setting 5.2])**.** *For $L \in \mathbb{N}$ let $\Phi^1 = \Big((A_1^{(1)}, b_1^{(1)}), \ldots, (A_L^{(1)}, b_L^{(1)})\Big)$ and $\Phi^2 = \Big((A_1^{(2)}, b_1^{(2)}), \ldots, (A_L^{(2)}, b_L^{(2)})\Big)$ be two NNs with the same depth L, with input dimensions $N_0^1 = d_1$ and $N_0^2 = d_2$, respectively.*

Then, the NN defined by

$$\mathrm{FP}(\Phi^1, \Phi^2) := ((A_1, b_1), \ldots, (A_L, b_L)),$$

$$A_\ell = \begin{pmatrix} A_\ell^{(1)} & 0 \\ 0 & A_\ell^{(2)} \end{pmatrix}, \qquad b_\ell = \begin{pmatrix} b_\ell^{(1)} \\ b_\ell^{(2)} \end{pmatrix}, \qquad \textit{for } \ell = 1, \ldots L,$$

with $d = d_1 + d_2$-dimensional input and depth L, called full parallelization of Φ^1 and Φ^2, *satisfies that for all $x = (x_1, x_2) \in \mathbb{R}^d$ with $x_i \in \mathbb{R}^{d_i}$, $i = 1, 2$*

$$\mathrm{R}(\mathrm{FP}(\Phi^1, \Phi^2))(x_1, x_2) = \big(\mathrm{R}(\Phi^1)(x_1), \mathrm{R}(\Phi^2)(x_2)\big)$$

and $M(\mathrm{FP}(\Phi^1, \Phi^2)) = M(\Phi^1) + M(\Phi^2)$.

Finally, we recall the concatenation of two NNs.

Definition 7.8 (Concatenation of NNs [18, Definition 2.2])**.** *For $L^{(1)}, L^{(2)} \in \mathbb{N}$, let $\Phi^1 = \Big((A_1^{(1)}, b_1^{(1)}), \ldots, (A_{L^{(1)}}^{(1)}, b_{L^{(1)}}^{(1)})\Big)$ and $\Phi^2 = \Big((A_1^{(2)}, b_1^{(2)}), \ldots, (A_{L^{(2)}}^{(2)}, b_{L^{(2)}}^{(2)})\Big)$ be two NNs such that the input dimension of Φ^1, which we will denote by k, equals the output dimension of Φ^2. Then, the* concatenation *of Φ^1 and Φ^2 is the NN of depth $L := L^{(1)} + L^{(2)} - 1$ defined as*

$$\begin{aligned}
\Phi^1 \bullet \Phi^2 &:= ((A_1, b_1), \ldots, (A_L, b_L)),\\
(A_\ell, b_\ell) &= (A_\ell^{(2)}, b_\ell^{(2)}), \qquad \text{for } \ell = 1, \ldots, L^{(2)} - 1,\\
A_{L^{(2)}} &= A_1^{(1)} A_{L^{(2)}}^{(2)}, \qquad b_{L^{(2)}} = A_1^{(1)} b_{L^{(2)}}^{(2)} + b_1^{(1)},\\
(A_\ell, b_\ell) &= (A_{\ell - L^{(2)}+1}^{(1)}, b_{\ell - L^{(2)}+1}^{(1)}), \qquad \text{for } \ell = L^{(2)} + 1, \ldots, L^{(1)} + L^{(2)} - 1.
\end{aligned}$$

It follows immediately from this definition that $\mathrm{R}(\Phi^1 \bullet \Phi^2) = \mathrm{R}(\Phi^1) \circ \mathrm{R}(\Phi^2)$.

7.4 tanh Emulation of Identity and Products

Unlike ReLU NNs, tanh-activated NNs can not represent the identity exactly. As various constructions require identity maps, we provide a corresponding tanh NN emulation of the identity. We also recall the tanh NN emulation of products from [4].

Lemma 7.9 (See [4, Lemma 3.1] and [4, Corollary 3.7])**.** *For all $\tau, M > 0$ and all $L \in \mathbb{N}$, $L \geq 2$ there exists a* tanh*-activated NN $\Phi^{\mathrm{Id}}_{1,L,\tau,M}$ of depth L, with input dimension one and output dimension one, such that*

$$\big\|\mathrm{Id}_{\mathbb{R}} - \mathrm{R}(\Phi^{\mathrm{Id}}_{1,L,\tau,M})\big\|_{W^{1,\infty}((-M,M))} \leq \tau. \tag{7.5}$$

There exists $C > 0$ such that for all $L \in \mathbb{N}$, $L \geq 2$ there holds $M(\Phi^{\mathrm{Id}}_{1,L,\tau,M}) \leq CL$ for a constant C independent of τ, M and L, and also the layer dimensions of the hidden layers (denoted by $N_1, \ldots, N_{L-1}$ in the notation of Definition 4.1) are at most C.

For all $\tau, M > 0$ there exists a tanh *NN $\Phi^{\mathrm{Prod}}_{\tau,M}$ of depth 2, with input dimension two and output dimension one such that*

$$\left\|\prod_{i=1}^{2} x_i - \mathrm{R}(\Phi^{\mathrm{Prod}}_{\tau,M})(x_1, x_2)\right\|_{W^{1,\infty}((-M,M)^2)} \leq \tau.$$

There exists $C > 0$ such that $M(\Phi^{\mathrm{Prod}}_{\tau,M}) \leq C$ for a constant C independent of τ, M.

Proof. We first prove the statements regarding identity networks. Without loss of generality, assume that $\tau \leq 1$ (if $\tau > 1$, we can use the identity network defined below with 1 instead of τ).

By [4, Lemma 3.1] there exists a tanh NN $\Phi^{\mathrm{Id}}_{1,2,\tau,M}$ of depth 2 with input dimension one and output dimension one and with a fixed size independent of τ, M, such that (7.5) holds.

For $L > 2$ we use Definition 7.8 to define $\Phi^{\mathrm{Id}}_{1,L,\tau,M} := \Phi^{\mathrm{Id}}_{1,2,\tau/3,M+\tau/3} \bullet \Phi^{\mathrm{Id}}_{1,L-1,\tau/3,M}$. In this proof, we will use the shorthand notation $\Phi^1 := \Phi^{\mathrm{Id}}_{1,2,\tau/3,M+\tau/3}$ and $\Phi^2 := \Phi^{\mathrm{Id}}_{1,L-1,\tau/3,M}$. We can estimate the error by

$$
\begin{aligned}
&\|\operatorname{Id}_{\mathbb{R}} - \mathrm{R}(\Phi^1 \bullet \Phi^2)\|_{L^\infty((-M,M))} \\
\leq{}& \|\operatorname{Id}_{\mathbb{R}} - \mathrm{R}(\Phi^2)\|_{L^\infty((-M,M))} + \|(\operatorname{Id}_{\mathbb{R}} - \mathrm{R}(\Phi^1)) \circ \mathrm{R}(\Phi^2)\|_{L^\infty((-M,M))} \\
\leq{}& \tau/3 + \tau/3 \leq \tau, \\
&\|\operatorname{Id}'_{\mathbb{R}} - \mathrm{R}(\Phi^1 \bullet \Phi^2)'\|_{L^\infty((-M,M))} \\
\leq{}& \|\operatorname{Id}'_{\mathbb{R}} - \mathrm{R}(\Phi^2)'\|_{L^\infty((-M,M))} + \left\| \big((\operatorname{Id}_{\mathbb{R}} - \mathrm{R}(\Phi^1)) \circ \mathrm{R}(\Phi^2) \big)' \right\|_{L^\infty((-M,M))} \\
\leq{}& \|\operatorname{Id}'_{\mathbb{R}} - \mathrm{R}(\Phi^2)'\|_{L^\infty((-M,M))} \\
&+ \|\operatorname{Id}'_{\mathbb{R}} - \mathrm{R}(\Phi^1)'\|_{L^\infty((-M-\tau/3,M+\tau/3))} \|\mathrm{R}(\Phi^2)'\|_{L^\infty((-M,M))} \\
\leq{}& \tau/3 + \tau/3(1+\tau/3) \leq \tau/3 + 2\tau/3 \leq \tau.
\end{aligned}
$$

In terms of the constant C, which is independent of $\tau, M > 0$, for which $M(\Phi^{\mathrm{Id}}_{1,2,\tau,M}) \leq C$ (its existence follows from [4, Lemma 3.1]), it follows that the number of neurons in each hidden layer of $\Phi^{\mathrm{Id}}_{1,L,\tau,M}$ (which are denoted by $N_1, \ldots, N_{L-1}$ in the notation of Definition 4.1) are all at most C.[4] Therefore, the number of nonzero weights in each layer is at most C^2 and the number of nonzero biases in each layer is at most C, giving a total network size of at most $LC(C+1)$, which is the desired bound (when we write C instead of the constant $C(C+1)$).

The statements for product networks correspond to [4, Corollary 3.7]. □

Identity networks with multiple inputs are obtained as the full parallelization of identity networks with one input.

Definition 7.10. *For all $d \in \mathbb{N}$, we define $\Phi^{\mathrm{Id}}_{d,L,\tau,M} := \mathrm{FP}(\Phi^{\mathrm{Id}}_{1,L,\tau,M}, \ldots, \Phi^{\mathrm{Id}}_{1,L,\tau,M})$ as the full parallelization of d identity networks from Lemma 7.9.*

7.5 tanh Emulation of Analytic Functions

Exponential convergence of tanh NNs for the approximation of analytic functions is obtained by combining the result from Appendix A with the following classical result on polynomial approximation. See e.g. [3, Section 12.4]. We use the formulation from [13, Lemma 9].

[4] If the number of neurons in a layer is larger than the number of nonzero weights and biases in that layer, any neurons for which all associated weights and biases vanish, can be removed. If in a layer all weights and biases vanish, the network realizes a constant function, which means that the network can be replaced by a network of depth and size at most 1 which exactly emulates the constant function. These facts are proved in [18, Lemma G.1].

Lemma 7.11. *Let $I = (-1, 1)$ and assume that $u : I \to \mathbb{R}$ is analytic on $\overline{I} = [-1, 1]$, i.e. there exist $C_u, K_u > 0$ such that for all $n \in \mathbb{N}_0$ holds*

$$\|u^{(n)}\|_{L^\infty(I)} \leq C_u K_u^n n!. \tag{7.6}$$

Then, there exist constants $C, \beta > 0$ such that for all $p \in \mathbb{N}$ there exists a polynomial $v \in \mathcal{P}_p$ such that $\|u - v\|_{W^{1,\infty}(I)} \leq Ce^{-\beta p}$.

For example, such a polynomial v can be obtained from u by polynomial interpolation in the Gauß–Lobatto points, see [13, Lemma 11], or in the Clenshaw–Curtis points, see [25, Theorem 8.2] (the Clenshaw–Curtis points are introduced in [25, Chapter 2] and referred to as "Chebyshev points").

Proposition 7.12. *Assume that u satisfies* (7.6).

Then, there exist constants $C, \beta > 0$ such that for all $p \in \mathbb{N}$ there exists a tanh *NN $\Phi^{u,p}$ such that*

$$\|u - \mathrm{R}(\Phi^{u,p})\|_{W^{1,\infty}(I)} \leq Ce^{-\beta p}$$

with

$$L(\Phi^{u,p}) = \lceil \log_2(p) \rceil + 1, \quad \text{and} \quad M(\Phi^{u,p}) \leq \tilde{C}p$$

for a constant $\tilde{C} > 0$ independent of p, u, C and β.

The weights and biases in the hidden layers are independent of u. The weights and biases in the output layer are linear combinations of the function values of u in the Clenshaw–Curtis points.

Proof. For all $p \in \mathbb{N}$, let $v \in \mathcal{P}_p$ and $\beta > 0$ be as given by Lemma 7.11, let $\delta = \exp(-\beta p)$ and let $\Phi^{u,p} := \Phi^{v,p}_\delta$ be the network constructed in Corollary A.3. Then, for a constant $C > 0$ independent of p,

$$\begin{aligned} \|u - \mathrm{R}(\Phi^{u,p})\|_{W^{1,\infty}(I)} &\leq \|u - v\|_{W^{1,\infty}(I)} + \|v - \mathrm{R}(\Phi^{v,p}_\delta)\|_{W^{1,\infty}(I)} \\ &\leq C\exp(-\beta p) + \exp(-\beta p) \sum_{\ell=1}^{p} |v_\ell| \leq C\exp(-\beta p). \end{aligned}$$

In addition, we recall from Corollary A.3 that $L(\Phi^{u,p}) = L(\Phi^{v,p}_\delta) = \lceil \log_2(p) \rceil + 1$ and $M(\Phi^{u,p}) = M(\Phi^{v,p}_\delta) \leq \tilde{C}p$ for a constant $\tilde{C} > 0$ independent of p, u, C and β.

We observe from the proof of Corollary A.3 that the hidden layer weights are independent of u, and that the output layer weights are linear combinations of the Chebyšev coefficients of v. As mentioned in the text after Lemma 7.11, we may take v to be the interpolant of u in the Clenshaw–Curtis points, as in [25, Theorem 8.2]. Then, the Chebyšev coefficients of v can be computed from the function values of v in the Clenshaw–Curtis points, which equal those of u (v interpolates u in those points). □

Proposition 7.12 also holds for NNs with more general activation functions. In Appendix B, we prove that it holds for NNs whose activation function is in $C^2(U) \setminus \mathcal{P}_1$ for a nonempty, connected open subset $U \subset \mathbb{R}$.

7.6 Proof of Theorem 7.4

Proof. As in the proof of [13, Theorem 16], which we recalled in Proposition 3.2, for $\kappa_0 > 0$ and $\kappa \in (0, \kappa_0)$ as in that proposition we distinguish two cases. In the first, *asymptotic* case, the polynomial degree is so large with respect to ε that u_ε can be approximated directly, without treating the boundary layers separately. In the second, *pre-asymptotic* case, we treat the boundary layers separately according to (2.10)–(2.11).

If $\kappa p \varepsilon \geq \frac{1}{2}$, then the first part of the proof of [13, Theorem 16] shows that for all $p \in \mathbb{N}$ there exists a polynomial[5] $v \in \mathcal{P}_p$ such that $\|u - v\|_{W^{1,\infty}(I)} \leq Ce^{-\beta p}$ for constants $C, \beta > 0$ independent of p and ε. This polynomial can be approximated by a tanh NN by Corollary A.3. Defining $\delta = \exp(-\beta p)$ and $\Phi_\varepsilon^{u_\varepsilon,p} := \Phi_\delta^{v,p}$, there exists a constant $C > 0$ independent of p and ε such that

$$\begin{aligned}\|u - \mathrm{R}(\Phi_\varepsilon^{u_\varepsilon,p})\|_{W^{1,\infty}(I)} &\leq \|u - v\|_{W^{1,\infty}(I)} + \|v - \mathrm{R}(\Phi_\delta^{v,p})\|_{W^{1,\infty}(I)} \\ &\leq C\exp(-\beta p) + \exp(-\beta p)\sum_{\ell=1}^{p} |v_\ell| \leq C\exp(-\beta p),\end{aligned}$$

where $(v_\ell)_{\ell=0}^p$ are the Chebyšev coefficients of v. In addition, we recall from Corollary A.3 that $L(\Phi_\varepsilon^{u_\varepsilon,p}) = L(\Phi_\delta^{v,p}) = \lceil \log_2(p) \rceil + 1$ and $M(\Phi_\varepsilon^{u_\varepsilon,p}) = M(\Phi_\delta^{v,p}) \leq \tilde{C}p$ for a constant $\tilde{C} > 0$ independent of f, p, ε, C and β. In Corollary A.3, the hidden layer weights depend only on p.

If $\kappa p \varepsilon < \frac{1}{2}$, we use the decomposition (2.5) and separately approximate the ε-independent smooth part u_ε^S and the boundary layer functions, which equal $\tilde{u}_\varepsilon^\pm(x) = C^\pm e^{-(1\mp x)/\varepsilon}$ by Remark 2.4, for some constants $C^\pm > 0$ which are bounded independently of ε.

We approximate u_ε^S by the tanh NN $\Phi^{u_\varepsilon^S,p}$ from Proposition 7.12. There exist constants $C, \beta > 0$ independent of p and ε, and $\tilde{C} > 0$ independent of f, p, ε, C and β, such that it satisfies $\|u_\varepsilon^S - \mathrm{R}(\Phi^{u_\varepsilon^S,p})\|_{W^{1,\infty}(I)} \leq Ce^{-\beta p}$, $L(\Phi^{u_\varepsilon^S,p}) = \lceil \log_2(p) \rceil + 1$ and $M(\Phi^{u_\varepsilon^S,p}) \leq \tilde{C}p$.

For the approximation of $u_\varepsilon^\pm$, we use the approximation of the exponential function from Proposition 7.2, the concatenation from Definition 7.8 and the identity network from Lemma 7.9 to define

$$\Phi_\varepsilon^\pm := ((A_1^\pm, b_1^\pm, \mathrm{Id}_\mathbb{R})) \bullet \Phi_\tau^{\exp} \bullet \Phi_{1,L-1,\delta,M}^{\mathrm{Id}} \bullet ((A_2^\pm, b_2^\pm, \mathrm{Id}_\mathbb{R})), \tag{7.7}$$

for $A_1^\pm = C^\pm \exp(1) \in \mathbb{R}^{1\times 1}$, $b_1^\pm = 0 \in \mathbb{R}^1$, $A_2^\pm = \mp 1/\varepsilon \in \mathbb{R}^{1\times 1}$, $b_2^\pm = 1/\varepsilon + 1 \in \mathbb{R}^1$, $\tau := \exp(-\beta p)\varepsilon$, $\delta := \exp(-\beta p)\varepsilon$, $M := 2/\varepsilon + 1$ and $L := L(\Phi^{u_\varepsilon^S,p}) = \lceil \log_2(p) \rceil + 1$.

For $p = 1$, we have $L = 1$ and omit "$\bullet\Phi_{1,L-1,\delta,M}^{\mathrm{Id}}$" from the definition (because identity networks of depth 0 have not been defined). This will allow minor simplifications in the error bounds and the bounds on the network size for the case $p = 1$, which we will not consider explicitly.

[5] In [13], $v \in \mathcal{P}_p$ is taken to be the interpolant of u_ε in the $p + 1$ Gauß–Lobatto points.

Denoting by P the affine transformation $P : x \mapsto A_2^{\pm} x + b_2^{\pm} = (1 \mp x)/\varepsilon + 1$, the networks $\Phi_\varepsilon^{\pm}$ realize

$$\mathrm{R}(\Phi_\varepsilon^{\pm}) = C^{\pm} \exp(1) \mathrm{R}(\Phi_\tau^{\exp}) \circ \mathrm{R}(\Phi_{1,L-1,\delta,M}^{\mathrm{Id}}) \circ P. \tag{7.8}$$

We note that for $x \in [-1, 1]$ holds $P(x) = (1 \mp x)/\varepsilon + 1 \in [1, 1 + 2/\varepsilon]$, and that $\delta < 1$, from which it follows that $\mathrm{R}(\Phi_{1,L-1,\delta,M}^{\mathrm{Id}}) \circ P(x) \in [1 - \delta, 1 + \delta + 2/\varepsilon] \subset [0, 2 + 2/\varepsilon]$. It is necessary to add a positive number to the input (or output) of the identity network in order to guarantee that the input of $\Phi_\tau^{\exp}$ is nonnegative. This is necessary in order to apply the error bound for $\Phi_\tau^{\exp}$ on $[0, \infty)$. Specifically, we added $+1$ in the input layer, which is compensated for by the factor $\exp(1)$ in the output layer. Also, we see that the inputs of the identity network are indeed bounded in absolute value by $M = 2/\varepsilon + 1$. The error can be bounded as follows:

$$\begin{aligned}
&\|\tilde{u}_\varepsilon^{\pm} - \mathrm{R}(\Phi_\varepsilon^{\pm})\|_{L^\infty(I)} \\
&= C^{\pm} \exp(1) \| \exp(-\cdot) \circ \mathrm{Id}_{\mathbb{R}} \circ P - \mathrm{R}(\Phi_\tau^{\exp}) \circ \mathrm{R}(\Phi_{1,L-1,\delta,M}^{\mathrm{Id}}) \circ P\|_{L^\infty(I)} \\
&\leq C^{\pm} \exp(1) \left\| \big(\exp(-\cdot) \circ \mathrm{Id}_{\mathbb{R}} - \exp(-\cdot) \circ \mathrm{R}(\Phi_{1,L-1,\delta,M}^{\mathrm{Id}}) \big) \circ P \right\|_{L^\infty(I)} \\
&\quad + C^{\pm} \exp(1) \left\| \big(\exp(-\cdot) - \mathrm{R}(\Phi_\tau^{\exp}) \big) \circ \mathrm{R}(\Phi_{1,L-1,\delta,M}^{\mathrm{Id}}) \circ P \right\|_{L^\infty(I)} \\
&\leq C^{\pm} \exp(1) \, \|-\exp(-\cdot)\|_{L^\infty((0,2+2/\varepsilon))} \left\| \mathrm{Id}_{\mathbb{R}} - \mathrm{R}(\Phi_{1,L-1,\delta,M}^{\mathrm{Id}}) \right\|_{L^\infty((1,1+2/\varepsilon))} \\
&\quad + C^{\pm} \exp(1) \, \|\exp(-\cdot) - \mathrm{R}(\Phi_\tau^{\exp})\|_{L^\infty((0,2+2/\varepsilon))} \\
&\leq C^{\pm} \exp(1) (\delta + \tau) \leq C \exp(-\beta p),
\end{aligned}$$

for a constant C independent of ε and p. To obtain bounds on the error in the derivative, we first estimate

$$\begin{aligned}
&\left\| \big(\exp(-\cdot) \circ \mathrm{Id}_{\mathbb{R}} - \mathrm{R}(\Phi_\tau^{\exp}) \circ \mathrm{R}(\Phi_{1,L-1,\delta,M}^{\mathrm{Id}}) \big)' \right\|_{L^\infty((1,1+2/\varepsilon))} \\
\leq{}& \left\| \big((-\exp(-\cdot)) \circ \mathrm{Id}_{\mathbb{R}} - (-\exp(-\cdot)) \circ \mathrm{R}(\Phi_{1,L-1,\delta,M}^{\mathrm{Id}}) \big) \cdot \mathrm{Id}_{\mathbb{R}}' \right\|_{L^\infty((1,1+2/\varepsilon))} \\
&+ \left\| \big((-\exp(-\cdot)) \circ \mathrm{R}(\Phi_{1,L-1,\delta,M}^{\mathrm{Id}}) \big) \cdot \big(\mathrm{Id}_{\mathbb{R}}' - \mathrm{R}(\Phi_{1,L-1,\delta,M}^{\mathrm{Id}})' \big) \right\|_{L^\infty((1,1+2/\varepsilon))} \\
&+ \left\| \big((-\exp(-\cdot)) - \mathrm{R}(\Phi_\tau^{\exp})' \big) \circ \mathrm{R}(\Phi_{1,L-1,\delta,M}^{\mathrm{Id}}) \cdot \mathrm{R}(\Phi_{1,L-1,\delta,M}^{\mathrm{Id}})' \right\|_{L^\infty((1,1+2/\varepsilon))} \\
\leq{}& \|\exp(-\cdot)\|_{L^\infty((0,2+2/\varepsilon))} \left\| \mathrm{Id}_{\mathbb{R}} - \mathrm{R}(\Phi_{1,L-1,\delta,M}^{\mathrm{Id}}) \right\|_{L^\infty((1,1+2/\varepsilon))} \left\| \mathrm{Id}_{\mathbb{R}}' \right\|_{L^\infty((1,1+2/\varepsilon))} \\
&+ \|-\exp(-\cdot)\|_{L^\infty((0,2+2/\varepsilon))} \left\| \mathrm{Id}_{\mathbb{R}}' - \mathrm{R}(\Phi_{1,L-1,\delta,M}^{\mathrm{Id}})' \right\|_{L^\infty((1,1+2/\varepsilon))} \\
&+ \|(-\exp(-\cdot)) - \mathrm{R}(\Phi_\tau^{\exp})'\|_{L^\infty((0,2+2/\varepsilon))} \left\| \mathrm{R}(\Phi_{1,L-1,\delta,M}^{\mathrm{Id}})' \right\|_{L^\infty((1,1+2/\varepsilon))} \\
\leq{}& \delta + \delta + \tau(1 + \delta) \leq 2\delta + 2\tau,
\end{aligned}$$

and get

$$\|(\tilde{u}_\varepsilon^{\pm})' - \mathrm{R}(\Phi_\varepsilon^{\pm})'\|_{L^\infty(I)}$$

$$
\begin{aligned}
&= C^{\pm}\exp(1)\left\|\big(\exp(-\cdot)\circ\mathrm{Id}_{\mathbb{R}} - \mathrm{R}(\Phi^{\exp}_{\tau})\circ\mathrm{R}(\Phi^{\mathrm{Id}}_{1,L-1,\delta,M})\big)'\circ P\cdot P'\right\|_{L^\infty(I)}\\
&\le C^{\pm}\exp(1)\left\|\big(\exp(-\cdot)\circ\mathrm{Id}_{\mathbb{R}} - \mathrm{R}(\Phi^{\exp}_{\tau})\circ\mathrm{R}(\Phi^{\mathrm{Id}}_{1,L-1,\delta,M})\big)'\right\|_{L^\infty((1,1+2/\varepsilon))}\|P'\|_{L^\infty(I)}\\
&\le C^{\pm}\exp(1)(2\delta+2\tau)/\varepsilon \le C\exp(-\beta p),
\end{aligned}
$$

where $C>0$ again denotes a constant independent of ε and p. The depth indeed equals

$$
L(\Phi^{\pm}_{\varepsilon}) = L(\Phi^{\exp}_{\tau}) - 1 + L(\Phi^{\mathrm{Id}}_{1,L-1,\delta,M}) = 2-1+(L-1) = L.
$$

The NN size can be estimated by arguments similar to those used in the proof of Lemma 7.9, as follows. The first $L-2$ hidden layer dimensions of $\Phi^{\pm}_{\varepsilon}$ equal those of $\Phi^{\mathrm{Id}}_{1,L-1,\delta,M}$, which are bounded by C_* because of Lemma 7.9. The dimension of the last hidden layer of $\Phi^{\pm}_{\varepsilon}$ equals the dimension of the one hidden layer of $\Phi^{\exp}_{\tau}$, which is 1. As a result, all layer dimensions of $\Phi^{\pm}_{\varepsilon}$ are bounded by C_*, and the network size is bounded by $LC_*(C_*+1)\le\tilde{C}(\log_2(p)+1)$ for some constant $\tilde{C}>0$ independent of f, p, ε, C and β.

The last term u^R_{ε} in (2.10) is small and can be neglected. As shown in the last step in the proof of [13, Theorem 16], there exist constants $C,\beta>0$, independent of p and ε, such that

$$
\|u^R_{\varepsilon}\|_{W^{1,\infty}(I)} \le C\exp(-\beta p).
$$

Finally, we use Proposition 7.6 to add the subnetworks and define

$$
\Phi^{u_\varepsilon,p}_{\varepsilon} := \Phi^{u^S_\varepsilon,p} + \Phi^{+}_{\varepsilon} + \Phi^{-}_{\varepsilon}.
$$

It satisfies the error bound

$$
\begin{aligned}
&\|u_\varepsilon - \mathrm{R}(\Phi^{u_\varepsilon,p}_{\varepsilon})\|_{W^{1,\infty}(I)}\\
&\le \|u^S_\varepsilon - \mathrm{R}(\Phi^{u^S_\varepsilon,p})\|_{W^{1,\infty}(I)} + \|\tilde{u}^+_\varepsilon - \mathrm{R}(\Phi^+_\varepsilon)\|_{W^{1,\infty}(I)} + \|\tilde{u}^-_\varepsilon - \mathrm{R}(\Phi^-_\varepsilon)\|_{W^{1,\infty}(I)}\\
&\quad + \|u^R_\varepsilon\|_{W^{1,\infty}(I)}\\
&\le Ce^{-\beta p}.
\end{aligned}
$$

Moreover, $L(\Phi^{u_\varepsilon,p}_{\varepsilon}) = \lceil\log_2(p)\rceil + 1$ and

$$
\begin{aligned}
M(\Phi^{u_\varepsilon,p}_{\varepsilon}) &\le M(\Phi^{u^S_\varepsilon,p}) + M(\Phi^+_\varepsilon) + M(\Phi^-_\varepsilon)\\
&\le \tilde{C}p + \tilde{C}(\log_2(p)+1) + \tilde{C}(\log_2(p)+1) \le \tilde{C}p,
\end{aligned}
$$

for a constant $\tilde{C}>0$ independent of f, p, ε, C and β.

The hidden layer weights and biases of $\Phi^{u^S_\varepsilon,p}$ from Proposition 7.12 are independent of u^S_ε and depend only on p. The hidden layer weights and biases of $\Phi^{\pm}_{\varepsilon}$ only depend on ε, p and β. □

Remark 7.13. *Exact boundary conditions can be imposed by slightly adjusting the constants $C^{\pm}$ in the formula for the boundary layers. As the approximation error is exponentially small in $L^{\infty}(I)$, also the necessary change in $C^{\pm}$ is exponentially small, and so is the additional error in $\varepsilon^{-1/2}\|\circ\|_{L^2(I)}$, $\|\circ\|_{L^\infty(I)}$ and $\varepsilon^{1/2}\|\circ\|_{H^1(I)}$.*

8 Conclusions and Generalizations

We summarize the principal findings of the present paper, i.e., robust expression rate bounds for solutions u^{ε} of the model singular perturbation problem in Section 2 by several classes of DNNs, with either ReLU, tanh or sigmoid activation, or of spiking type.

For the model singular perturbation problem in Section 2 we established robust w.r. to ε exponential expression rate bounds for the solution u_{ε} of (2.1)–(2.2) by deep neural networks. We considered in detail several types and architectures of DNNs, and the impact of architecture and activation on the approximation rates.

The robust, exponential expression rate bounds proved in Section 5 for strict ReLU NNs in Propositions 5.2 and 5.3 implied, with a ReLU NN-to-spiking NN conversion algorithm from [23], corresponding expression rate bounds also for so-called spiking NNs, in Section 6.

We also proved in a particular case of (2.1)–(2.2) with exponential boundary layer functions in Section 7 that tanh()-activated NNs provide better (still robust, exponential) expression rates. In Section B, we show the same result for sigmoid NNs.

These results indicate that *in order to resolve multivariate exponential boundary layers, deep NNs with* tanh *or sigmoid activations afford expression rates which are uniform w.r. to the length-scale parameter.* With these activations in particular, boundary layer resolution in deep NN approximations of PDE solutions does not require "augmentation" of the DNN feature space with analytic boundary layers as proposed e.g. in the recent [2] and in the references there. See e.g. [1] for a recent computational approach which does not rely on such augmentations.

In the proofs in Section 7, use was made of novel tanh-NN emulation rate bounds for Chebyšev polynomials, which are of independent interest, with their lengthy proofs relegated to Appendix A.

Appendix A tanh Emulation of Univariate Chebyšev Polynomials

A key step in the expression rate analysis is the analysis of tanh-expression rates of Chebyšev polynomials. As these rates are, due to the wide use of Chebyšev polynomials in spectral methods and regression (e.g. [17, 19]), of independent interest, we provide a detailed analysis. We also recall that corresponding results for strict ReLU NNs have been obtained in [17]. In this appendix, we maintain all notations from the main text.

For $m \in \mathbb{N}$, $m \geq 2$, we construct in Definition A.1 below a tanh NN $\Psi_{\delta}^{\mathrm{Cheb},m}$ approximating all univariate Chebyšev polynomials of degree $1,\ldots,m$ with $W^{1,\infty}((-1,1))$-error at most $\delta \in (0,1)$. We denote by T_k, $k \in \mathbb{N}_0$ the Chebyšev polynomial of degree k (of the first kind), normalized such that $T_k(1) = 1$ for all $k \in \mathbb{N}_0$.

As in [15, Proposition 7.2.2] (see also [8, Appendix A]), the network has a binary tree structure, similar to the one in [16, Proposition 4.2]. It is based on the recursion

$$T_{m+n} = 2T_m T_n - T_{|m-n|}, \quad T_0(x) = 1, \quad T_1(x) = x, \qquad \text{for all } x \in \mathbb{R} \text{ and } m, n \in \mathbb{N}_0.$$

This recursion is related to the addition rule for cosines and was first used for NN construction in [24].

The construction uses the identity and product networks from Lemma 7.9. In Lemma 7.9, the size of product networks and identity networks is independent of the desired accuracy. This allows us to make minor simplifications to the construction that was used in the context of ReLU NNs in [15, 16].

Definition A.1. *For all* $\delta \in (0,1)$ *we define*

$$\Psi_\delta^{\mathrm{Cheb},2} := ((A_{\delta,2}^{\mathrm{Cheb},2}, b_{\delta,2}^{\mathrm{Cheb},2}, \mathrm{Id}_{\mathbb{R}^2})) \bullet \mathrm{FP}(\Phi_{1,2,\delta,1}^{\mathrm{Id}}, \Phi_{\delta/4,1}^{\mathrm{Prod}}) \bullet ((A_{\delta,1}^{\mathrm{Cheb},2}, b_{\delta,1}^{\mathrm{Cheb},2}, \mathrm{Id}_{\mathbb{R}^3}))$$

where $A_{\delta,2}^{\mathrm{Cheb},2} := \mathrm{diag}(1,2) \in \mathbb{R}^{2\times 2}$ *is a diagonal matrix with diagonal entries* 1 *and* 2*,* $b_{\delta,2}^{\mathrm{Cheb},2} := (0,-1)^\top \in \mathbb{R}^2$*,* $A_{\delta,1}^{\mathrm{Cheb},2} := (1,1,1)^\top \in \mathbb{R}^{3\times 1}$ *and* $b_{\delta,1}^{\mathrm{Cheb},2} := 0 \in \mathbb{R}^3$*. Its realization is*

$$\mathrm{R}(\Psi_\delta^{\mathrm{Cheb},2}) : \mathbb{R} \to \mathbb{R}^2 : x \mapsto \big(\mathrm{R}(\Phi_{1,2,\delta,1}^{\mathrm{Id}})(x), 2\mathrm{R}(\Phi_{\delta/4,1}^{\mathrm{Prod}})(x,x) - 1\big).$$

To define $\Psi_\delta^{\mathrm{Cheb},m}$*, for* $m \in \mathbb{N}$ *satisfying* $m > 2$*, let* $\tilde{m} := \min\{2^k : 2^k \geq m, k \in \mathbb{N}\} < 2m$*. This means that there exists* $k \in \mathbb{N}$ *such that* $\tilde{m} = 2^k$*. Then* $2^{k-1} = \tilde{m}/2 \geq 2$ *implies that* $k \geq 2$*, which implies that* $\tilde{m}/2 = 2^{k-1}$ *is an even number.*

Let $\theta := \delta/(4m^2)$ *and note that* $4m^2 \geq 36$*. We use the following auxiliary matrices and vectors. Let* $A_{\delta,1}^{\mathrm{Cheb},m} \in \mathbb{R}^{(2m-\tilde{m}/2)\times(\tilde{m}/2)}$ *be defined by*

$$(A_{\delta,1}^{\mathrm{Cheb},m})_{ij} := \begin{cases} 1 & \text{if } i = j \leq \tilde{m}/2, \\ 1 & \text{if } i > \tilde{m}/2 \text{ and } j = \tilde{m}/4 + \lceil \frac{i-1-\tilde{m}/2}{4} \rceil, \\ 0 & \text{else,} \end{cases}$$

and let $b_{\delta,1}^{\mathrm{Cheb},m} := 0 \in \mathbb{R}^{2m-\tilde{m}/2}$*. In addition, let* $A_{\delta,2}^{\mathrm{Cheb},m} \in \mathbb{R}^{m\times m}$ *and* $b_{\delta,2}^{\mathrm{Cheb},m} \in \mathbb{R}^m$ *be defined by*

$$(A_{\delta,2}^{\mathrm{Cheb},m})_{ij} := \begin{cases} 1 & \text{if } i = j \leq \tilde{m}/2, \\ 2 & \text{if } i = j > \tilde{m}/2, \\ -1 & \text{if } i > \tilde{m}/2 \text{ is odd and } j = 1, \\ 0 & \text{else.} \end{cases}$$

$$(b_{\delta,2}^{\mathrm{Cheb},m})_i := \begin{cases} -1 & \text{if } i > \tilde{m}/2 \text{ is even,} \\ 0 & \text{else.} \end{cases}$$

Then, we recursively define

$$\Psi_\delta^{\mathrm{Cheb},m} := ((A_{\delta,2}^{\mathrm{Cheb},m}, b_{\delta,2}^{\mathrm{Cheb},m}, \mathrm{Id}_{\mathbb{R}^m})) \bullet \mathrm{FP}(\Phi^{\mathrm{Id}}_{\tilde{m}/2,2,\theta,2}, \Phi^{\mathrm{Prod}}_{\theta,2}, \dots, \Phi^{\mathrm{Prod}}_{\theta,2}) \bullet ((A_{\delta,1}^{\mathrm{Cheb},m}, b_{\delta,1}^{\mathrm{Cheb},m}, \mathrm{Id}_{\mathbb{R}^{2m-\tilde{m}/2}})) \bullet \Psi_\theta^{\mathrm{Cheb},\tilde{m}/2}, \tag{A1}$$

where the full parallelization contains $m - \tilde{m}/2$ product networks. In the remainder of this definition, we will abbreviate $\Psi_\delta^m := \Psi_\delta^{\mathrm{Cheb},m}$ and $\Psi_\theta^{\tilde{m}/2} := \Psi_\theta^{\mathrm{Cheb},\tilde{m}/2}$. By construction, the realization $\mathrm{R}(\Psi_\delta^m) : \mathbb{R} \to \mathbb{R}^m$ satisfies

$$\begin{aligned}
\mathrm{R}(\Psi_\delta^m)_j &= \mathrm{R}(\Phi^{\mathrm{Id}}_{1,2,\theta,2}) \circ \mathrm{R}(\Psi_\theta^{\tilde{m}/2})_j, \qquad \text{if } j \le \tilde{m}/2,\\
\mathrm{R}(\Psi_\delta^m)_j &= 2\mathrm{R}(\Phi^{\mathrm{Prod}}_{\theta,2})(\mathrm{R}(\Psi_\theta^{\tilde{m}/2})_{\lfloor j/2 \rfloor}, \mathrm{R}(\Psi_\theta^{\tilde{m}/2})_{\lceil j/2 \rceil}) - 1, \qquad \text{if } j > \tilde{m}/2 \text{ is even},\\
\mathrm{R}(\Psi_\delta^m)_j &= 2\mathrm{R}(\Phi^{\mathrm{Prod}}_{\theta,2})(\mathrm{R}(\Psi_\theta^{\tilde{m}/2})_{\lfloor j/2 \rfloor}, \mathrm{R}(\Psi_\theta^{\tilde{m}/2})_{\lceil j/2 \rceil}) - \mathrm{R}(\Phi^{\mathrm{Id}}_{1,2,\theta,2}) \circ \mathrm{R}(\Psi_\theta^{\tilde{m}/2})_1,\\
&\qquad \text{if } j > \tilde{m}/2 \text{ is odd}.
\end{aligned}$$

We used that by Definition 7.10 the subnetwork $\Phi^{\mathrm{Id}}_{\tilde{m}/2,2,\theta,2}$ is the full parallelization of $\tilde{m}/2$ networks $\Phi^{\mathrm{Id}}_{1,2,\theta,2}$, thus $\mathrm{R}(\Phi^{\mathrm{Id}}_{\tilde{m}/2,2,\theta,2})(x)_j = \mathrm{R}(\Phi^{\mathrm{Id}}_{1,2,\theta,2})(x_j)$.

Properties of the NNs in Definition A.1 are as follows.

Proposition A.2. *For all $m \in \mathbb{N}$, $m \ge 2$ and for every $\delta \in (0,1)$ the NN $\Psi_\delta^{\mathrm{Cheb},m}$ in Definition A.1 satisfies*

$$\|T_k - \mathrm{R}(\Psi_\delta^{\mathrm{Cheb},m})_k\|_{W^{1,\infty}((-1,1))} \le \delta, \qquad \text{for all } k = 1, \dots, m,$$

and, for some constant C which is independent of m and δ,

$$L(\Psi_\delta^{\mathrm{Cheb},m}) = \lceil \log_2(m) \rceil + 1 \ , \quad \text{and} \ M(\Psi_\delta^{\mathrm{Cheb},m}) \le Cm.$$

Proof. This proof is by induction with respect to the number of hidden layers, which equals $\lceil \log_2(m) \rceil$. In Step 1, we treat the case $m = 2$, for which we will use one hidden layer. Then, for $m > 2$ and $\tilde{m} := \min\{2^k : 2^k \ge m, k \in \mathbb{N}\}$, assuming that the result has been shown for $\tilde{m}/2 < m$, we prove the statements for m, increasing the number of hidden layers by one. In Step 2, we give the error estimates, and in Step 3, we analyze the network depth and size.

Step 1. For $m = 2$, we have

$$\|T_1 - \mathrm{R}(\Psi_\delta^{\mathrm{Cheb},2})_1\|_{W^{1,\infty}((-1,1))} = \|\mathrm{Id}_\mathbb{R} - \mathrm{R}(\Phi^{\mathrm{Id}}_{1,2,\delta,1})\|_{W^{1,\infty}((-1,1))} \le \delta,$$

as well as

$$\begin{aligned}
\|T_2 - \mathrm{R}(\Psi_\delta^{\mathrm{Cheb},2})_2\|_{L^\infty((-1,1))} &= \|(2x^2 - 1) - \big(2\mathrm{R}(\Phi^{\mathrm{Prod}}_{\delta/4,1})(x,x) - 1\big)\|_{L^\infty((-1,1))}\\
&= 2\|x^2 - \mathrm{R}(\Phi^{\mathrm{Prod}}_{\delta/4,1})(x,x)\|_{L^\infty((-1,1))} \le 2\delta/4 \le \delta
\end{aligned}$$

and

$$\begin{aligned}
&\|T_2' - \mathrm{R}(\Psi_\delta^{\mathrm{Cheb},2})_2'\|_{L^\infty((-1,1))}\\
&= \|(2x^2-1)' - \big(2\mathrm{R}(\Phi^{\mathrm{Prod}}_{\delta/4,1})(x,x) - 1\big)'\|_{L^\infty((-1,1))}\\
&\leq \|4x - 2[D\mathrm{R}(\Phi^{\mathrm{Prod}}_{\delta/4,1})]_1(x,x) - 2[D\mathrm{R}(\Phi^{\mathrm{Prod}}_{\delta/4,1})]_2(x,x)\|_{L^\infty((-1,1))}\\
&\leq 2\delta/4 + 2\delta/4 \leq \delta,
\end{aligned}$$

where $[D\mathrm{R}(\Phi^{\mathrm{Prod}}_{\delta/4,1})]_j$ denotes the derivative with respect to the j-th argument. For the depth we obtain with repeated application of the formula for the depth from Definition 7.8 that $L(\Psi_\delta^{\mathrm{Cheb},2}) = 1-1+2-1+1 = 2$. To estimate the network size, let $C_* > 0$ be as in Lemma 7.9, such that $M(\Phi^{\mathrm{Id}}_{1,2,\delta,1}) \leq 2C_*$ and $M(\Phi^{\mathrm{Prod}}_{\delta/4,1}) \leq 2C_*$. The number of neurons in the hidden layer of $\Psi_\delta^{\mathrm{Cheb},2}$ equals that of $\mathrm{FP}(\Phi^{\mathrm{Id}}_{1,2,\delta,1}, \Phi^{\mathrm{Prod}}_{\delta/4,1})$, which is at most $2\cdot 2C_*$. Thus, in the notation of Definition 4.1 we have $N_0 = 1$, $N_1 \leq 4C_*$ and $N_2 = 2$, which gives $M(\Psi_\delta^{\mathrm{Cheb},2}) \leq N_1(N_0+1) + N_2(N_1+1) \leq 4C_*\cdot 2 + 2(4C_*+1)$, which is a constant independent of δ. Below, we will use that for $m = 2$ the number of nonzero weights and biases in the last layer is bounded by $5C_*m$. This follows from $N_2(N_1+1) = 2(4C_*+1) \leq 10C_*$, using that w.l.o.g. $C_* \geq 1$ (recall that C_* is an upper bound on the size of identity networks and product networks). Similarly, $M(\Psi_\delta^{\mathrm{Cheb},2}) \leq 4C_*\cdot 2 + 2(4C_*+1) \leq 18C_* \leq 9C_*m$.

Step 2. Let now $m \in \mathbb{N}$, $m > 2$.

Step 2a. We first estimate the emulation error in the $L^\infty(-1,1)$-norm. The arguments which we use closely follow the proof of [15, Proposition 7.2.2]. We use the shorthand notation $\Psi_\delta^m := \Psi_\delta^{\mathrm{Cheb},m}$ and $\Psi_\theta^{\tilde m/2} := \Psi_\theta^{\mathrm{Cheb},\tilde m/2}$ already used in Definition A.1. From $\|T_k\|_{L^\infty((-1,1))} = 1$ and $\|T_k - \mathrm{R}(\Psi_\theta^{\tilde m/2})_k\|_{L^\infty((-1,1))} \leq \theta$ we obtain that $\|\mathrm{R}(\Psi_\theta^{\tilde m/2})_k\|_{L^\infty((-1,1))} \leq 1+\theta \leq 2$, which means that the inputs of the identity networks and product networks in (A1) are indeed bounded in absolute value by 2, as is necessary in order to apply the error bounds from Lemma 7.9. Also, we will use that each component of $\mathrm{R}(\Phi^{\mathrm{Id}}_{\tilde m/2,2,\theta,2})$ equals $\mathrm{R}(\Phi^{\mathrm{Id}}_{1,2,\theta,2})$, see Definition 7.10. To simplify the notation, for all $k = 1,\ldots,\tilde m/2$ within this proof we will abbreviate $\widetilde T_k = \mathrm{R}(\Psi_\theta^{\tilde m/2})_k$. Now, for $j \leq \tilde m/2$,

$$\begin{aligned}
\|T_j - \mathrm{R}(\Psi_\delta^m)_j\|_{L^\infty((-1,1))} &= \|T_j - \mathrm{R}(\Phi^{\mathrm{Id}}_{1,2,\theta,2}) \circ \widetilde T_j\|_{L^\infty((-1,1))}\\
&\leq \|T_j - \widetilde T_j\|_{L^\infty((-1,1))} + \|(\mathrm{Id}_{\mathbb{R}} - \mathrm{R}(\Phi^{\mathrm{Id}}_{1,2,\theta,2})) \circ \widetilde T_j\|_{L^\infty((-1,1))}\\
&\leq \|T_j - \widetilde T_j\|_{L^\infty((-1,1))} + \|\mathrm{Id}_{\mathbb{R}} - \mathrm{R}(\Phi^{\mathrm{Id}}_{1,2,\theta,2})\|_{L^\infty((-2,2))}\\
&\leq \theta + \theta = 2\theta \leq \delta.
\end{aligned}$$

For even $j > \tilde m/2$, we note that the terms -1 in T_j and $\mathrm{R}(\Psi_\delta^m)_j$ cancel, and obtain

$$\begin{aligned}
\|T_j - \mathrm{R}(\Psi_\delta^m)_j\|_{L^\infty((-1,1))} &\leq \|2T_{\lfloor j/2\rfloor}T_{\lceil j/2\rceil} - 2\widetilde T_{\lfloor j/2\rfloor}\widetilde T_{\lceil j/2\rceil}\|_{L^\infty((-1,1))}\\
&\quad + \|2\widetilde T_{\lfloor j/2\rfloor}\widetilde T_{\lceil j/2\rceil} - 2\mathrm{R}(\Phi^{\mathrm{Prod}}_{\theta,2})(\widetilde T_{\lfloor j/2\rfloor}, \widetilde T_{\lceil j/2\rceil})\|_{L^\infty((-1,1))}\\
&\leq 2\|T_{\lfloor j/2\rfloor} - \widetilde T_{\lfloor j/2\rfloor}\|_{L^\infty((-1,1))}\|T_{\lceil j/2\rceil}\|_{L^\infty((-1,1))}
\end{aligned}$$

$$
\begin{aligned}
&+ 2\|\widetilde{T}_{\lfloor j/2\rfloor}\|_{L^\infty((-1,1))}\|T_{\lceil j/2\rceil} - \widetilde{T}_{\lceil j/2\rceil}\|_{L^\infty((-1,1))} \\
&+ 2\theta \\
\leq\, & 2\cdot\theta\cdot 1 + 2\cdot 2\cdot\theta + 2\cdot\theta = 8\theta \leq \delta.
\end{aligned}
$$

For odd $j > \tilde{m}/2$

$$
\begin{aligned}
\|T_j - \mathrm{R}(\Psi_\delta^m)_j\|_{L^\infty((-1,1))} \leq\, & \|T_1 - \mathrm{R}(\Phi^{\mathrm{Id}}_{1,2,\theta,2}) \circ \widetilde{T}_1\|_{L^\infty((-1,1))} \\
&+ \|2T_{\lfloor j/2\rfloor}T_{\lceil j/2\rceil} - 2\widetilde{T}_{\lfloor j/2\rfloor}\widetilde{T}_{\lceil j/2\rceil}\|_{L^\infty((-1,1))} \\
&+ \|2\widetilde{T}_{\lfloor j/2\rfloor}\widetilde{T}_{\lceil j/2\rceil} - 2\mathrm{R}(\Phi^{\mathrm{Prod}}_{\theta,2})(\widetilde{T}_{\lfloor j/2\rfloor}, \widetilde{T}_{\lceil j/2\rceil})\|_{L^\infty((-1,1))} \\
\leq\, & 2\theta + 8\theta = 10\theta \leq \delta.
\end{aligned}
$$

Step 2b. We use the same notation as in Step 2a. To derive $W^{1,\infty}((-1,1))$-bounds, we first recall that $\|T_k'\|_{L^\infty((-1,1))} = k^2$ for all $k \in \mathbb{N}$. The fact that $\|T_k'\|_{L^\infty((-1,1))} \geq k^2$ follows from $T_m' = mU_{m-1}$, which is the Chebyšev polynomial of the second kind of degree $m-1$, and $U_{m-1}(1) = m$, which is [6, Section 1.5.1]. The opposite inequality $\|T_k'\|_{L^\infty((-1,1))} \leq k^2$ is Markov's inequality, which holds for all polynomials of degree at most k. It follows that for all $k = 1, \dots, \tilde{m}/2$ there holds $\|\widetilde{T}_k'\|_{L^\infty((-1,1))} \leq \|T_k'\|_{L^\infty((-1,1))} + \|T_k' - \widetilde{T}_k'\|_{L^\infty((-1,1))} \leq k^2 + \theta \leq k^2 + 1$. Now, for $j \leq \tilde{m}/2$,

$$
\begin{aligned}
&\|T_j' - \mathrm{R}(\Psi_\delta^m)_j'\|_{L^\infty((-1,1))} \\
\leq\, & \|T_j' - \widetilde{T}_j'\|_{L^\infty((-1,1))} + \left\|\big((\mathrm{Id}_{\mathbb{R}} - \mathrm{R}(\Phi^{\mathrm{Id}}_{1,2,\theta,2})) \circ \widetilde{T}_j\big)'\right\|_{L^\infty((-1,1))} \\
\leq\, & \|T_j' - \widetilde{T}_j'\|_{L^\infty((-1,1))} + \|\mathrm{Id}_{\mathbb{R}}' - \mathrm{R}(\Phi^{\mathrm{Id}}_{1,2,\theta,2})'\|_{L^\infty((-2,2))}\|\widetilde{T}_j'\|_{L^\infty((-1,1))} \\
\leq\, & \theta + \theta\cdot(j^2+1) = (j^2+2)\theta \leq 3j^2\theta \leq \delta.
\end{aligned}
$$

For even $j > \tilde{m}/2$, the terms -1 in T_j and $\mathrm{R}(\Psi_\delta^m)_j$ cancel, and we obtain

$$
\begin{aligned}
&\|T_j' - \mathrm{R}(\Psi_\delta^m)_j'\|_{L^\infty((-1,1))} \\
\leq\, & \left\|2T_{\lfloor j/2\rfloor}'T_{\lceil j/2\rceil} - 2[D\mathrm{R}(\Phi^{\mathrm{Prod}}_{\theta,2})]_1(\widetilde{T}_{\lfloor j/2\rfloor}, \widetilde{T}_{\lceil j/2\rceil})\widetilde{T}_{\lfloor j/2\rfloor}'\right\|_{L^\infty((-1,1))} \\
&+ \left\|2T_{\lfloor j/2\rfloor}T_{\lceil j/2\rceil}' - 2[D\mathrm{R}(\Phi^{\mathrm{Prod}}_{\theta,2})]_2(\widetilde{T}_{\lfloor j/2\rfloor}, \widetilde{T}_{\lceil j/2\rceil})\widetilde{T}_{\lceil j/2\rceil}'\right\|_{L^\infty((-1,1))} \\
\leq\, & \left\|2T_{\lfloor j/2\rfloor}'(T_{\lceil j/2\rceil} - \widetilde{T}_{\lceil j/2\rceil})\right\|_{L^\infty((-1,1))} \\
&+ \left\|2T_{\lfloor j/2\rfloor}' - \widetilde{T}_{\lfloor j/2\rfloor}')\widetilde{T}_{\lceil j/2\rceil}\right\|_{L^\infty((-1,1))} \\
&+ \left\|2\big(\widetilde{T}_{\lceil j/2\rceil} - [D\mathrm{R}(\Phi^{\mathrm{Prod}}_{\theta,2})]_1(\widetilde{T}_{\lfloor j/2\rfloor}, \widetilde{T}_{\lceil j/2\rceil})\big)\widetilde{T}_{\lfloor j/2\rfloor}'\right\|_{L^\infty((-1,1))} \\
&+ \left\|2(T_{\lfloor j/2\rfloor} - \widetilde{T}_{\lfloor j/2\rfloor})T_{\lceil j/2\rceil}'\right\|_{L^\infty((-1,1))} \\
&+ \left\|2\widetilde{T}_{\lfloor j/2\rfloor}(T_{\lceil j/2\rceil}' - \widetilde{T}_{\lceil j/2\rceil}')\right\|_{L^\infty((-1,1))}
\end{aligned}
$$

$$
\begin{aligned}
&+ \left\| 2\big(\widetilde{T}_{\lfloor j/2 \rfloor} - [D\mathrm{R}(\Phi^{\mathrm{Prod}}_{\theta,2})]_2(\widetilde{T}_{\lfloor j/2 \rfloor}, \widetilde{T}_{\lceil j/2 \rceil})\big)\widetilde{T}'_{\lceil j/2 \rceil} \right\|_{L^\infty((-1,1))} \\
&\le 2\lfloor j/2 \rfloor^2 \cdot \theta + 2\theta \cdot 2 + 2\theta \cdot (\lfloor j/2 \rfloor^2 + 1) + 2\theta \cdot \lceil j/2 \rceil^2 + 2 \cdot 2 \cdot \theta + 2\theta \cdot (\lceil j/2 \rceil^2 + 1) \\
&= (4\lfloor j/2 \rfloor^2 + 4\lceil j/2 \rceil^2 + 12)\theta = (4(\lfloor j/2 \rfloor + \lceil j/2 \rceil)^2 - 8\lfloor j/2 \rfloor \lceil j/2 \rceil + 12)\theta \\
&= (4j^2 - 8\lfloor j/2 \rfloor \lceil j/2 \rceil + 12)\theta \\
&\le 4j^2\theta \le \delta,
\end{aligned}
$$

where $[D\mathrm{R}(\Phi^{\mathrm{Prod}}_{\theta,2})]_\ell$ denotes the derivative with respect to the ℓ-th argument. We used that $j > \tilde{m}/2 \ge 2$ and thus $\lfloor j/2 \rfloor \ge 1$ and $\lceil j/2 \rceil \ge 2$, such that $8\lfloor j/2 \rfloor \lceil j/2 \rceil \ge 16$. For odd $j > \tilde{m}/2$

$$
\begin{aligned}
&\|T'_j - \mathrm{R}(\Psi^m_\delta)'_j\|_{L^\infty((-1,1))} \\
&\le \left\| T'_1 - \big(\mathrm{R}(\Phi^{\mathrm{Id}}_{1,2,\theta,2}) \circ \widetilde{T}_1\big)' \right\|_{L^\infty((-1,1))} \\
&\quad + \left\| 2T'_{\lfloor j/2 \rfloor} T_{\lceil j/2 \rceil} - 2[D\mathrm{R}(\Phi^{\mathrm{Prod}}_{\theta,2})]_1(\widetilde{T}_{\lfloor j/2 \rfloor}, \widetilde{T}_{\lceil j/2 \rceil})\widetilde{T}'_{\lfloor j/2 \rfloor} \right\|_{L^\infty((-1,1))} \\
&\quad + \left\| 2T_{\lfloor j/2 \rfloor} T'_{\lceil j/2 \rceil} - 2[D\mathrm{R}(\Phi^{\mathrm{Prod}}_{\theta,2})]_2(\widetilde{T}_{\lfloor j/2 \rfloor}, \widetilde{T}_{\lceil j/2 \rceil})\widetilde{T}'_{\lceil j/2 \rceil} \right\|_{L^\infty((-1,1))} \\
&\le (1^2 + 2)\theta + (4j^2 - 8\lfloor j/2 \rfloor \lceil j/2 \rceil + 12)\theta = (4j^2 - 8\lfloor j/2 \rfloor \lceil j/2 \rceil + 15)\theta \\
&\le 4j^2\theta \le \delta.
\end{aligned}
$$

Step 3. We again consider $m \in \mathbb{N}$, $m > 2$.

Step 3a. To determine the depth, we obtain by repeated use of the formula for the depth from Definition 7.8 that for all $m \in \mathbb{N}$, $m > 2$ holds $L(\Psi^{\mathrm{Cheb},m}_\delta) = 1 - 1 + 2 - 1 + 1 - 1 + L(\Psi^{\mathrm{Cheb},\tilde{m}/2}_\theta) = 1 + L(\Psi^{\mathrm{Cheb},\tilde{m}/2}_\theta)$. Together with $L(\Psi^{\mathrm{Cheb},2}_\delta) = 2$, this implies that for all $m \in 2^{\mathbb{N}}$ we have $L(\Psi^{\mathrm{Cheb},m}_\delta) = \log_2(m) + 1$, because $\tilde{m} = m$ for all $m \in 2^{\mathbb{N}}$. For general $m \in \mathbb{N}$, $m \ge 2$ we obtain $L(\Psi^{\mathrm{Cheb},m}_\delta) = 1 + (\log_2(\tilde{m}/2) + 1) = \log_2(\tilde{m}) + 1 = \lceil \log_2(m) \rceil + 1$.

Step 3b. To estimate the network size, again let $C_* > 0$ be as in Lemma 7.9, such that $M(\Phi^{\mathrm{Id}}_{1,2,\theta,2}) \le 2C_*$ and $M(\Phi^{\mathrm{Prod}}_{\theta,2}) \le 2C_*$. By Definition 7.10, it holds that

$$
\mathrm{FP}(\Phi^{\mathrm{Id}}_{\tilde{m}/2,2,\theta,2}, \Phi^{\mathrm{Prod}}_{\theta,2}, \ldots, \Phi^{\mathrm{Prod}}_{\theta,2}) = \mathrm{FP}(\Phi^{\mathrm{Id}}_{1,2,\theta,2}, \ldots, \Phi^{\mathrm{Id}}_{1,2,\theta,2}, \Phi^{\mathrm{Prod}}_{\theta,2}, \ldots, \Phi^{\mathrm{Prod}}_{\theta,2}),
$$

which has depth 2 and contains $\tilde{m}/2$ identity networks and $m - \tilde{m}/2$ product networks. This full parallelization is defined by repeated application of Proposition 7.7, from which we see that its weight matrices are block diagonal matrices. The number of nonzero coefficients in each submatrix is bounded from above by the size of the subnetwork of which it is part, which is at most $2C_*$. In particular, each row and each column contain at most $2C_*$ nonzero coefficients. From the definition of $A^{\mathrm{Cheb},m}_{\delta,2}$, we see that $\|A^{\mathrm{Cheb},m}_{\delta,2}\|_0 \le 2m$. Denoting the last layer weight matrix and bias vector of $\mathrm{FP}(\Phi^{\mathrm{Id}}_{\tilde{m}/2,2,\theta,2}, \Phi^{\mathrm{Prod}}_{\theta,2}, \ldots, \Phi^{\mathrm{Prod}}_{\theta,2})$ by A_2 and b_2, respectively, those of $((A^{\mathrm{Cheb},m}_{\delta,2}, b^{\mathrm{Cheb},m}_{\delta,2}, \mathrm{Id}_{\mathbb{R}^m})) \bullet \mathrm{FP}(\Phi^{\mathrm{Id}}_{\tilde{m}/2,2,\theta,2}, \Phi^{\mathrm{Prod}}_{\theta,2}, \ldots, \Phi^{\mathrm{Prod}}_{\theta,2})$ equal $A^{\mathrm{Cheb},m}_{\delta,2} A_2$ and

$A_{\delta,2}^{\mathrm{Cheb},m} b_2 + b_{\delta,2}^{\mathrm{Cheb},m}$. Because in the matrix multiplication $A_{\delta,2}^{\mathrm{Cheb},m} A_2$ each element of $A_{\delta,2}^{\mathrm{Cheb},m}$ gets multiplied with all elements of a row of A_2, of which at most $2C_*$ are nonzero, we obtain that $\|A_{\delta,2}^{\mathrm{Cheb},m} A_2\|_0 \leq \|A_{\delta,2}^{\mathrm{Cheb},m}\|_0 2C_* \leq 2m2C_* = 4C_*m$. Also, $\|A_{\delta,2}^{\mathrm{Cheb},m} b_2 + b_{\delta,2}^{\mathrm{Cheb},m}\|_0 \leq m$ because this is a vector in $\mathbb{R}^m$, so the total number of nonzero coefficients in the last layer is bounded by $(4C_* + 1)m \leq 5C_*m$ (w.l.o.g. $C_* \geq 1$). Denoting the first layer weight matrix and bias vector of $\mathrm{FP}(\Phi^{\mathrm{Id}}_{\tilde{m}/2,2,\theta,2}, \Phi^{\mathrm{Prod}}_{\theta,2}, \ldots, \Phi^{\mathrm{Prod}}_{\theta,2})$ by A_1 and b_1, respectively, recall that each row and each column of A_1 contain at most $2C_*$ nonzero coefficients. From the definition of $A_{\delta,1}^{\mathrm{Cheb},m}$, we see that each column has at most 5 nonzero coefficients, namely one for which $i = j$ and at most four for which $i > \tilde{m}/2$ and $j = \tilde{m}/4 + \lceil \frac{i-1-\tilde{m}/2}{4} \rceil$. The j-th column of $A_1 A_{\delta,1}^{\mathrm{Cheb},m}$ is the sum of columns of A_1 multiplied with elements of the j-th column of $A_{\delta,1}^{\mathrm{Cheb},m}$. Each column of A_1 contains at most $2C_*$ nonzero coefficients and each column of $A_{\delta,1}^{\mathrm{Cheb},m}$ at most 5, thus each column of $A_1 A_{\delta,1}^{\mathrm{Cheb},m}$ at most $10C_*$. Now, the weight matrix in the second to last layer of Ψ^m_δ is the product of $A_1 A_{\delta,1}^{\mathrm{Cheb},m}$ with the weight matrix of the last layer of $\Psi_\theta^{\tilde{m}/2}$. Each element of that matrix is multiplied with all coefficients in one column of $A_1 A_{\delta,1}^{\mathrm{Cheb},m}$, of which at most $10C_*$ are nonzero. As shown above, the number of nonzero weights and biases in the last layer of $\Psi_\theta^{\tilde{m}/2}$ is at most $5C_*\tilde{m}/2 \leq 5C_*m$, which means that the total number of nonzero weights in the second to last layer of Ψ^m_δ is at most $50C_*^2 m$. The bias vector in that layer has $2m - \tilde{m}/2 \leq 2m$ entries, thus the total number of nonzero weights and biases in the second to last layer of Ψ^m_δ is at most $52C_*^2 m$ (w.l.o.g. $C_* \geq 1$). All layers of Ψ^m_δ except the last two are identical to those of $\Psi_\theta^{\tilde{m}/2}$. Thus, we find that $M(\Psi^m_\delta) \leq 5C_*m + 52C_*^2 m + M(\Psi_\theta^{\tilde{m}/2}) \leq 57C_*^2 m + M(\Psi_\theta^{\tilde{m}/2})$. For all $m \in 2^{\mathbb{N}}$, it holds that $\tilde{m} = m$ and we find by induction that $M(\Psi^m_\delta) \leq 114C_*^2 m$, because $M(\Psi^m_\delta) \leq 57C_*^2 m + 114C_*^2(\tilde{m}/2) = 114C_*^2 m$. For general $m \in \mathbb{N}$, $m > 2$, we obtain $M(\Psi^m_\delta) \leq 57C_*^2 m + 114C_*^2(\tilde{m}/2) \leq 57C_*^2 m + 114C_*^2 m = 171C_*^2 m$ and we recall that for $m = 2$ holds $M(\Psi^2_\delta) \leq 9C_*m$. □

In [4, Lemma 3.2], a shallow tanh network is constructed which approximates all univariate monomials of degree $1, \ldots, m$ and has width bounded by Cm for some constant $C > 0$. This implies that the size of the NN constructed in [4, Lemma 3.2] is $O(m^2)$. It does not imply that the size of that network can be bounded by a constant times m.

Corollary A.3. *For all $p \in \mathbb{N}$ and $v \in \mathcal{P}_p$, let $v = \sum_{\ell=0}^p v_\ell T_\ell$ denote the Chebyšev expansion of v.*

Then, for all $\delta \in (0,1)$, there exists a tanh *NN $\Phi^{v,p}_\delta$ which satisfies*

$$\|v - \mathrm{R}(\Phi^{v,p}_\delta)\|_{W^{1,\infty}((-1,1))} \leq \delta \sum_{\ell=1}^p |v_\ell|,$$

and, for some constant $C > 0$ which is independent of p, δ and of v,

$$L(\Phi^{v,p}_\delta) = \lceil \log_2(p) \rceil + 1 \ , \ \ M(\Phi^{v,p}_\delta) \leq Cp.$$

The hidden layer weights and biases only depend on p and δ and are independent of v. Those in the output layer are linear combinations of $(v_\ell)_{\ell=0}^p$.

Proof. For $p = 1$, v is an affine function, which can be realized exactly by a NN of depth 1 and size at most 2.

For $p \geq 2$, we use the NN $\Psi_\delta^{\mathrm{Cheb},p}$ from Definition A.1 and Proposition A.2 and define

$$\Phi_\delta^{v,p} := ((A_\delta^{v,p}, b_\delta^{v,p}, \mathrm{Id}_\mathbb{R})) \bullet \Psi_\delta^{\mathrm{Cheb},p},$$

where $A_\delta^{v,p} = (v_1, \dots, v_p) \in \mathbb{R}^{1\times p}$ and $b_\delta^{v,p} = v_0 \in \mathbb{R}$. Its realization satisfies

$$\mathrm{R}(\Phi_\delta^{v,p})(x) = v_0 + \sum_{\ell=1}^{p} v_\ell \mathrm{R}(\Psi_\delta^{\mathrm{Cheb},p})_\ell(x), \qquad x \in \mathbb{R},$$

$$\|v - \mathrm{R}(\Phi_\delta^{v,p})\|_{W^{1,\infty}(I)} \leq \sum_{\ell=1}^{p} v_\ell \|T_\ell - \mathrm{R}(\Psi_\delta^{\mathrm{Cheb},p})_\ell\|_{W^{1,\infty}(I)} \leq \sum_{\ell=1}^{p} |v_\ell| \delta.$$

For the formula for the NN depth, we compute

$$L(\Phi_\delta^{v,p}) = L(((A_\delta^{v,p}, b_\delta^{v,p}, \mathrm{Id}_\mathbb{R}))) - 1 + L(\Psi_\delta^{\mathrm{Cheb},p}) = L(\Psi_\delta^{\mathrm{Cheb},p}) = \lceil \log_2(p) \rceil + 1.$$

To estimate the NN size, we observe from the Definition 7.8 of concatenation that all layers of $\Phi_\delta^{v,p}$ except for the last layer equal those of $\Psi_\delta^{\mathrm{Cheb},p}$. Denoting the weights and biases in the last layer of $\Psi_\delta^{\mathrm{Cheb},p}$ by $A_\delta^{\mathrm{Cheb},p}$ and $b_\delta^{\mathrm{Cheb},p}$, respectively, those in the last layer of $\Phi_\delta^{v,p}$ are $A := A_\delta^{v,p} A_\delta^{\mathrm{Cheb},p}$ and $b := A_\delta^{v,p} b_\delta^{\mathrm{Cheb},p} + b_\delta^{v,p}$, respectively. Denoting by N the dimension of the second to last layer of $\Psi_\delta^{\mathrm{Cheb},p}$, $A \in \mathbb{R}^{1\times N}$, and each element of this matrix is the matrix product of the matrix $A_\delta^{v,p} \in \mathbb{R}^{1\times p}$ with a column of $A_\delta^{\mathrm{Cheb},p}$. Hence $\|A\|_0 \leq \|A_\delta^{\mathrm{Cheb},p}\|_0$. In addition, $b \in \mathbb{R}^1$, thus $\|b\|_0 \leq 1$. Finally, we obtain that

$$M(\Phi_\delta^{v,p}) \leq M(\Psi_\delta^{\mathrm{Cheb},p}) + 1 \leq Cp + 1 \leq Cp,$$

for a constant $C > 0$ independent of p, δ and v. The statement on the NN weights follows directly from the definition of $\Phi_\delta^{v,p}$. □

Remark A.4. *By [20, Theorem 3.13], one can efficiently compute numerically the Chebyšev coefficients $(v_\ell)_{\ell=0}^p$ using the inverse fast Fourier transform. The sum of their absolute values grows at most algebraically with p as we have the upper bound $\sum_{\ell=2}^p |v_\ell| \leq p^4 \|v\|_{L^\infty(I)}$. For more details, see [17, Section 2].*

Appendix B General Activation Functions

The results in the present paper considered specifically ε-uniform DNN emulation rates for the solution set $\{u^\varepsilon : 0 < \varepsilon \leq 1\} \subset H_0^1(I)$ of (2.1)–(2.2) by strict ReLU, spiking, and tanh-activated deep NNs. Lemma 7.9, the key result in the proofs of expression rate bounds for tanh-DNNs, holds more generally, as we state in Lemma B.3 below. Based on this, a result similar to Theorem 7.4 follows also for more general activations, such as the sigmoid introduced in Remark 7.3.

We prepare the proof of the extension of Lemma 7.9 with two auxiliary lemmas. Throughout this section, we will use the calculus of NNs from Section 7.3. Those results hold regardless of the used activation function.

Lemma B.1. *For a nonempty, connected, open subset $U \subset \mathbb{R}$, consider an activation function $\varrho \in C^1(U) \setminus \mathcal{P}_0$.*

For all $M \geq 1$ and $\tau > 0$, there exists a depth 2 ϱ-network $\Phi^{\mathrm{Id},\varrho}_{\tau,M}$ satisfying $\mathrm{R}(\Phi^{\mathrm{Id},\varrho}_{\tau,M})(0) = 0$ and $\|\,\mathrm{Id} - \mathrm{R}(\Phi^{\mathrm{Id},\varrho}_{\tau,M})\|_{W^{1,\infty}((-M,M))} \leq \tau$. Its number of neurons and network size are bounded independently of τ and M. The dimension of the hidden layer is 1.

Proof. Let $t_0 \in U$ be such that $\varrho'(t_0) \neq 0$. There exists $\delta > 0$ such that $[t_0 - \delta, t_0 + \delta] \subset U$ and $\max_{t\in[t_0-\delta,t_0+\delta]} |1 - \varrho'(t)/\varrho'(t_0)| \leq \tau/M$.

Now, let

$$\Phi^{\mathrm{Id},\varrho}_{\tau,M} := \left(\left(\tfrac{\delta}{M}, t_0 \right), \left(\tfrac{M}{\delta \varrho'(t_0)}, -\tfrac{M \varrho(t_0)}{\delta \varrho'(t_0)} \right) \right),$$

which has depth 2, size 4, hidden layer width 1, and realization

$$\mathrm{R}(\Phi^{\mathrm{Id},\varrho}_{\tau,M})(x) = \tfrac{M}{\delta \varrho'(t_0)} \left(\varrho(t_0 + \tfrac{\delta}{M} x) - \varrho(t_0) \right), \qquad x \in [-M, M].$$

It follows that $\mathrm{R}(\Phi^{\mathrm{Id},\varrho}_{\tau,M})(0) = 0$ and that for all $x \in [-M, M]$ there holds

$$\begin{aligned} |1 - \mathrm{R}(\Phi^{\mathrm{Id},\varrho}_{\tau,M})'(x)| &= |1 - \varrho(t_0 + \tfrac{\delta}{M} x)/\varrho'(t_0)| \leq \tau/M \leq \tau, \\ |x - \mathrm{R}(\Phi^{\mathrm{Id},\varrho}_{\tau,M})(x)| &\leq M\tau/M = \tau. \end{aligned}$$

The latter estimate follows from integrating the former one, using exactness in 0 of the identity network. □

Lemma B.2. *For a nonempty, connected, open subset $U \subset \mathbb{R}$, consider the function $f : \mathbb{R} \to \mathbb{R} : x \mapsto x^2$ and an activation function $\varrho \in C^2(U) \setminus \mathcal{P}_1$.*

For all $M \geq 1$ and $\tau > 0$, there exists a depth 2 ϱ-network $\Phi^{f}_{\tau,M}$ satisfying $\mathrm{R}(\Phi^{f}_{\tau,M})(0) = 0$ and $\|f - \mathrm{R}(\Phi^{f}_{\tau,M})\|_{W^{1,\infty}((-M,M))} \leq \tau$. Its number of neurons and network size are bounded independently of τ and M.

Proof. Let $\Phi^{\mathrm{Id},\varrho'}_{\tau/(4M),M}$ be the identity network from Lemma B.1 with activation function ϱ'. Because its hidden layer dimension is 1, all its weight matrices are of dimension 1×1 and its bias vectors are of dimension 1. We identify them with real numbers. We write $\Phi^{\mathrm{Id},\varrho'}_{\tau/(4M),M} = ((a^{(1)}_1, b^{(1)}_1), (a^{(1)}_2, b^{(1)}_2))$, such that $\mathrm{R}(\Phi^{\mathrm{Id},\varrho'}_{\tau/(4M),M})(x) = a^{(1)}_2 \varrho'(a^{(1)}_1 x + b^{(1)}_1) + b^{(1)}_2$ for all $x \in [-M, M]$.

The idea of this proof is to construct a ϱ-NN which approximates the antiderivative of $\mathrm{R}(\Phi^{\mathrm{Id},\varrho'}_{\tau/(4M),M})$, multiplied by 2. To approximate the antiderivative of the constant term, we use an identity network with activation function ϱ. Let $\Phi^{\mathrm{Id},\varrho}_{\tau',M}$ be the identity network from Lemma B.1 with accuracy $\tau' = \tau/(4Mb_2^{(1)})$. We write $\Phi^{\mathrm{Id},\varrho}_{\tau',M} = ((a_1^{(0)}, b_1^{(0)}), (a_2^{(0)}, b_2^{(0)}))$.

Now, define $\Phi^f_{\tau,M} := ((A_1, b_1), (A_2, b_2))$, where

$$A_1 = (a_1^{(1)}, a_1^{(0)})^\top, \quad b_1 = (b_1^{(1)}, b_1^{(0)})^\top, \quad A_2 = (\tfrac{2a_2^{(1)}}{a_1^{(1)}}, 2b_2^{(1)} a_2^{(0)}),$$
$$b_2 = 2b_2^{(1)} b_2^{(0)} - \Big(\tfrac{2a_2^{(1)}}{a_1^{(1)}} \varrho(b_1^{(1)}) + 2b_2^{(1)} \Big(a_2^{(0)} \varrho(b_1^{(0)}) + b_2^{(0)} \Big) \Big),$$

so that for all $x \in [-M, M]$ there holds

$$\begin{aligned} \mathrm{R}(\Phi^f_{\tau,M})(x) = & \tfrac{2a_2^{(1)}}{a_1^{(1)}} \varrho(a_1^{(1)} x + b_1^{(1)}) + 2b_2^{(1)} \mathrm{R}(\Phi^{\mathrm{Id},\varrho}_{\tau',M})(x) \\ & - \Big(\tfrac{2a_2^{(1)}}{a_1^{(1)}} \varrho(b_1^{(1)}) + 2b_2^{(1)} \mathrm{R}(\Phi^{\mathrm{Id},\varrho}_{\tau',M})(0) \Big). \end{aligned}$$

We see that $\mathrm{R}(\Phi^f_{\tau,M})(0) = 0$ and that

$$\begin{aligned} \mathrm{R}(\Phi^f_{\tau,M})'(x) &= 2a_2^{(1)} \varrho(a_1^{(1)} x + b_1^{(1)}) + 2b_2^{(1)} \mathrm{R}(\Phi^{\mathrm{Id},\varrho}_{\tau',M})'(x) \\ &= 2\mathrm{R}(\Phi^{\mathrm{Id},\varrho'}_{\tau/(4M),M})(x) + 2b_2^{(1)} (\mathrm{R}(\Phi^{\mathrm{Id},\varrho}_{\tau',M})'(x) - 1), \\ |2x - \mathrm{R}(\Phi^f_{\tau,M})'(x)| &\le |2x - 2\mathrm{R}(\Phi^{\mathrm{Id},\varrho'}_{\tau/(4M),M})(x)| + 2b_2^{(1)} |\mathrm{R}(\Phi^{\mathrm{Id},\varrho}_{\tau',M})'(x) - 1| \\ &\le 2\tfrac{\tau}{4M} + 2b_2^{(1)} \tau' = \tfrac{\tau}{2M} + \tfrac{\tau}{2M} = \tau/M. \end{aligned}$$

Integrating this error bound gives $|x^2 - \mathrm{R}(\Phi^f_{\tau,M})(x)| \le \tau$ for all $x \in [-M, M]$. The network has depth 2, one hidden layer comprising two neurons, and size 7, independently of τ and M. ☐

Lemma B.3. *Lemma 7.9 also holds if* tanh *is replaced by any activation function $\varrho \in C^2(U) \setminus \mathcal{P}_1$ for a nonempty, connected, open subset $U \subset \mathbb{R}$.*

In addition, it holds that $\mathrm{R}(\Phi^{\mathrm{Id}}_{1,L,\tau,M})(0) = 0$ *and that* $x_1 x_2 = 0$ *implies* $\mathrm{R}(\Phi^{\mathrm{Prod}}_{\tau,M})(x_1, x_2) = 0$.

Proof. Throughout this proof, we assume that $M \ge 1$. If $M < 1$, then we consider the networks constructed for $M = 1$, which also satisfy the statements in the lemma for $M < 1$.

For $L = 2$, the network $\Phi^{\mathrm{Id}}_{1,2,\tau,M} := \Phi^{\mathrm{Id},\varrho}_{\tau,M}$ from Lemma B.1 satisfies all the desired properties.

The definition and the analysis of identity networks of depth $L > 2$ are identical to those in the proof of Lemma 7.9. From the definition $\Phi^{\mathrm{Id}}_{1,L,\tau,M} := \Phi^{\mathrm{Id}}_{1,2,\tau/3,M+\tau/3} \bullet \Phi^{\mathrm{Id}}_{1,L-1,\tau/3,M}$ we inductively obtain that $\mathrm{R}(\Phi^{\mathrm{Id}}_{1,L,\tau,M})(0) = 0$.

To construct product networks, for all $\tau > 0$, $M \ge 1$, let $\Phi^f_{\tau,M}$ be the ϱ-NN from Lemma B.2 approximating $f : [-M, M] \to \mathbb{R} : x \mapsto x^2$.

Now, let $A_1 \in \mathbb{R}^{3\times 2}$ be such that for all $x = (x_1, x_2) \in \mathbb{R}^2$ there holds $A_1 x = (\frac{1}{2}x_1 + \frac{1}{2}x_2, \frac{1}{2}x_1, \frac{1}{2}x_2)$, let $b_1 := 0 \in \mathbb{R}^3$, $A_2 := (2, -2, -2) \in \mathbb{R}^{1\times 3}$ and $b_2 := 0 \in \mathbb{R}$. Then, the NN $\Phi^{\mathrm{Prod}}_{\tau,M} := ((A_2, b_2)) \bullet \mathrm{P}(\Phi^f_{\tau/6,M}, \Phi^f_{\tau/6,M}, \Phi^f_{\tau/6,M}) \bullet ((A_1, b_1))$ has realization $\mathrm{R}(\Phi^{\mathrm{Prod}}_{\tau,M})(x_1, x_2) = 2\mathrm{R}(\Phi^f_{\tau/6,M})(\frac{1}{2}x_1 + \frac{1}{2}x_2) - 2\mathrm{R}(\Phi^f_{\tau/6,M})(\frac{1}{2}x_1) - 2\mathrm{R}(\Phi^f_{\tau/6,M})(\frac{1}{2}x_2)$ for all $(x_1, x_2) \in \mathbb{R}^2$. With $\mathrm{R}(\Phi^f_{\tau/6,M})(0) = 0$ we obtain that $\mathrm{R}(\Phi^{\mathrm{Prod}}_{\tau,M})(x_1, 0) = 0 = \mathrm{R}(\Phi^{\mathrm{Prod}}_{\tau,M})(0, x_2)$ for all $x_1, x_2 \in [-M, M]$. To estimate the $L^\infty((-M, M)^2)$ error, for all $(x_1, x_2) \in [-M, M]^2$ there holds

$$\begin{aligned}
|x_1 x_2 - \mathrm{R}(\Phi^{\mathrm{Prod}}_{\tau,M})(x_1, x_2)| \leq\, & 2|(\tfrac{1}{2}x_1 + \tfrac{1}{2}x_2)^2 - \mathrm{R}(\Phi^f_{\tau/6,M})(\tfrac{1}{2}x_1 + \tfrac{1}{2}x_2)| \\
& + 2|(\tfrac{1}{2}x_1)^2 - \mathrm{R}(\Phi^f_{\tau/6,M})(\tfrac{1}{2}x_1)| \\
& + 2|(\tfrac{1}{2}x_2)^2 - \mathrm{R}(\Phi^f_{\tau/6,M})(\tfrac{1}{2}x_2)| \\
\leq\, & 6\tau/6 = \tau.
\end{aligned}$$

For the error in the derivative with respect to x_1, we find that for all $(x_1, x_2) \in [-M, M]^2$ there holds

$$\begin{aligned}
|x_2 - \tfrac{\partial}{\partial x_1}\mathrm{R}(\Phi^{\mathrm{Prod}}_{\tau,M})(x_1, x_2)| \leq\, & \left|(x_1 + x_2) - 2\mathrm{R}(\Phi^f_{\tau/6,M})'(\tfrac{1}{2}x_1 + \tfrac{1}{2}x_2) \cdot \tfrac{1}{2}\right| \\
& + \left|x_1 - 2\mathrm{R}(\Phi^f_{\tau/6,M})'(\tfrac{1}{2}x_1) \cdot \tfrac{1}{2}\right| + 0 \\
\leq\, & 2\tau/6 \leq \tau,
\end{aligned}$$

where the factors $\frac{1}{2}$ are due to the chain rule. The analogous bounds for differentiation with respect to x_2 also hold.

Because the number of neurons of $\Phi^f_{\tau/6,M}$ is bounded independently of M and τ, so is that of $\Phi^{\mathrm{Prod}}_{\tau,M}$. Denoting by N_1 the number of neurons in the hidden layer of $\Phi^{\mathrm{Prod}}_{\tau,M}$, because the input dimension is $N_0 = 2$ and the output dimension is $N_2 = 1$, the total number of nonzero weights and biases is at most $\sum_{\ell=1}^{2} N_\ell(N_{\ell-1} + 1) = 4N_1 + 1$, which is independent of M and τ. □

Remark B.4. *As a result of Lemma B.3, all constructions and results in Section 7.4, Appendix A and Section 7.5 also hold for NNs whose activation function satisfies the conditions of Lemma B.3, which includes ReLU^2 and the sigmoid defined in Remark 7.3.*

With the approximation of the exponential boundary layer functions by sigmoid NNs in Remark 7.3, the main result of Section 7 also holds for sigmoid NNs.

Theorem B.5. *Theorem 7.4 also holds for sigmoid NNs.*

References

[1] Ainsworth, M. and J. Dong. 2022. Galerkin neural network approximation of singularly-perturbed elliptic systems. *Comput. Methods Appl. Mech. Engrg.* 402: Paper No. 115169, 29. https://doi.org/10.1016/j.cma.2022.115169 .

[2] Chang, T.Y., G.M. Gie, Y. Hong, and C.Y. Jung. 2023. Singular layer physics informed neural network method for plane parallel flows. ArXiv:2311.15304.

[3] Davis, P.J. 1975. *Interpolation and approximation.* Dover Publications, Inc., New York. Republication, with minor corrections, of the 1963 original, with a new preface and bibliography.

[4] De Ryck, T., S. Lanthaler, and S. Mishra. 2021. On the approximation of functions by tanh neural networks. *Neural Networks* 143: 732–750. https://doi.org/10.1016/j.neunet.2021.08.015 .

[5] Elbrächter, D., P. Grohs, A. Jentzen, and C. Schwab. 2022. DNN expression rate analysis of high-dimensional PDEs: Application to option pricing. *Constructive Approximation 55*(1): 3–71. https://doi.org/10.1007/s00365-021-09541-6 .

[6] Gautschi, W. 2004. *Orthogonal polynomials : computation and approximation.* Numerical mathematics and scientific computation. Oxford: Oxford University Press.

[7] Gie, G.M., M. Hamouda, C.Y. Jung, and R.M. Temam. 2018. *Singular perturbations and boundary layers*, Volume 200 of *Applied Mathematical Sciences.* Springer, Cham.

[8] Herrmann, L., J.A.A. Opschoor, and C. Schwab. 2022. Constructive deep ReLU neural network approximation. *Journal of Scientific Computing 90*(2): 75. https://doi.org/10.1007/s10915-021-01718-2 .

[9] Maass, W. 1997a. Fast sigmoidal networks via spiking neurons. *Neural Computation 9*(2): 279–304. https://doi.org/10.1162/neco.1997.9.2.279 .

[10] Maass, W. 1997b. Networks of spiking neurons: The third generation of neural network models. *Neural Networks 10*(9): 1659–1671. https://doi.org/10.1016/S0893-6080(97)00011-7 .

[11] Marcati, C., J.A.A. Opschoor, P.C. Petersen, and C. Schwab. 2023. Exponential ReLU neural network approximation rates for point and edge singularities. *Journ. Found. Comp. Math. 23*(3): 1043–1127. https://doi.org/https://doi.org/10.1007/s10208-022-09565-9 .

[12] Marcati, C. and C. Schwab. 2023. Exponential convergence of deep operator networks for elliptic partial differential equations. *SIAM J. Numer. Anal. 61*(3): 1513–1545. https://doi.org/10.1137/21M1465718 .

[13] Melenk, J.M. 1997. On the robust exponential convergence of *hp* finite element method for problems with boundary layers. *IMA J. Numer. Anal. 17*(4): 577–601. https://doi.org/10.1093/imanum/17.4.577 .

[14] Melenk, J.M. and C. Xenophontos. 2016. Robust exponential convergence of *hp*-FEM in balanced norms for singularly perturbed reaction-diffusion equations. *Calcolo 53*(1): 105–132. https://doi.org/10.1007/s10092-015-0139-y .

[15] Opschoor, J.A.A. 2023. *Constructive deep neural network approximations of weighted analytic solutions to partial differential equations in polygons*. Ph. D. thesis, ETH Zürich. Diss. ETH No. 29278.

[16] Opschoor, J.A.A., P.C. Petersen, and C. Schwab. 2020. Deep ReLU networks and high-order finite element methods. *Analysis and Applications 18*(05): 715–770. https://doi.org/10.1142/S0219530519410136 .

[17] Opschoor, J.A.A. and C. Schwab 2023. Deep ReLU networks and high-order finite element methods II: Chebyshev emulation. Technical Report 2023-38, Seminar for Applied Mathematics, ETH Zürich, Switzerland.

[18] Petersen, P. and F. Voigtlaender. 2018. Optimal approximation of piecewise smooth functions using deep ReLU neural networks. *Neural Netw.* 108: 296 – 330. https://doi.org/10.1016/j.neunet.2018.08.019 .

[19] Rauhut, H. and C. Schwab. 2017. Compressive sensing Petrov-Galerkin approximation of high-dimensional parametric operator equations. *Math. Comp. 86*(304): 661–700. https://doi.org/10.1090/mcom/3113 .

[20] Rivlin, T.J. 1974. *The Chebyshev polynomials*. Pure and Applied Mathematics. Wiley-Interscience [John Wiley & Sons], New York-London-Sydney.

[21] Schwab, C. 1998. *p- and hp-finite element methods*. Numerical Mathematics and Scientific Computation. The Clarendon Press, Oxford University Press, New York.

[22] Schwab, C. and M. Suri. 1996. The *p* and *hp* versions of the finite element method for problems with boundary layers. *Math. Comp. 65*(216): 1403–1429. https://doi.org/10.1090/S0025-5718-96-00781-8 .

[23] Stanojevic, A., S. Woźniak, G. Bellec, G. Cherubini, A. Pantazi, and W. Gerstner. 2022. An exact mapping from ReLU networks to spiking neural networks. ArXiv:2212.12522.

[24] Tang, S., B. Li, and H. Yu 2019. ChebNet: Efficient and stable constructions of deep neural networks with rectified power units using Chebyshev approximations. Technical report. ArXiv: 1911.05467.

[25] Trefethen, L.N. 2019. *Approximation theory and approximation practice* (Extended ed.). Philadelphia: Society for Industrial and Applied Mathematics.