Adaptive numerical approximation of two-point boundary value problems: a neural network-based approach

Ferenc Izsák

Department of Applied Analysis and Computational Mathematics & AI ELTE Research Group, Eötvös Loránd University, Pázmány P. stny. 1.C, 1117 Budapest, Hungary

ferenc.izsak@ttk.elte.hu

Abstract

An adaptive finite element method is developed here for the numerical solution of one-dimensional boundary value problems. The method is based a on a neural network representation of continuous, piecewise linear functions. The proposed optimization procedure is demonstrated in a test problem.

1 Introduction

Neural networks have proven their usefulness in a wide range of scientific computing. For classical problems in the numerical analysis, its application is less usual. In this contribution, we propose a way to contribute to this research direction.

For the formal introduction of neural networks, we refer to [3] and [4]. For our purpose, it is sufficient to use that one can assign to any neural network NN a real vector function $\mathcal{NN} : \mathbb{R}^{d_0} \to \mathbb{R}^{d_N}$, which maps to the input an output value. The function itself is given in concrete terms using some internal parameters. The main power of this approach lies in the efficient optimization procedure, which drives the optimal choice of these model parameters. For this, we mostly use given input-output pairs and choose parameters, which lead to smallest deviation between computed and known outputs. In the absence of given pairs, we can also try to define a meaningful loss function, which should be minimized directly to get optimal parameters.

The automatic differentiation in the related program packages makes possible to deal also with millions of parameters. At the same time, we should not misuse this capability and keep the number of parameters at a moderate level to avoid overfitting and enhance the computational efficiency.

Accordingly, we use here the idea to solve a problem in numerical analysis by converting it into a multidimensional minimization.

2 Problem statement and methods

We investigate two-point boundary value problems for second-order ordinary differential equations of the following form:

$$\begin{cases} -u"(x) + c(x)u(x) = f(x) \quad x \in (a,b) \\ u(a) = u(b) = 0, \end{cases} \tag{1}$$

where $c \in L_\infty(\Omega)$ and $f \in L_2(\Omega)$ are given.

We are looking for a numerical approximation $u_h : [a, b] \to \mathbb{R}$ of u as a piecewise linear, continuous function. In concrete terms, we assume that they are linear on $[t_j, t_{j+1}]$ with the slope s_j, where $t_0 = a, t_{N+1} = b$. Such a function can be characterized with $[t_1, t_2, \dots, t_N]$ and $[s_1, s_2, \dots, s_N]$.

According to [4], such a function x_h can be identified with the neural network

$$x_h(t) := \mathcal{NN}(t) = \mathbf{a}_2 \cdot \mathrm{ReLu}(\mathbf{a}_1 t + \mathbf{b}_1) + b_2, \tag{2}$$

with the input t, and parameters on the first layer

$$\mathbf{a}_1 = (1, 1, \dots, 1) \in \mathbb{R}^{N+1} \quad \text{and} \quad \mathbf{b}_1 = (0, -t_1, \dots, -t_N) \in \mathbb{R}^{N+1}$$

and on the second layer

$$\mathbf{a}_2 = (s_1, s_2 - s_1, \dots, s_N - s_{N-1}, s_{N+1} - s_N) \in \mathbb{R}^{N+1} \quad \text{and} \quad b_2 = x_0 \in \mathbb{R},$$

respectively. Note that here s_{N+1} is a known parameter. For the details, see [4].

To find optimal parameters in the above setting, we cannot use known input-output pairs. Instead, the following statement delivers an appropriate loss function.

Theorem 1 *$u \in H_0^1(a, b)$ is the unique solution of (1) if and only if $u \in H_0^1(\Omega)$ is the unique minimum of $J : H_0^1(a, b) \to \mathbb{R}$:*

$$J(u) = \frac{1}{2} \int_a^b (u')^2 + cu^2 - \int_a^b f \cdot u. \tag{3}$$

In this way, our approach is to find parameters $\mathbf{t} = (t_1, t_2, \dots t_{N-1}) \in \mathbb{R}^{N-1}$ and $\mathbf{s} = (s_1, s_2, \dots s_{N-1}) \in \mathbb{R}^{N-1}$ such that $J(u_{\mathbf{s},\mathbf{t}}) := J_{\mathbf{s},\mathbf{t}}$ is minimal, where $u_{\mathbf{s},\mathbf{t}}$ denotes the piecewise linear function described at the beginning of the section.

To optimize the performance of our algorithm, we use the following statement.

Lemma 2 *For any fixed parameter set $\mathbf{t}$ above, the minimum of $J_{\mathbf{s},\mathbf{t}}$ is attained, if the corresponding function $u_{\mathbf{s},\mathbf{t}}$ is the finite element solution of (1) using a piecewise first order basis with internal vertices $t_1 \le t_2 \le \dots \le t_N$.*

For the proof of the above two statements, we refer to [1]. Using these results, we can reduce the number of parameters in the minimization problem and consider henceforth the following problem:

Find the parameter $\mathbf{t}$ such that $J_{\mathbf{s}(\mathbf{t}),\mathbf{t}}$ is minimal, where $\mathbf{s}(\mathbf{t})$ corresponds to the finite element solution in Lemma 2.

Observe that this is, indeed, an adaptive finite element algorithm, where the basis points $t_1, t_2, \dots, t_N$ are to find in an optimal way.

It is important to ensure that we have an optimal parameter set also in the discrete case, which is stated in the following:

Lemma 3 *For any fixed N, we have $\mathbf{t}$ and $\mathbf{s}$ above such that $J_{\mathbf{s},\mathbf{t}}$ is minimal.*

Proof:

According to Lemma 2, it is sufficient to ensure the existence of $\mathbf{t} \in \mathbb{R}^N$, for which $J_{\mathbf{s}(\mathbf{t}),\mathbf{t}}$ is minimal. Since the mappings $\mathbf{t} \to \mathbf{s}(\mathbf{t})$ and $(\mathbf{t}, \mathbf{s}) \to J_{\mathbf{t},\mathbf{s}}$ are continuous, the same applies for $\mathbf{t} \to J_{\mathbf{s}(\mathbf{t}),\mathbf{s}}$. On the other hand, indeed the definition domain of this mapping is

$$\mathcal{T}_N = \{\mathbf{t} = (t_1, t_2, \dots, t_N) : a \le t_1 \le t_2 \le \cdots \le t_N \le b\} \subset \mathbb{R}^N,$$

which is compact, and therefore, we really have a local minimum at some $\mathbf{t} \in \mathcal{T}_N$. □

Observe, if the minimum is attained at the boundary of $\mathcal{T}_N$, then $t_j = t_{j+1}$ for some index $j \in \{0, 1, \dots, N\}$. This results in exactly the same piecewise linear approximation as $\dots, t_{j-1}, t_j, \frac{t_j + t_{j+2}}{2}, t_{j+2}, \dots$ with the slopes $\dots, s_{j-1}, s_{j+1}, s_{j+1}, s_{j+2}, \dots$. In this way, the minimum should also attained in the interior of $\mathcal{T}_N$.

3 Implementation issues and numerical results

Indeed, to find an optimal $\mathbf{t} \in \mathcal{T}_N$, we had to perform a conditional minimization. It turns out that unconditional minimization can harm the order of the components in $\mathbf{t}$.

To reduce the computational complexity, we introduce the following penalty term to avoid conditional minimization:

$$P_{\mathbf{t}} = K \cdot (|t_1 - 0| + |t_2 - t_1| + \cdots + |1 - t_N| - 1),$$

where $K = 1000$ in the experiments. Clearly, if $\mathbf{t} \in \mathcal{T}_N$, this term should be zero. Altogether, we computed the minimum of

$$\mathbf{t} \to J_{\mathbf{s}(\mathbf{t}),\mathbf{t}} + P_{\mathbf{t}}$$

starting from a uniform division of the interval (a, b). To approximate integrals in the loss term and in the finite element method, we applied a three-point Gauß integral. One can increase the accuracy of integration using the built-in Matlab subroutines but this does not increase further the accuracy of the final result. To compare our method with a similar one in [2], we use the same test problem

$$\begin{cases} \ddot{u}(x) = \frac{200}{9} \cdot \exp\{-100(x - \frac{1}{3})^2\} \cdot (1800 \cdot x^3 - 1200 \cdot x^2 + 173 \cdot x + 6) & x \in (0, 1) \\ u(0) = u(1) = 0, \end{cases} \tag{4}$$

where the analytic solution is given by $u(x) = x \cdot (\exp\{-100(x - \frac{1}{3})^2\} - \exp\{-400/9\})$.

The finite element solution, i.e. optimal piecewise linear approximation for (4) with the starting value $\mathbf{t}$ and with the optimal $\mathbf{t}$ are shown in Figure 1 and 2, respectively.

Also, we have tested the computational error of the adaptive finite element method given by the above optimization process in the $H_0^1(a, b)$-norm. The results are shown in Table 3.

N	4	9	19	39	79
$\mathrm{err}_{\mathrm{ad}}$	4.45	0.637	0.305	0.155	0.0774
$\mathrm{err}_{\mathrm{un}}$	7.52	0.282	0.188	0.109	0.0608

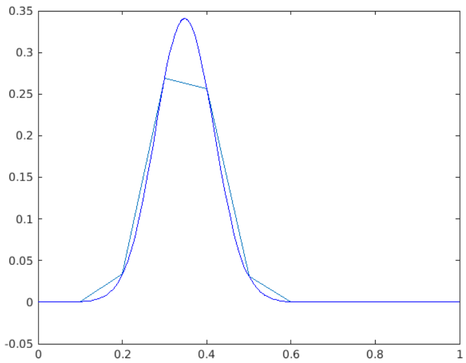

Figure 1: Finite element solution of (4) with $\mathbf{t} = (0.1, 0.2, \ldots, 0.9)$ (dashed) together with the analytic solution of (4).

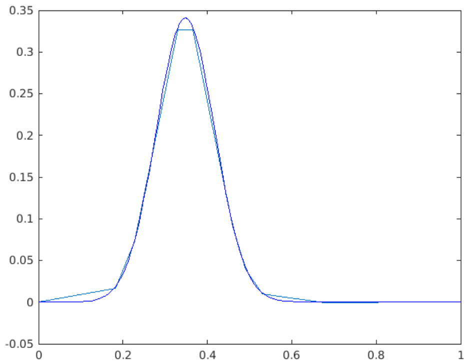

Figure 2: Finite element solution of (4) with the optimal $\mathbf{t}$ together with the analytic solution of (4).

One can realize that the advance of adaptive methods is significant only in the case of relatively coarse meshes. On the other hand, the test problem in (4) has smooth solution. Therefore, on a sufficiently fine mesh, its solution can be approximated well also without adaptive refinement.

Acknowledgments

The research was supported by the Ministry of Innovation and Technology NRDI Office within the framework of the Artificial Intelligence National Laboratory Program.

References

[1] **Axelsson, O., Barker, V.A.,** *Finite Element Solution of Boundary Value Problems*, Academic Press, San Diego, 1984.

[2] **He, J., Li, L., Xu, J. and Zheng, C.,** ReLu deep neural networks and linear finite elements, *J. Comput. Math.*, **38**(3) (2020), 502–527.

[3] **Kröse, B., van der Smagt, P.,** *An introduction to Neural Networks*, The University of Amsterdam, Amsterdam, 1996.

[4] **Opschoor, J., Petersen, P. and C. Schwab, C.,** Deep ReLu networks and high-order finite element methods. *Anal. Appl.*, **18**(5) (2020), 715–770.