ReLU artificial neural networks for the grid adaptation in finite element method

Xuemei Fu, Luoping Chen and Fanyun Wu

1School of Mathematics, Southwest Jiaotong University, Chengdu 611756, China

Abstract. In this paper, we study the rectified linear unit (ReLU) artificial neural network (ANN) for grid adaptation in finite element method, which is used for solving differential equations (DEs) with initial/boundary condition. Compared with the classical adaptive finite element method (AFEM), ReLU ANN based on finite element method can keep the number of grid-points constant but change their relative location. Our numerical experiments show that approximate solutions obtained from the classical finite element method by ReLU ANN are more accurate than those obtained by AFEM.

1. Introduction

Differential equations are important tools to solve practical problems of physics or engineering. Since many differential equations cannot be solved analytically, studies of numerical solutions of differential equations play a significant role in mathematical physics such as elastic mechanics problems [1, 2]. There are tremendous studies of numerical methods for solving differential equations in the literature, such as the finite element method [3], the finite difference method [4] and the finite volume method [5]. When solving some problems [6], one often encounters the difficulty that the overall approximate accuracy of numerical solutions is deteriorated by local singularities arising that may come from discontinuity coefficient, interior or boundary layers and sharp shock-lick fronts and so on. One of the remedies is to refine the discretization near the critical regions, i.e., put more grid points where the solution is less regular. A large number of investigations of adaptive methods based on posterior error estimation are proposed to refine the meshes for different kinds of differential equations. The adaptive process of large part of studies is by adding more grid points in the critical region. The error estimation for the adaptation of various models has been well studied such as [7]. The objective of this paper is to present a new grid adaptation method in numerical approximation by ANN. This method can be used in solving differential equations with less regular solutions, which arise in many real applications such as model problems in physics and engineering.

Artificial neural network has yielded enormous achievements in many applied areas in the recent ten years, such as pattern recognition, nature language processing and reinforcement learning [8]. The ANN can be effectively trained with different activation functions, i.e., sigmoidal activation function and leaky ReLU activation function. [9] shows that any continuous piecewise linear function in $\mathbb{R}^d$ can be represented by ANN model with ReLU activation function which has at most $\left\lceil \log_2 (d+1) \right\rceil$ hidden layers.

In this paper, we use the popular artificial neural network algorithm with a rectified linear unit as an activation function for the grid adaptation in finite element method. As we will see, the adaptation
of finite element method by ReLU ANN not only keeps the number of grid-points constant, but also achieves good accuracy. We will use several DEs in one dimensional to illustrate that the adaptation of ReLU ANN has better approximation than AFEM. We also compare the ReLU ANN method with moving-point technique in [10] that also keeps grid-points stable in this paper to show the efficiency. $H^1$ error estimation for two examples are shown in the numerical experiments.

The rest of the paper is organized as follows. In Section 2, we provide notations and algorithm of grid adaptation by ReLU ANN. In Section 3, we simply present the classical adaptive finite element method. In Section 4, some numerical experiments are given to implement ReLU ANN as well as AFEM for DEs. At the end, a general conclusion is provided.

2. The grid adaptation based on ReLU ANN

In this section, we mainly introduce the definition and notations of the ReLU ANN and the algorithm of grid adaptation by ReLU ANN.

2.1. Preliminary

An ANN contains three parts: input layers, hidden layers and output layers. A shallow neural network with only one hidden layer is shown in Figure 1 (a). A multi-layer neural network with more than one hidden layer is shown in Figure 1 (b). As shown in [11], a finite element function in $R^1$ with $n$ degrees of freedom can be represented by a ReLU ANN model with at most $O(n)$ number of neurons.

In this paper, we mainly use an ANN with one hidden layer for grid adaptation. We take $x \in R^{nx1}$ as the input vector, $w \in R^{wmxn}$ as the weight matrix and $b \in R^{mx1}$ as the bias term ($m,n \geq 1$). The hidden layer is computed by the following formula:

$$z(x) = \sigma(wx + b),$$

(2.1)

where $\sigma$ is the activation function. In this paper, we mainly use ReLU as the activation function which is given by

$$\text{ReLU}(x) = \max(0, x).$$

(2.2)

![Figure 1. A shallow ANN (a) and ANN with $k$ hidden layers (b).](image)

The output layer is represented by the following equation:

$$output = \alpha z + b,$$

(2.3)

where $\alpha \in R^{1xn}$ is a weight vector and $b \in R$ is a bias term.

The set generated by ReLU ANN with one hidden layer can be represented as
\[ AN^{1} = \{ f : f = \alpha \text{ReLU}(wx + b_1) + b_2 \}, \quad \text{(2.4)} \]

where \( w, \alpha, b_1, b_2 \) are adjustable parameters which determine the effect of approximation.

### 2.2. ReLU ANN for the grid adaptation

For simplicity, we consider the following one dimensional model problem:

\[
-u^*(x) = f(x), \quad x \in (0, 1) \\
u(0) = u(1) = 0
\]

\[
 f(x) \in L^2(0, 1) = \{ f : \|f\|_{L^2(0, 1)} < \infty \}
\]

is a given function. To solve equation (2.5), it is sufficient to minimize the following energy functional:

\[
 J(v) = \int_{0}^{1} \left( \frac{1}{2} |v'|^2 - fv \right) dx .
\]

where \( v \in C^1(0, 1) \), a continuous differential function space defined in \((0, 1)\).

Creating a mesh with \( N + 1 \) elements on the interval \((0, 1)\)

\[
 T_N : 0 = t_0 < t_1 < ... < t_{N+1} = 1,
\]

the space of ReLU ANN with one hidden layer can be defined as

\[
 U = \left\{ v_h(x; t, \theta) \mid v_h(x; t, \theta) = \sum_{i=0}^{N} (\theta_{i+1} - \theta_i) \text{ReLU}(x - t_i) \right\},
\]

where \( \theta_i \) is an unknown coefficient. Boundary condition \( u(0) = u(1) = 0 \) implies \( \theta_0 = 0 \) and \( v_h(1; t, \theta) = 0 \) which results in

\[
 v_h(1; t, \theta) = \sum_{i=0}^{N} (\theta_{i+1} - \theta_i) \text{ReLU}(1 - t_i) = \sum_{i=0}^{N} \theta_{i+1} (t_{i+1} - t_i) = 0 .
\]

We replace \( v \) in equation (2.6) by \( v_h \) and minimize the energy functional:

\[
 J(v_h) = \int_{0}^{1} \frac{1}{2} |v_h'|^2 - f v_h dx,
\]

then we have

\[
 J(v_h) = \int_{0}^{1} \frac{1}{2} \left| \sum_{i=0}^{N} (\theta_{i+1} - \theta_i) \text{ReLU}(x - t_i) \right|^2 - f(x) \sum_{i=0}^{N} (\theta_{i+1} - \theta_i) \text{ReLU}(x - t_i) dx .
\]

In our algorithm, we employ the gradient descent (GD) method \([12, 13]\) to update \( \{t_i\}_{i=1}^{N} \). We do the iteration as below

\[
 t_{k+1} = t_k - \eta \nabla_v J(v_h(x; t_k, \theta^k)) ,
\]

(2.9)
where $\eta$ is the step length. Once $t$ is fixed, $\theta_i$ is derived by minimizing equation (2.8). Moreover, $\text{ReLU}(x-t_i)$ can be seen as the finite element basis function and $\theta_i$ is the coefficient which can be computed by the finite element method.

To compute the equation (2.8), we need to finish two parts. The first part

$$
\int_0^1 \left| \frac{1}{2} \left( \sum_{i=0}^{N} (\theta_{i+1} - \theta_i) \text{ReLU}(x-t_i) \right) \right|^2 dx = \frac{1}{2} \sum_{i=1}^{N+1} \theta_i^2 (t_i - t_{i-1})
$$

(2.10)

and the second part

$$
\int_0^1 f(x) \sum_{i=0}^{N} (\theta_{i+1} - \theta_i) \text{ReLU}(x-t_i) dx = \sum_{i=0}^{N} \int_{t_i}^{t_{i+1}} f(x)(\theta_{i+1} - \theta_i)(x-t_i) dx.
$$

(2.11)

During the process of GD, the grids $\{t_i\}_{i=1}^N$ are successively updated which results in grid adaptation.

2.3. Brief description of algorithm process

The first step gives the initial grid $t$, max iteration $M$ and step length $\eta$. Then we solve initial $\theta$ by the traditional finite element method on the given grid $t$. The second step is to iteratively update $t$ through equation (2.9) and compute $\theta$ by minimizing equation (2.8). So we obtain parameters $t$, $\theta$ and the solution of the problem (2.5). The brief algorithm process is shown in the Figure 2.

![Figure 2. The process of the algorithm.](image)

3. Adaptive finite element method

The adaptive finite element method (AFEM) is a general numerical technique for solving DEs with singularity [14-16].

Assuming that $I$ is a bounded domain and $T_h$ is a simplicial partition, the corresponding set of nodal points is denoted by $N_h$. Let $V_{h,0}$ be a finite element space which contains all continuous piecewise linear functions with respect to $T_h$ and satisfies the boundary condition of the problem (2.5),

$$
V_{h,0} = \left\{ v : \|v\|_{L^2(I)} < \infty, \|v'\|_{L^2(I)} < \infty, v(0) = v(1) = 0 \right\}.
$$

It is easy to see the finite element solution $u_h \in V_{h,0}$. The variational formulation of equation (2.5) is given by

$$\int_0^1 u_h' v' dx - u_h' (1)v(1) + u_h' (0)v(0) = \int_0^1 f v dx,$$

where $v \in V_{h,0}$ is a test function.

Given $x_i \in N_h$, there exists a nodal linear basis function $\phi_i (x) \in V_{h,0}$. Obviously, any $v \in V_{h,0}$ can be represented as

$$v(x) = \sum_{i=1}^{N-1} \xi_i \phi_i (x),$$

where $\{ \xi_i \}_{i=1}^{N-1}$ are unknown coefficients. In order to solve the problem adaptively, we use the following error estimation:

$$\left\| (u - u_h') \right\| \leq C \sum_{i=1}^{N-1} h_i \| f \|_{L^2(t_i)},$$

where $h_i = x_{i+1} - x_i$ and $C$ is a constant that independent of $h$.

4. Numerical results
In this section, we will implement the ReLU ANN proposed in 2.2 by several examples. Roughly speaking, the ReLU ANN is also a moving finite element method (MFEM) because the original number of grid-points keep the same. For comparison, we test AFEM as well as MFEM proposed in [17, 18].

**Example 1.** We choose the analytical solution $u(x) = x^2 - x$ which results in $f = -2$. Figure 3 shows the adaptive grid by ReLU ANN method, AFEM and MFEM. Table 1 shows the $H^1$ semi-norm errors of the above three methods. The $H^1$ semi-norm is defined as

$$\| u - u_h \| = \left( \int_t \left( u' - u_h' \right)^2 dx \right)^{1/2}.$$

![Figure 3. ReLU ANN (a), AFEM (b) and MFEM (c).](image)

**Example 2.** We choose the analytical solution $u(x) = x e^{-(x^2)} - e^{-\frac{x^2}{3}}$ and $k = 0.008$. Figure 4 shows the adaptive grid by ReLU ANN method, AFEM and MFEM. The $H^1$ semi-norm of the three methods can be obtained from Table 2.
Example 3. The model problem with discontinuity coefficient is given by

\[-(k(x)u'(x))' = 1, \quad x \in (0,1),\]

with the initial boundary condition \(u(0) = u(1) = 0\), where

\[k(x) = \begin{cases} \frac{x}{3}, & 0 \leq x \leq \frac{1}{3} \\ 2x, & \frac{1}{3} < x \leq 1 \end{cases},\]

whose exact solution is \(u(x) = \begin{cases} 3x, & 0 \leq x \leq \frac{1}{3} \\ \frac{3}{2}(1-x), & \frac{1}{3} < x \leq 1 \end{cases}.

Table 1. The \(H^1\) semi-norm error.

| \(N\) | \(|u_{\text{ANN}} - u|_1\) | \(|u_{\text{AFEM}} - u|_1\) | \(|u_{\text{MFEM}} - u|_1\) |
|---|---|---|---|
| 8  | 0.11706          | 0.6721          | 1.15543          |
| 16 | 0.04703          | 0.52857         | 0.68925          |
| 24 | 0.02890          | 0.44148         | 0.41290          |

Figure 4. ReLU ANN (a), AFEM (b) and MFEM (c).

Figure 5. ReLU ANN (a), AFEM (b) and MFEM (c).
Table 2. The $H^1$ semi-norm error.

| $N$ | $|u_{\text{ANN}} - u|_1$ | $|u_{\text{AFEM}} - u|_1$ | $|u_{\text{MFEM}} - u|_1$ |
|-----|-------------------------|-------------------------|-------------------------|
| 8   | 0.28991                 | 1.08983                 | 1.82505                 |
| 16  | 0.13927                 | 0.51184                 | 1.58071                 |
| 24  | 0.09185                 | 0.33681                 | 0.19566                 |

The Tables 1-2 with $\eta = 0.0001$, $v = 1$, $C = 1$ demonstrate that the $H^1$ semi-norm of the ANN solution are smaller than $u_{\text{AFEM}}$ and $u_{\text{MFEM}}$. The numerical results imply the algorithm of grid adaptation by ReLU ANN is better than the AFEM and the MFEM. The Figures 3-5 shows the approximate ability of the ReLU ANN is better than the AFEM and MFEM under the same number of grid points.

5. Conclusions
In this paper, we present the popular artificial neural network algorithm with a rectified linear unit as an activation function for the grid adaptation in finite element method for solving DEs. As indicated by our experiments, the grid adaptation method by ReLU ANN maintains the number of grid points and still achieves good accuracy. In our future work, we will consider using ReLU ANN to solve high-dimensional PDEs.

Acknowledgments
This paper is supported by Natural Science Foundation of China (No. 11501473).

References
[1] Woodward W H, Utyuzhnikov S and Massin P 2018 Developments of the Method of Difference Potentials for Linear Elastic Fracture Mechanics Problems Int. J. Numer. Meth. Eng.
[2] Parker S G 2006 A component-based architecture for parallel multi-physics PDE simulation Future. Gener. Comput. Syst. 22(1-2) 204-216
[3] Chen S H 2019 Adaptive Techniques in the Finite Element Method Computational Geomechanics and Hydraulic Structures
[4] Chen C J, Bernatz R, Carlson K D and Lin W 2020 The Finite Difference Method Finite Analytic Method in Flows and Heat Transfer
[5] Chen S, Li X and Rui H 2019 The finite volume method based on the Crouzeix–Raviart element for a fracture model Numer. Meth. Part. D. E. 35(2)
[6] Belyaev, Vasily 2020 Solving a Poisson equation with singularities by the least-squares collocate-on method Numerical Analysis and Applications 23(3) 207-218
[7] Irisarri D and Hauke G 2020 A posteriori error estimation and adaptivity based on VMS for the incompressible Navier-Stokes equations Comput. Methods Appl. Mech. Engrg. 373
[8] LeCun Y, Bengio Y and Hinton G 2015 Deep learning nature 521(7553) 436-444
[9] Arora R, Basu A, Mianjy P and Mukherjee A 2016 Understanding deep neural networks with rectified linear units
[10] Hulsemann T F 1998 A New Moving Mesh Algorithm for the Finite Element Solution of Variational Problems Siam. J. Numer. Anal. 35(4) 1416-1438
[11] He J, Li L, Xu J and Zheng C 2018 ReLU deep neural networks and linear finite elements J. Comput. Math. 38(2020) 502-527
[12] Jiang H 2019 Why learning of large-scale neural networks behaves like convex optimization
[13] Chen Y, Chi Y, Fan J and Ma C 2018 Gradient descent with random initialization: Fast Global Convergence for Nonconvex Phase Retrieval Math. Program.
[14] Ghesmati A, Bangerth W and Turcksin B 2019 Residual-based a posteriori error estimation for ftp-adaptive finite element methods for the stokes equations J. numer. Math. 27(4) 237-252
[15] Hayhurst B, Keller M, Rai C, Sun X and Westphal C 2018 Adaptively weighted least squares finite element methods for partial differential equations with singularities Comm. App. Math. Com. Sc. 13(1) 1-25

[16] Prato Torres R, Domínguez C and Díaz S 2018 An adaptive finite element method for a time-dependent stokes problem Numer. Meth. Part. D. E.

[17] Lu C, Huang W and Qiu J 2018 An adaptive moving mesh finite element solution of the Regularized long wave equation J. Sci. Comput. 74(1) 122-144

[18] Madzvamuse A, Wathen A J and Maini P K 2003 A moving grid finite element method applied to a model biological pattern generator J. Comput. Phys. 190(2) 478-500