Applications of Deep Learning Framework to Accelerate the Solutions of Parabolic PDEs

Zhang Maoqi, Mathematics Supervised by Dr. Li Guanglian

Zhang Maoqi 3035534347
Mathematics
Research Colloquium for Science UG Students 2021-22

Abstract - This study aims at exploring the applications of the machine learning framework proposed by S. Mishra on parabolic PDEs. The framework and corresponding algorithm aim at accelerating the solutions of time-dependent PDEs and ODEs. The method is based on modeling the current numerical methods as an artificial framework with trainable parameters. By building up our loss function and determining the parameters, the PDEs can be solved for particular points in the range of the function. As a continuation of the summer research which mainly focused on ODEs, this report aims at applying the framework to parabolic PDEs and its variation. Basic heat equations and its variation with non-constant coefficient are used as illustrative examples. The efficiency and limitations of the algorithms will be discussed in the applications.

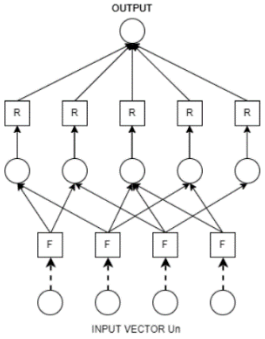
Methodology – machine learning framework and generalized difference method

- Explicit Scheme:
- Implicit Scheme:

Newton's Iteration Method with second order accuracy

(Implicit has more applications for its stability, e.g. backward Euler is unconditionally stable while forward is not)

ML framework in next section



Algorithm 1: Machine Learning Framework PDE solver

Data: A PDE with known constant coefficients and unknown initial boundary condition

Result: A model that can solve given PDEs with any boundary condition and coefficients on the coarse grid

1 Choose a consistent (and stable) numerical method (alternatively neural network) on the coarse grid. At the same time, embed parameters into the scheme. One has the freedom to choose the number or dimensions of the parameters by choosing schemes with different order and also different mesh size.

2 Generate initial data as training sets for the PDEs. (Usually, generate random initial functions as training data. For example, generate Fourier series for Dirichlet boundary condition.)

3 If the PDE is solvable, use exact solution as reference points on the coarse grid. Otherwise, generate the reference points by using classic finite difference method on fine grid and project on the coarse grids as reference points.

4 Set up loss function and use (stochastic) gradient descent method to find the locally optimal parameter. The loss function is the sum of norms of errors of the results calculated in The trained result of the parameter, denoted by θ^*.

$$E(\theta) = \sum_i \sum_{n=1}^{N} \|U^{n,i} - U^{n,i}_{ref}\|^p_{L^p}$$

desired $\theta = \arg\min_{\theta \in D} E(\theta)$ for domain D.

5 Testing: generate a test set to test the accuracy of the model with trained parameters. Calculate the error with respect to the reference points, compared the error with the result of standard parameters (standard scheme like central scheme). Assuming $E(\theta^*) > 0$, define

$$Gain(\theta^*) = \frac{E(\theta_{ref})}{E(\theta^*)}$$

Algorithm 2: Stochastic Gradient Descent (Standard Mini Batch)

Data: objective function $\sum_{i=1}^{n} Q_i(\omega)$, learning rate η_0

Result: ω^*

1 $\omega \leftarrow \omega_0$, $\eta \leftarrow \eta_0$, $n \leftarrow 0$

2 randomly pick m numbers in [1,n] as mini-batch M, calculate

$$\nabla \sum_{i \in M} Q_i(\omega_n) \text{ then, } \omega_{n+1} = \omega_n - \frac{\eta}{m} \nabla \sum_{i \in M} Q_i(\omega_n)$$

,

3 check stopping criteria. (including steps, norm of changes)

4 if the algorithm continues, go back to 2 and repeat.

Examples and Experiments -

1. Ordinary Differential Equations

1.1 linear ODEs

$$u_{tt} + c^2 u(t) = 0, u(0) = u_0$$

c	g_2^*	g_3^*	E_{train}	E_{ref}	Gain
1	-0.0017	0.1088	1.3091	140.33	107
10	-0.8285	1.0932	770.0160	1523.50	1.98
100	5.6346	3.517	522.1608	838.09	1.60

1.2 non-linear ODEs

$$u_x = cu(1-u), u(0) = u_0$$
$$u(t) = \frac{u_0}{u_0 + (1-u_0)e^{-ct}}$$

c	Iterations	g2*	E_{train}	E_{ref}	Gain
0.2	131	0.4230	112.8	146.8	1.3
1	322	0.2310	707	1344.5	1.9
5	721	0.0320	2162.5	20760	9.6

2. Partial Differential Equations

2.1 Heat equation with constant coefficients

i	$\forall n$ g^n	b^n_{-1}	b^n_{-2}	Scheme in time	Scheme in space
1	0	0	1	Backward Euler	second order accurate
2	0	$-\frac{1}{12}$	$\frac{4}{3}$	Backward Euler	4-th order accurate
3	$\frac{1}{2}$	0	1	Crank-Nicolson	second order accurate
4	$\frac{1}{2}$	$-\frac{1}{12}$	$\frac{4}{3}$	Crank-Nicolson	4-th order accurate

2.2 Heat equation with non-constant coefficients

$$\begin{cases} \partial_t u - \partial_x(c(x)\partial_x u) = 0 \\ u|_{t=0} = u_0(x) \end{cases} \quad \text{where } c(x) = (1 + \frac{1}{2}sin(\frac{x}{\epsilon})) \text{ and } \epsilon \to +\infty$$

c	g_2^*	b^*_{-1}	b^*_{-2}	$Gain_1$	$Gain_2$
0.2	0.55	0	1	87.65	50.43
1	0.45	0	1.08	2.37	2.29
10	0.06	2.4	5.6	5.29	5.64

Discussion -

The key contribution of this study is it clarifies the computational process and potential challenges for constant-coefficient heat equations. The framework highly increases the accuracy of the coarse grid approximations within limited steps for parabolic PDEs such as heat equations. Also, it applied the proposed framework to the non-constant coefficient PDEs, which is a long-standing open problem with limited accuracy on the coarse grid approximation.