

# 基于深度学习的间断有限元求解方法

常芸凡<sup>1</sup>

指导老师：杨顶辉教授<sup>1</sup>、贺茜君副教授<sup>2</sup>

<sup>1</sup>清华大学数学科学系

<sup>2</sup>北京工商大学数学科学系

2022 年 3 月 14 日



- ① 研究背景
- ② 基于深度学习的间断有限元方法
- ③ 数值算例
- ④ 下一步工作安排
- ⑤ 参考文献

- 1 研究背景
- 2 基于深度学习的间断有限元方法
- 3 数值算例
- 4 下一步工作安排
- 5 参考文献

## 研究背景

对于物理空间中具有不连续解的方程，间断有限元(DG)方法精度高、数值频散小且容易处理复杂边界问题 [Cockburn and Shu, 1989]，但DG方法的计算速度不高。

目前有很多利用神经网络来快速求解偏微分方程的思想 [Weinan and Yu, 2017] [Sirignano and Spiliopoulos, 2018]等，我们希望将深度学习引入到DG方法中。

## 1 研究背景

## 2 基于深度学习的间断有限元方法

深度神经网络简介

声波方程的间断有限元方法

神经网络的近似方法

## 3 数值算例

## 4 下一步工作安排

## 5 参考文献

- 1 研究背景
- 2 基于深度学习的间断有限元方法
  - 深度神经网络简介
  - 声波方程的间断有限元方法
  - 神经网络的近似方法
- 3 数值算例
- 4 下一步工作安排
- 5 参考文献

# DNN简介

Deep neural network(DNN)包含一系列的层，每一层都有若干个神经元与前层和后层神经元相连。神经元通过仿射变换和非线性激活函数连接。该universal approximation theorem(万能近似定理) [Cybenko, 1989]表明，DNN可以近似任何有限维空间的Borel可测函数。

# 神经网络的基本结构

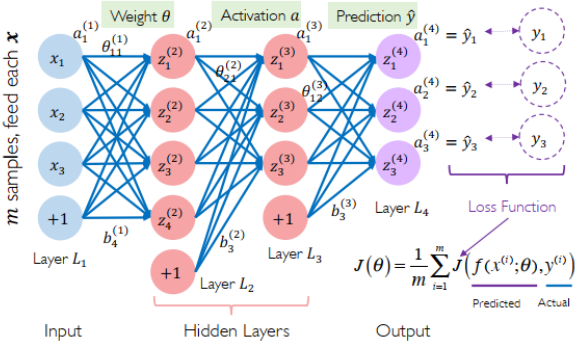
Recursive transformations in propagation

- $a_i^{(1)} = x_i$  #units
- $z_i^{(l)} = \sum_{j=1}^{s_{l-1}} \theta_{ij}^{(l-1)} a_j^{(l-1)} + b_i^{(l-1)}$
- $a_i^{(l)} = g(z_i^{(l)})$
- $\hat{y}_i = a_i^{(n_l)}$  #layers

Superscript:  
layer index

Subscript:  
unit index

Activation function (element-wise)



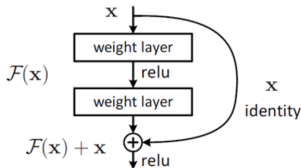
$$J(\theta) = \frac{1}{m} \sum_{i=1}^m J(\underbrace{f(x^{(i)}; \theta)}_{\text{Predicted}}, \underbrace{y^{(i)}}_{\text{Actual}})$$



# ResNet

从经验来看，网络的深度对模型的性能至关重要(一些深度网络可表示的函数可能需要浅层网络指数级的隐藏单元才能表示 [Montúfar et al., 2014])。但是在实际应用中，深层网络存在着梯度消失或者爆炸的问题，这使得深度学习模型很难训练，效果也不好。

为此 [He et al., 2016]提出了残差学习(ResNet)来解决退化问题



这种网络类似于一种短路连接，保证了深层网络的效果至少不会比浅层网络效果差。

# 我们尝试的网络结构

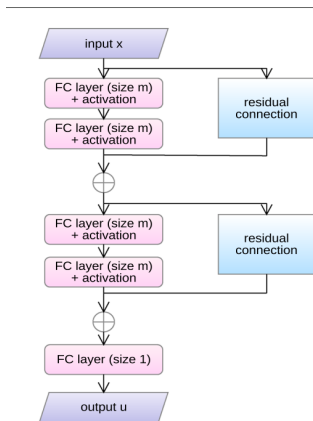


图 1: 残差网络

我们初步尝试的网络结构：3残差块，每层网络的神经元为20。

- 1 研究背景
- 2 基于深度学习的间断有限元方法
  - 深度神经网络简介
  - 声波方程的间断有限元方法
  - 神经网络的近似方法
- 3 数值算例
- 4 下一步工作安排
- 5 参考文献

我们以二维声波方程为例介绍基于通量函数的 DG 方法。二维声波方程的一般形式为：

$$\frac{\partial^2 u}{\partial t^2} = c^2 \left( \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial z^2} \right) + f \tag{1}$$

其中  $c$  为声速， $f(t)$  为震源函数。

# 声波方程的双曲守恒形式

引入变量  $p, q$ , 使它们分别满足:

$$\frac{\partial p}{\partial t} = \frac{\partial u}{\partial x}, \quad \frac{\partial q}{\partial t} = \frac{\partial u}{\partial z}$$

则方程(1)积分后可改写为:

$$\frac{\partial W}{\partial t} + \nabla \cdot F(W) = \hat{f}, \quad (2)$$

$$\text{其中 } W = (u, p, q)^T, F(W) = \begin{pmatrix} -c^2 p & -c^2 q \\ -u & 0 \\ 0 & -u \end{pmatrix}, \hat{f} = \begin{pmatrix} f \\ 0 \\ 0 \end{pmatrix}$$

# 空间离散

设区域为  $\Omega$ , 首先将  $\Omega$  划分为互不相交的单元  $\{\Omega_i\}$  。我们采用的试验函数空间为:

$$V_h = \left\{ v \in L^1(\Omega) : v|_{\Omega_i} \in P^k(\Omega_i) \right\}$$

其中  $P^k(\Omega_i)$  表示单元  $\Omega_i$  上至多  $k$  次多项式。  
由变分法, 我们将问题转化为

$$\int_{\Omega_i} \left( v \frac{\partial \mathbf{W}}{\partial t} - \mathbf{F}(\mathbf{W}) \cdot \nabla v \right) dV + \int_{\partial \Omega_i} v \hat{\mathbf{F}}(\mathbf{W}_i, \mathbf{W}_e, \mathbf{n}) dS = \int_{\Omega_i} \hat{\mathbf{f}}_v dV \tag{3}$$

其中  $\hat{\mathbf{F}}(\mathbf{W}_i, \mathbf{W}_e, \mathbf{n})$  表示依赖于边界内外两侧数值通量。

# 数值通量和基函数的选择

我们采用局部Lax-Friedrichs通量：

$$\hat{F}^{LLF}(a,b,n) = \frac{1}{2}(F(a) + F(b)) \cdot n - \frac{C}{2}(b-a)$$

其中  $C = \max_{\min(a,b) \leq s \leq \max(a,b)} |F'(s) \cdot n|$ 。

基函数方面，我们选取正交Legendre基函数：

$$\begin{aligned}\phi_0(x) &= 1 \\ \phi_1(x) &= \frac{2}{\Delta x} \left(x - x_{i+\frac{1}{2}}\right) \\ \phi_2(x) &= \frac{6}{\Delta x^2} \left(x - x_{i+\frac{1}{2}}\right)^2 - \frac{1}{2} \\ &\dots\end{aligned}$$

# TVD时间离散格式

一般的TVD时间离散为：

$$\begin{aligned} (u^h)^{(i)} &= \sum_{\ell=0}^{i-1} \left[ \alpha_{i\ell} (u^h)^{(\ell)} + \beta_{i\ell} \Delta t L_h \left( (u^h)^{(\ell)}, t^n + d_\ell \Delta t \right) \right], \\ i &= 1, \dots, r \end{aligned}$$

其中， $(u^h)^{(0)} = (u^h)^n$ ， $(u^h)^{(r)} = (u^h)^{n+1}$

我们常用的两种格式为：

- 2阶格式(r=2):

$$\alpha_{10} = \beta_{10} = 1, \alpha_{20} = \alpha_{21} = \beta_{21} = \frac{1}{2}, \beta_{20} = 0; d_0 = 0, d_1 = 1;$$

- 3阶格式(r=3):

$$\begin{aligned} \alpha_{10} = \beta_{10} = 1, \alpha_{20} = \frac{3}{4}, \beta_{20} = 0, \alpha_{21} = \beta_{21} = \frac{1}{4}, \alpha_{30} = \\ \frac{1}{3}, \beta_{30} = \alpha_{31} = \beta_{31} = 0, \alpha_{32} = \beta_{32} = \frac{2}{3}; d_0 = 0, d_1 = 1, d_2 = \\ \frac{1}{2}; \end{aligned}$$



具体而言，二阶精度的TVD时间离散格式为：

$$C^{(0)} = C^{(n)}$$

$$C^{(1)} = C^{(0)} + \Delta t L \left( C^{(0)} \right)$$

$$C^{(2)} = \frac{1}{2} C^{(0)} + \frac{1}{2} C^{(1)} + \frac{1}{2} \Delta t L \left( C^{(1)} \right)$$

$$C^{(n+1)} = C^{(2)}$$

三阶精度的TVD时间离散格式为：

$$C^{(0)} = C^{(n)}$$

$$C^{(1)} = C^{(0)} + \Delta t L \left( C^{(0)} \right)$$

$$C^{(2)} = \frac{3}{4} C^{(0)} + \frac{1}{4} C^{(1)} + \frac{1}{4} \Delta t L \left( C^{(1)} \right)$$

$$C^{(2)} = \frac{1}{3} C^{(0)} + \frac{2}{3} C^{(2)} + \frac{2}{3} \Delta t L \left( C^{(2)} \right)$$

$$C^{(n+1)} = C^{(3)}$$

- 1 研究背景
- 2 基于深度学习的间断有限元方法
  - 深度神经网络简介
  - 声波方程的间断有限元方法
  - 神经网络的近似方法
- 3 数值算例
- 4 下一步工作安排
- 5 参考文献

我们选取基函数 $\varphi_i^j$ 为单元 $I_i$ 上的 $j$ 阶正交Legendre多项式(在其他单元上该函数值为0)，则DG方法最终的解可以写成

$$W_h(t, \mathbf{x}) = \sum_{j=0}^K \sum_{i=0}^M C_i^j(t) \varphi_i^j(x)$$

我们可以利用神经网络来近似系数 $C_i^j$ ，具体而言，我们把解写成如下形式 [Chen et al., 2021]:

$$W_{h,\theta}(t, \mathbf{x}) = \sum_{j=0}^K \sum_{i=0}^M \mathcal{N}_{\theta}^j(t, x_{i+1/2}) \varphi_i^j(x) \tag{4}$$

这里我们一共用了 $K + 1$ 个神经网络 $\mathcal{N}_{\theta}^0, \mathcal{N}_{\theta}^1, \dots, \mathcal{N}_{\theta}^K$ ，每个神经网络的输入是时间 $t$ 和空间坐标 $x$ ，我们希望神经网络的输出 $\mathcal{N}_{\theta}^j(t, x_{i+1/2}) = C_i^j(t)$

# 声波方程推导

把神经网络的格式(4)带入声波方程的DG格式(3)，我们可以得到：

$$\begin{aligned} \frac{\partial \mathcal{N}_{\theta}^j(t, x_{i+1/2})}{\partial t} &= (\varphi_i^j(x), \varphi_i^j(x))^{-1} [\int_{\Omega_i} \mathbf{F}(\mathbf{W}) \cdot \nabla \varphi_i^j(x) \\ &\quad - \int_{\partial \Omega_i} \varphi_i^j(x) \hat{\mathbf{F}}(\mathbf{W}_i, \mathbf{W}_e, \mathbf{n}) dS + \int_{\Omega_i} \hat{\mathbf{f}}_v dV] \end{aligned} \tag{5}$$

我们把方程简单的记作：

$$\frac{\partial \mathcal{N}_{\theta}^j(t, x_{i+1/2})}{\partial t} = L(\mathcal{N}_{\theta}(t, x)) \tag{6}$$

# 声波方程TVD时间离散推导

由方程(6)，我们利用TVD时间离散，如2阶格式：

$$\begin{aligned} C^{(0)} &= \mathcal{N}_{\theta}^j(t_n, x_{i+1/2}) \\ C^{(1)} &= C^{(0)} + \Delta t L \left( C^{(0)} \right) \\ C^{(2)} &= \frac{1}{2} C^{(0)} + \frac{1}{2} C^{(1)} + \frac{1}{2} \Delta t L \left( C^{(1)} \right) \\ C^{(n+1)} &= C^{(2)} \end{aligned}$$

按照传统方法，我们计算的 $C^{(n+1)}$ 应该是 $n+1$ 时间层的系数，也就是 $C^{(n+1)} = \mathcal{N}_{\theta}^j(t_{n+1}, x_{i+1/2})$ 应当成立，所以我们设

$$L_{i,j,n} = C^{(n+1)} - \mathcal{N}_{\theta}^j(t_{n+1}, x_{i+1/2})$$

# 损失函数

对应的第 $j$ 阶的损失函数我们设为：

$$\mathcal{L}_j(\theta) = \left( \sum_{i,n} L_{i,n}^2 \right)^{1/2} + \lambda \|\theta\|_2 \quad j = 0, \dots, K$$

这里的第二项为正则项，目的是防止神经网络的过拟合现象。  
为了减少计算量，我们在上式中并未对所有 $i, n$ 求和，我们每次随机的选取batch个 $i, n$ 进行小批量的运算，这样既可以加快我们的训练速度，效果也不错。

## 1 研究背景

## 2 基于深度学习的间断有限元方法

## 3 数值算例

Burgers' 方程  
1维声波方程

## 4 下一步工作安排

## 5 参考文献

## 1 研究背景

## 2 基于深度学习的间断有限元方法

## 3 数值算例

Burgers' 方程

1维声波方程

## 4 下一步工作安排

## 5 参考文献



我们考虑如下的Burgers方程：

$$u_t + \left(\frac{u^2}{2}\right)_x = 0 \tag{7}$$

初值条件为：

$$u(0, x) = \begin{cases} 1 & , x < 0 \\ 0 & , x > 0 \end{cases}$$

我们选取Burgers方程，一方面可以验证间断有限元方法对于具有间断性质问题的可行性，一方面也是验证间断有限元与深度学习结合后是否还能具有良好的效果。

方程(7)具有间断解析解：

$$u(t,x)=\begin{cases} 1 & ,x < t/2 \\ 0 & ,x > t/2 \end{cases}$$

我们将区间 $[-1,1]$ 均分成 $N$ 份，也就是分为 $N$ 个大小相同的单元 $l_1, l_2, \cdots, l_N$

Burgers'方程

## 间断有限元数值结果

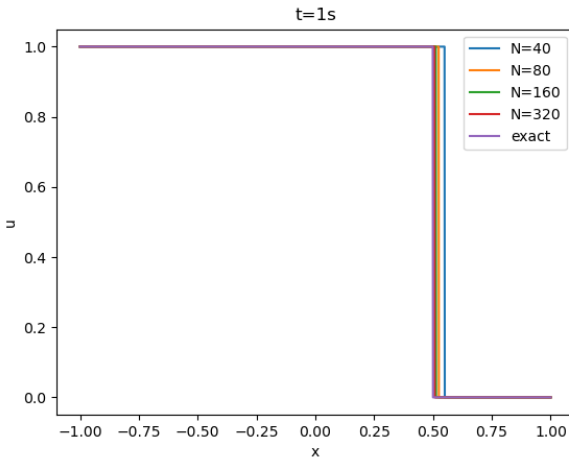


图 2: 0阶间断有限元方法数值结果

Burgers'方程

# 深度学习间断有限元数值结果

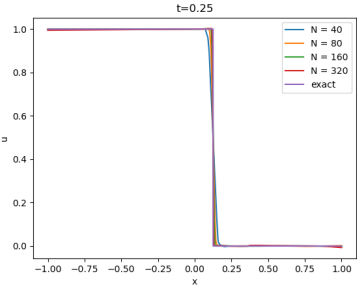


图 3:  $t = 0.25$

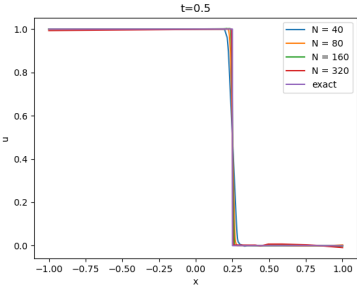


图 4:  $t = 0.5$

Burgers'方程

# 深度学习间断有限元数值结果

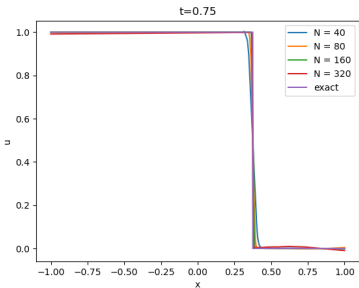


图 5:  $t = 0.75$

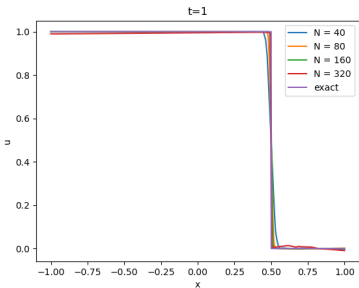


图 6:  $t = 1.0$

Burgers'方程

# 数据对比

传统DG方法	网格数N	用时(s)	相对误差(%)
	40	1.569	18.257
	80	2.710	12.909
	160	8.322	9.309
	320	28.157	6.831
深度学习DG方法	网格数N	相对误差达6.831%用时(s)	训练两万步相对误差(%)
	40	8.717	3.210
	80	7.820	2.164
	160	12.799	1.662
	320	18.774	1.086

图 7: burgers'方程数据对比

我们的计算区域是 $[x, t] \in [-1, 1] \times [0, 1]$ ，但是我们用训练好的神经网络预测1.05s和1.1s的波场如图：

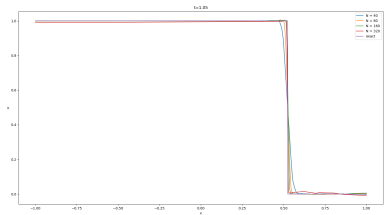


图 8:  $t = 1.05$

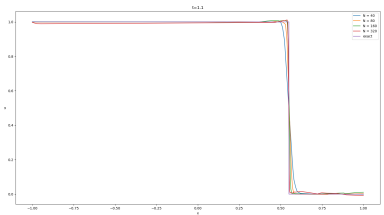


图 9:  $t = 1.1$

- 1 研究背景
- 2 基于深度学习的间断有限元方法
- 3 数值算例
  - Burgers' 方程
  - 1维声波方程
- 4 下一步工作安排
- 5 参考文献



我们考虑如下的一维声波方程

$$\frac{\partial^2 u}{\partial t^2} - c^2 \frac{\partial^2 u}{\partial x^2} = 0$$
(8)

初值条件为

$$\begin{cases} u(x, 0) = \cos\left(-\frac{2\pi f_0}{c}x\right) \\ \frac{\partial u(x, 0)}{\partial t} = -2\pi f_0 \sin\left(-\frac{2\pi f_0}{c}x\right) \end{cases}$$

该方程有解析解：

$$u(t, x) = \cos\left(2\pi f_0 t - \frac{2\pi f_0}{c}x\right)$$

计算区域为 $0 \leq x \leq 4\text{km}, 0 \leq t \leq 0.5\text{s}$ 。

## 0阶格式结果

我们采用LLF通量，并用二阶TVD时间离散格式得到结果如下：

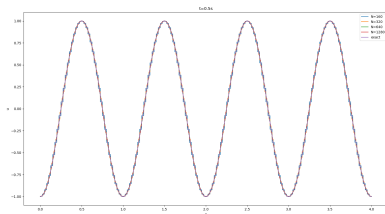


图 10: 传统DG方法0阶格式结果  $t = 0.5$

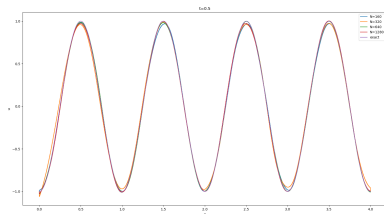


图 11: 深度学习DG方法0阶格式结果  $t = 0.5$

## 0阶格式结果数据对比

传统DG方法	网格数 N	用时(s)	相对误差 (%)	存储波场所用存储空间 (KB)	
	160	14.38	7.78	534	
	320	52.27	3.91	1920	
	640	207.39	1.96	6827	
	1280	825.53	0.98	26454	
深度学习DG方法	网格数 N	训练5千步用时 (s)	相对误差 (%)	存储波场所用存储空间 (KB)	相对误差达5%用时 (s)
	160	413	2.60	42	79.4
	320	369	4.37	42	271.2
	640	436	1.52	42	87.2
	1280	365	1.41	42	146

图 12: 声波方程0阶格式数据对比

在时间和存储空间上有优势。

# 深度学习DG方法的一些优势

传统方法只能得到诸如 $\Delta t, 2\Delta t, \dots, N\Delta t$ 时刻的波场信息，但是深度学习的方法可以得到任何时间的波场信息。

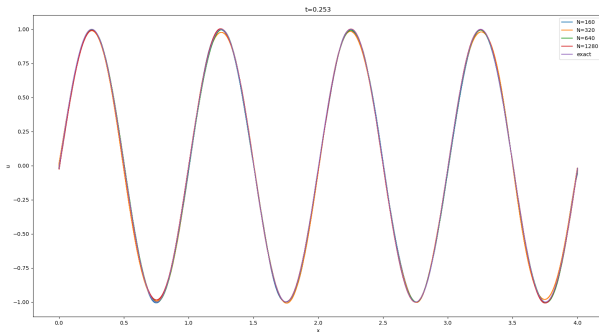


图 13: 深度学习DG方法0阶格式结果 $t = 0.253$

# 深度学习DG方法的一些优势

传统方法一定无法预测超过计算区域的时刻的波场信息，但是深度学习的方法可以在一定程度上预测波场信息。

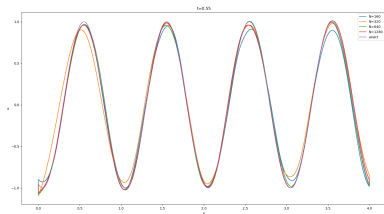


图 14: 深度学习DG方法0阶格式结果  $t = 0.55$

网格数N	0.55s的相对误差(%)
160	7.508
320	10.77
640	4.00
1280	4.50

图 15:  $t = 0.55s$ 误差数据

1维声波方程

# 1阶格式结果

我们采用LLF通量，并用二阶TVD时间离散格式得到结果如下：

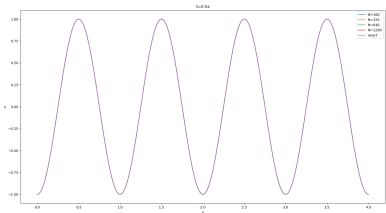


图 16: 传统DG方法1阶格式结果  $t = 0.5$

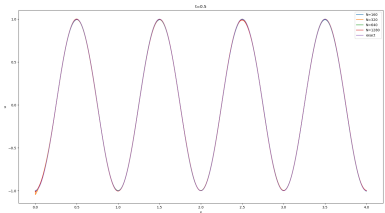


图 17: 深度学习DG方法1阶格式结果  $t = 0.5$

# 1阶格式结果数据对比

传统DG方法	网格数 N	用时(s)	相对误差 (%)	存储波场所用存储空间 (KB)	
	160	24.31	0.202	1068	
	320	95.14	0.051	3840	
	640	383.12	0.013	13654	
	1280	1993.51	0.003	52908	
深度学习DG方法	网格数 N	训练5千步用时 (s)	相对误差 (%)	存储波场所用存储空间 (KB)	相对误差达1%用时 (s)
	160	740	0.547	84	78
	320	722	0.811	84	195
	640	702	0.441	84	75
	1280	706	0.489	84	112

图 18: 声波方程1阶格式数据对比

在时间和存储空间上有优势。

# 深度学习DG方法的一些优势

传统方法只能得到诸如 $\Delta t, 2\Delta t, \dots, N\Delta t$ 时刻的波场信息，但是深度学习的方法可以得到任何时间的波场信息。

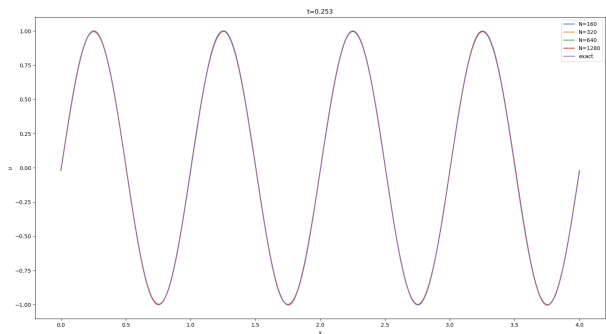


图 19: 深度学习DG方法1阶格式结果 $t = 0.253$



# 深度学习DG方法的一些优势

传统方法一定无法预测超过计算区域的时刻的波场信息，但是深度学习的方法可以在一定程度上预测波场信息。

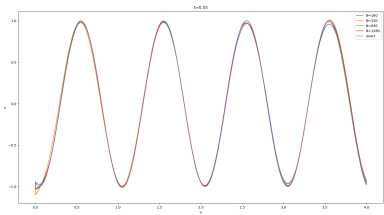


图 20: 深度学习DG方法1阶格式结果 $t = 0.55$

网格数N	0.55s的相对误差(%)
160	1.698
320	2.773
640	1.428
1280	1.525

图 21:  $t = 0.55s$ 误差数据

## 2阶格式结果

我们采用LLF通量，并用二阶TVD时间离散格式得到结果如下：

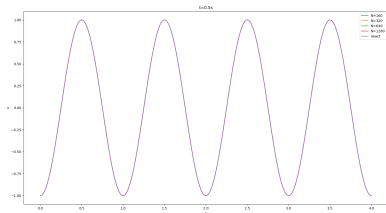


图 22: 传统DG方法2阶格式结果  $t = 0.5$

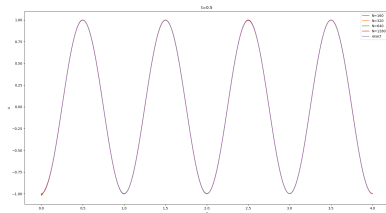


图 23: 深度学习DG方法2阶格式结果  $t = 0.5$

## 2阶格式结果数据对比

传统DG方法	网格数 N	用时(s)	相对误差( %)	存储波场所用存储空间 (KB)	
	160	94.36	0.259	5760	
	320	374.85	0.057	10602	
	640	778.66	0.013	20480	
	1280	3280.29	0.003	79360	
深度学习DG 方法	网格数 N	训练5千步用 时(s)	相对误差 (%)	存储波场所用存储空间 (KB)	相对误差达0.5%用 时(s)
	160	999	0.243	126	104
	320	1013	0.383	126	161
	640	1002	0.205	126	216
	1280	992	0.260	126	104

图 24: 声波方程2阶格式数据对比

在时间和存储空间上有优势。

# 深度学习DG方法的一些优势

传统方法只能得到诸如 $\Delta t, 2\Delta t, \dots, N\Delta t$ 时刻的波场信息，但是深度学习的方法可以得到任何时间的波场信息。

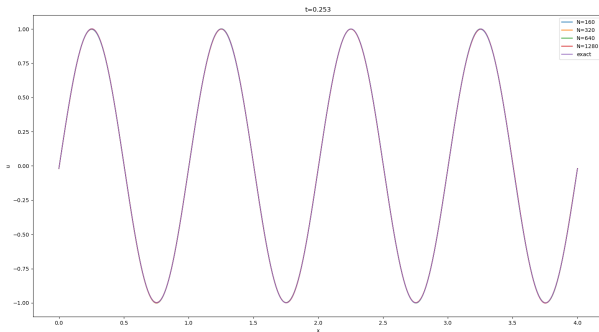


图 25: 深度学习DG方法2阶格式结果 $t = 0.253$

# 深度学习DG方法的一些优势

传统方法一定无法预测超过计算区域的时刻的波场信息，但是深度学习的方法可以在一定程度上预测波场信息。

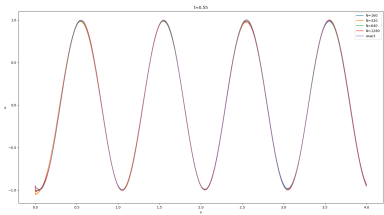


图 26: 深度学习DG方法2阶格式结果 $t = 0.55$

网格数N	0.55s的相对误差(%)
160	1.272
320	2.621
640	1.310
1280	1.451

图 27:  $t = 0.55s$ 误差数据

# 收敛阶的猜想

传统DG格式的收敛阶为 $O(h^{k+1})$ ，其中 $k$ 是近似的阶数， $h$ 为空间步长，但是深度学习DG方法总是一个近似方法，由于在0阶项就在近似，所以猜测深度学习的近似阶实际上是 $O(h/(k + 1)^3)$

传统DG方法	网格数 N	0阶格式0.5s相对误差 (%)	1阶格式0.5s相对误差 (%)	2阶格式0.5s相对误差( (%)
	160	7.78	0.202	0.259
	320	3.91	0.051	0.057
	640	1.96	0.013	0.013
	1280	0.98	0.003	0.003
深度学习DG方法	网格数 N	0阶格式0.5s相对误差 (%)	1阶格式0.5s相对误差 (%)	2阶格式0.5s相对误差 (%)
	160	2.60	0.547	0.243
	320	4.37	0.811	0.383
	640	1.52	0.441	0.205
	1280	1.41	0.489	0.260

图 28: 误差对比

- 1 研究背景
- 2 基于深度学习的间断有限元方法
- 3 数值算例
- 4 下一步工作安排
- 5 参考文献

## 下一步工作安排

- ① 向二维、三维空间尝试
- ② 对误差阶数进行理论或者数值研究



- ① 研究背景
- ② 基于深度学习的间断有限元方法
- ③ 数值算例
- ④ 下一步工作安排
- ⑤ 参考文献

# 参考文献 I

- [Chen et al., 2021] Chen, J., Jin, S., and Lyu, L. (2021).  
A deep learning based discontinuous galerkin method for hyperbolic equations with discontinuous solutions and random uncertainties.  
*arXiv preprint arXiv:2107.01127*.
- [Cockburn and Shu, 1989] Cockburn, B. and Shu, C.-W. (1989).  
Tvb runge-kutta local projection discontinuous galerkin finite element method for conservation laws ii: General framework.  
*Mathematics of Computation*, 52(186):411–435.
- [Cybenko, 1989] Cybenko, G. (1989).  
Approximation by superpositions of a sigmoidal function.  
*Mathematics of control, signals and systems*, 2(4):303–314.
- [He et al., 2016] He, K., Zhang, X., Ren, S., and Sun, J. (2016).  
Deep residual learning for image recognition.  
*In Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.
- [Montúfar et al., 2014] Montúfar, G., Pascanu, R., Cho, K., and Bengio, Y. (2014).  
On the number of linear regions of deep neural networks.  
*arXiv preprint arXiv:1402.1869*.

## 参考文献 II

[Sirignano and Spiliopoulos, 2018] Sirignano, J. and Spiliopoulos, K. (2018).

Dgm: A deep learning algorithm for solving partial differential equations.

*Journal of Computational Physics*, 375:1339–1364.

[Weinan and Yu, 2017] Weinan, E. and Yu, T. (2017).

The deep ritz method: A deep learning-based numerical algorithm for solving variational problems.

*Communications in Mathematics and Statistics*, 6:1–12.

*Thanks!*