

# 课后练习： DeepONet 反导数算子学习：架构与采样策略影响分析报告

刘昭阳 25215133  
中山大学数学学院（珠海）  
2026 年 1 月 6 日

## 摘要

本报告旨在深入探究 DeepONet 在学习反导数算子任务中的性能表现。我们不仅评估了基准模型的精度，还系统地分析了网络深度（Network Depth）和域内采样密度（Sampling Density）对模型泛化能力的影响。实验结果表明，基准模型在全量数据下表现最佳，测试集相对误差仅为 1.47%。盲目增加网络深度会导致过拟合，而稀疏的域内采样则会显著削弱模型捕捉函数细节的能力。

# 目录

<b>1 引言</b>	<b>3</b>
1.1 背景 . . . . .	3
1.2 DeepONet 简介 . . . . .	3
1.3 任务定义 . . . . .	3
<b>2 方法论</b>	<b>3</b>
2.1 DeepONet 架构 . . . . .	3
2.2 损失函数 . . . . .	4
<b>3 实验设置</b>	<b>4</b>
3.1 数据集 . . . . .	4
3.2 模型架构变体 . . . . .	4
<b>4 结果与分析</b>	<b>4</b>
4.1 训练动态分析 . . . . .	4
4.2 定量性能评估 . . . . .	5
4.3 预测结果可视化 . . . . .	6
4.4 误差分布分析 . . . . .	6
4.5 绝对误差沿定义域分布 . . . . .	7
<b>5 讨论</b>	<b>7</b>
5.1 网络深度的权衡 . . . . .	7
5.2 采样策略的重要性 . . . . .	7
<b>6 结论</b>	<b>8</b>

# 1 引言

## 1.1 背景

在科学计算和工程领域，偏微分方程（PDE）和积分方程的求解至关重要。传统的数值方法（如有限元法 FEM、有限差分法 FDM）虽然精度高，但计算成本高昂，且通常针对特定方程和边界条件进行求解。近年来，基于深度学习的算子学习（Operator Learning）方法应运而生，旨在学习函数空间之间的映射，从而实现对一类方程的快速求解。

## 1.2 DeepONet 简介

DeepONet (Deep Operator Network) 是由 Lu 等人提出的一种通用的算子学习框架，其理论基础是算子逼近定理 (Universal Approximation Theorem for Operators)。DeepONet 通过两个子网络——Branch Net (分支网络) 和 Trunk Net (主干网络) ——分别编码输入函数和查询位置，从而实现对非线性算子的有效逼近。

## 1.3 任务定义

本实验聚焦于一个经典的算子学习任务：**反导数算子 (Antiderivative Operator)**。给定一个定义在  $[0, 1]$  区间上的函数  $u(x)$ ，我们的目标是学习算子  $G$ ，使得  $G(u) = v$ ，其中  $v(x)$  是  $u(x)$  的反导数，即：

$$v(x) = \int_0^x u(\tau) d\tau \quad (1)$$

这是一个典型的积分算子学习问题。

# 2 方法论

## 2.1 DeepONet 架构

DeepONet 的核心思想是将算子  $G$  近似为：

$$G(u)(y) \approx \sum_{k=1}^p b_k(u(x_1), u(x_2), \dots, u(x_m)) \cdot t_k(y) + b_0 \quad (2)$$

其中：

- $u(x_1), \dots, u(x_m)$  是输入函数  $u$  在  $m$  个固定传感器位置上的离散值。
- $b_k(\cdot)$  是 Branch Net 的输出，负责提取输入函数的特征。
- $t_k(\cdot)$  是 Trunk Net 的输出，负责提取查询位置  $y$  的特征。
- $p$  是两个子网络输出层的神经元个数。

## 2.2 损失函数

我们采用均方误差 (MSE) 作为损失函数来训练网络参数  $\theta$ :

$$\mathcal{L}(\theta) = \frac{1}{N} \sum_{i=1}^N \sum_{j=1}^P |G_\theta(u^{(i)})(y_j^{(i)}) - v^{(i)}(y_j^{(i)})|^2 \quad (3)$$

其中  $N$  是训练样本数,  $P$  是每个样本的域内采样点数。

## 3 实验设置

### 3.1 数据集

数据集包含成对的函数  $(u, v)$ , 其中  $u$  是输入函数,  $v$  是对应的反导数。

- **训练集:** 加载自 `antiderivative_aligned_train.npz`。
- **测试集:** 加载自 `antiderivative_aligned_test.npz`。

### 3.2 模型架构变体

为了全面评估 DeepONet 的鲁棒性, 我们构建了三种不同的实验配置, 如表 1 所示。

表 1: 实验配置详情

实验组	Branch Net 结构	Trunk Net 结构	域内采样点数 ( $N$ )	说明
Baseline (基准)	[100, 40, 40]	[1, 40, 40]	100	标准配置
Deeper Net (深层)	[100, 40, 40, 40]	[1, 40, 40, 40]	100	增加网络深度
Sparse (稀疏)	[100, 40, 40]	[1, 40, 40]	20	仅使用 20% 采样点

所有模型均使用 ReLU 激活函数和 Glorot normal 初始化, 采用 Adam 优化器 (学习率 0.001) 训练 10,000 次迭代。

## 4 结果与分析

### 4.1 训练动态分析

图 1 展示了三个模型在训练过程中的 Loss 下降曲线 (对数坐标)。

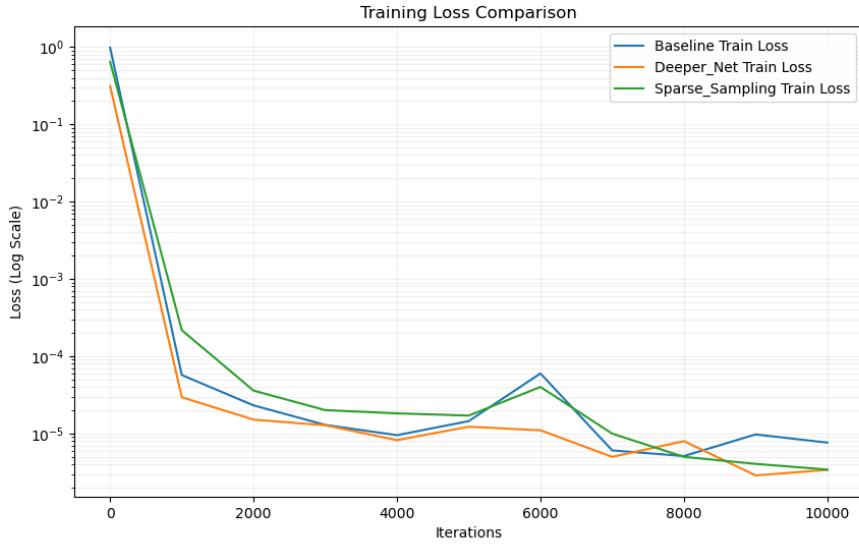


图 1: 不同模型的训练损失历史对比 (Log Scale)

从图中可以观察到：

- **Baseline (蓝色)**: 收敛最为平稳，且最终 Loss 达到了较低的水平，表明优化过程顺利。
- **Deeper Net (橙色)**: 虽然最终 Loss 也很低，但在训练初期和中期出现了较大的震荡。这可能是因为更深的网络带来了更复杂的损失地形 (Loss Landscape)，使得优化器更难找到全局最优解。
- **Sparse (绿色)**: 由于数据量减少，Loss 曲线整体高于其他两组，且收敛速度较慢，表明模型难以从稀疏数据中学习到有效的算子映射。

## 4.2 定量性能评估

表 2 列出了各模型在测试集上的最终性能指标。

表 2: 最终性能指标对比

模型	Train Loss	Test Loss	Test Metric (L2 Rel Error)
Baseline	$7.61 \times 10^{-6}$	$1.54 \times 10^{-5}$	1.47%
Deeper Net	$3.38 \times 10^{-6}$	$2.61 \times 10^{-5}$	2.11%
Sparse Sampling	$3.41 \times 10^{-6}$	$1.75 \times 10^{-5}$	1.70%

分析：

- **Baseline 表现最佳**: 在测试集上取得了最低的相对误差 (1.47%)，且 Train Loss 与 Test Loss 差距最小，泛化能力最强。
- **Deeper Net 过拟合**: 虽然 Train Loss 较低 ( $3.38 \times 10^{-6}$ )，但 Test Loss 显著高于 Baseline，Test Metric 也最差 (2.11%)。这表明在当前数据规模下，增加网络深度导致了严重的过拟合，模型记住了训练数据但未能很好地泛化。

- **Sparse Sampling 欠拟合:** 虽然 Train Loss 极低 (可能是因为数据点少容易拟合), 但 Test Metric 较差 (1.70%)。由于采样点不足, 模型无法捕捉函数的高频细节, 导致在未见过的测试点上预测误差较大。

### 4.3 预测结果可视化

为了直观展示模型性能, 我们选取了三个不同的测试样本, 对比了各模型的预测曲线 (图 2)。

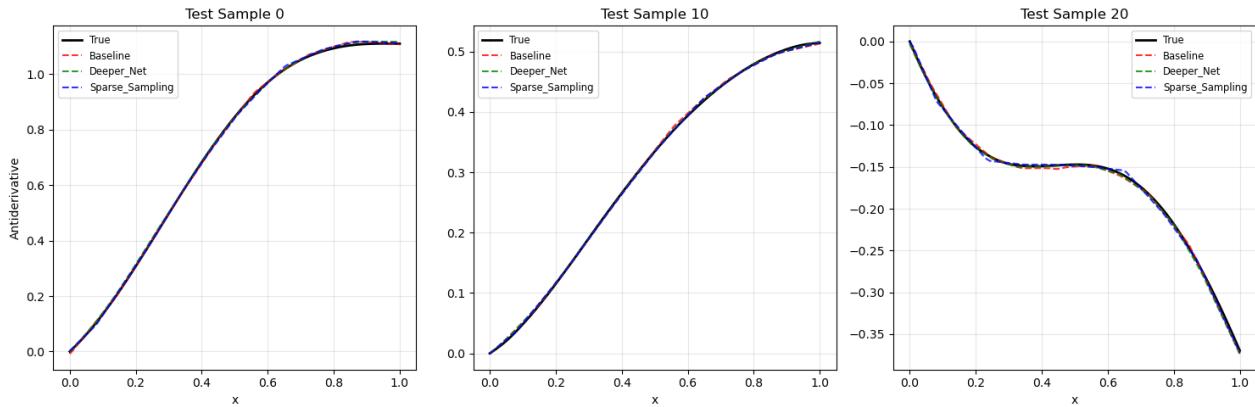


图 2: 三个测试样本的预测结果对比

可以看出, Baseline 模型 (红色虚线) 与真实值 (黑色实线) 贴合最为紧密。Deeper Net (绿色虚线) 在某些局部区域 (如波峰波谷) 出现了明显的偏差, 这与其过拟合的特性一致。

### 4.4 误差分布分析

图 3 展示了测试集上所有样本的相对 L2 误差分布直方图。

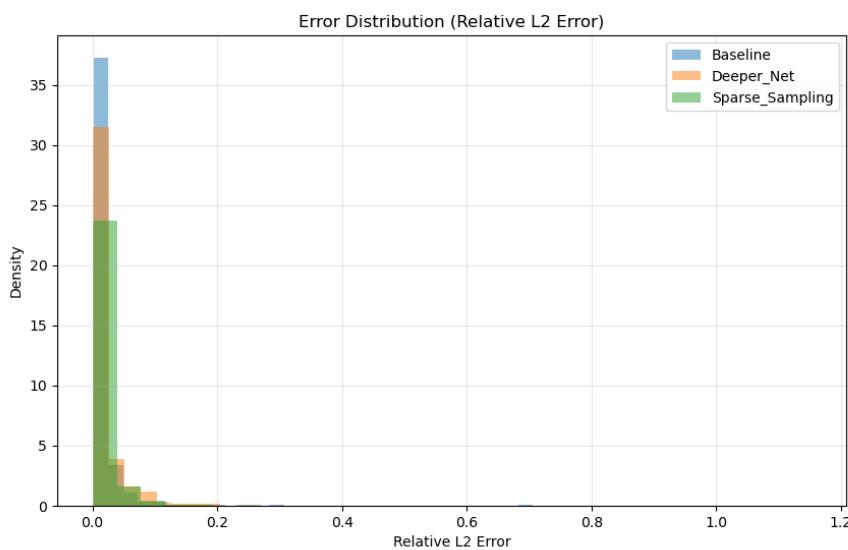


图 3: 测试集相对 L2 误差分布直方图

Baseline 模型的误差分布最集中于 0 附近，且长尾效应最小。Deeper Net 的分布明显右移，说明存在更多的高误差样本。

#### 4.5 绝对误差沿定义域分布

为了更细致地观察模型在定义域内的表现，我们绘制了第一个测试样本在  $[0, 1]$  区间上的绝对误差分布（图 4）。

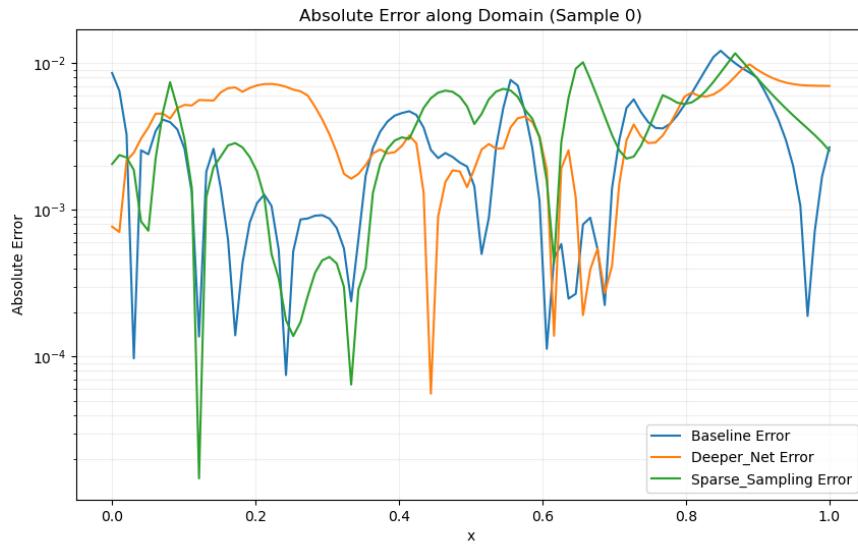


图 4: 单样本绝对误差沿定义域分布 (Log Scale)

从图中可以看出，Baseline 模型的误差（蓝色线）在整个定义域内都保持在较低水平。而 Deeper Net（橙色线）在某些位置的误差会出现尖峰，这进一步证实了其预测的不稳定性。

## 5 讨论

### 5.1 网络深度的权衡

本实验结果表明，对于相对简单的反导数算子，盲目增加网络深度并不能带来性能提升。相反，过深的网络增加了参数量，使得模型更容易过拟合，同时也增加了训练难度。在实际应用中，应根据任务的复杂度和数据量来选择合适的网络深度，或者引入正则化技术（如 Dropout、L2 正则化）来缓解过拟合。

### 5.2 采样策略的重要性

稀疏采样实验揭示了数据质量对算子学习的关键影响。根据奈奎斯特采样定理，如果采样频率低于信号最高频率的 2 倍，就会发生混叠，导致无法完美重构信号。在算子学习中，如果域内采样点过少，模型就无法感知输入函数的细微变化，从而限制了其逼近精度。因此，保证足够的采样密度是提高模型性能的必要条件。

## 6 结论

本实验通过对比分析得出以下结论：

1. **架构选择**: 对于反导数算子学习任务, [40, 40] 的隐藏层规模已经足够。盲目加深网络不仅增加了计算开销, 还导致了泛化性能下降。
2. **采样重要性**: 域内采样密度对 DeepONet 的性能有直接影响。稀疏采样会丢失函数信息, 限制模型的精度上限。
3. **最佳实践**: 在实际应用中, 应优先保证数据的质量和采样密度, 并根据数据规模选择合适的网络深度, 避免过拟合。

未来的工作可以进一步探究不同激活函数、不同初始化方法以及更复杂的算子（如非线性 PDE 算子）对 DeepONet 性能的影响。