DeepONet 기반의 포아송 방정식 해결기의 성능 분석

이상민
한국과학기술정보연구원 국가슈퍼컴퓨팅본부
e-mail: smlee@kisti.re.kr

Performance Evaluation of Poisson Solver Using DeepONet

Sang Min Lee
National Supercomputing Center
Korea Institute of Science & Technology Information,

포아송 방정식은 대표적인 경계치 문제로 타원형 편미분 방정식(Partial Differential Equations, PDE)으로, 중력장 및 전자기장, 유체역학, 열전달 및 확산과 같은 다양한 물리 현상에서 널리 응용된다. 전통적으로 유한 차분법(Finite Difference Methods, FDM)과 유한 요소법(Finite Element Methods, FEM)과 같은 수치적 방법이 포아송 방정식을 해결하는 데 사용되고 있다. 그러나 이러한 접근법은 고해상도 격자 또는 대규모 시뮬레이션에서는 계산 비용과 시간이 많이 소요된다. 이 연구에서는 함수 공간 간의 연산자를 근사하도록 설계된 딥러닝 프레임워크인 DeepONet(Deep Operation Networks)를 적용한 대체 방법을 제안한다. 본 연구에서는 DeepONet을 사용하여 포아송 방정식을 해결할 때의 성능을 정확성, 계산 효율성, 확장성의 세 가지 주요 측면에서 평가한다. 실험 결과, DeepONet은 전통적인 해결 방법과 유사한 정확성을 보일 뿐만 아니라, 특히 대규모 문제에서 계산 시간을 크게 단축시킨다. 또한 DeepONet의 확장성은 실시간 응용 프로그램 및 고차원 문제에 적합하여 DeepONet을 효율적인 PDE 해결자로서의 새로운 가능성을 제안한다.

1. 서론

포아송 방정식(Poisson equations)은 중력장 및 전자기장, 유체역학, 열전달 및 확산을 포함한 다양한 과학 및 공학 분야에서 중심적인 역할을 하는 기본적인 PDE이다. 포아송 방정식은 종종 다음과 같은 형태로 표현된다:

$$\nabla^2 u(x,y) = f(x,y),$$

여기서 $u(x,y)$는 해답(solution)을 나타내는 미지수 함수이며, $f(x,y)$는 주어진 원인항이다. $u(x,y)$에 대한 포아송 방정식을 해결하는 것은 전자기 및 중력 포텐셜, 재료의 온도 분포, 비압축성 유체 흐름과 같은 다양한 물리적 현상을 모델링 하는데 절대적으로 필요하다. 포아송 방정식을 정확하고 효율적인 방법으로 풀기 위한 다양한 계산 방법의 개발이 시도 되어왔다.

전통적으로, 포아송 방정식을 해결하기 위해 유한 차분법(FDM), 유한 요소법(FEM), 유한 체적법(FVM)과 같은 수치적 방법이 사용된다. 이러한 방법은 도메인을 작은 격자점이나 요소로 이산화한 다음, 결과로 생성된 대수 방정식을 반복적 계산으로 해를 찾는 방법을 따른다. 널리 사용되지만 이러한 접근 방식은 특히 고해상도(많은 수의 격자점의 경우) 또는 대규모 문제의 경우에 계산 비용과 시간이 많이 소요될 수 있다. 예를 들어, 이러한 방법의 계산 비용은 격자의 해상도가 증가함에 따라 증가하여 실시간 성능을 요구하거나 대규모 데이터 세트를 처리해야 하는 시뮬레이션에 어려움을 초래한다. 또한 복잡한 경계 조건이나 이질적인 도메인을 처리하는데 어려움을 겪어 추가 수정이나 조정이 필요하게 된다면 계산 비용을 더욱 증가하게 된다.

최근 기계 학습의 발전으로 딥러닝 기반 신경망 학습법이 전통적인 수치 해법의 유망한 대안으로 떠오르고 있다. 신경망은 복잡한 함수를 근사하고 대규모 데이터 세트로부터 각종 함수를 학습하는데 뛰어난 능력을 보여준다. 이러한 신경망은 이미지 분류, 자연어 처리 및 생성 모델링 등 다양한 작업에 성공적으로 적용된다. 이러한 성공은 연구자들에게 과학 컴퓨팅 및 미분 방정식의 해결에 대한 응용을 탐구하도록 영감을 주었다. 이러한 맥락에서 물리정보 신경망(Physics-Informed Neural Networks, PINN[1]), 컨볼루션 신경망(CNNs[2]) 및 최근에는 연산자 학습 네트워크(Operational Networks[3])와 같은 딥러닝 모델이 전통적인 수치 해결 방법에 대한 효율적

인 대안으로 제안되고 있다.

DeepONet[4]는 연산자 학습 네트워크의 일종으로 특정 함수를 근사하는 것뿐만 아니라 전체 함수 공간 간의 매핑 또는 연산자에 대한 학습을 목표로 하는 신경망 아키텍처의 새로운 패러다임이다. 이는 DeepONet이 일련의 특정 데이터 세트나 함수에 제한되지 않고 다양한 입력 함수에 대해 일반화할 수 있는 강력한 도구임을 의미한다. 특히 DeepONet은 입력 함수(예, 포아송 방정식의 원인항($f(x,y)$))에서 출력 함수(예, 해답($u(x,y)$))로의 매핑을 학습하도록 설계되었으며, 이를 위해 입력 함수를 처리하는 브랜치 네트워크와 공간 좌표를 처리하는 트렁크 네트워크의 두 가지 별도의 신경망 구성 요소를 활용한다.

DeepONet은 포아송 방정식과 같은 PDE를 해결하는 데 잠재적으로 많은 이점을 제공한다[5]. 첫째, 훈련이 완료되면 DeepONet은 새로운 입력에 대해 거의 즉시 해결책을 생성할 수 있어 실시간 응용에 적합하다. 둘째, DeepONet의 일반화 능력 덕분에 다양한 입력 함수와 경계 조건을 처리할 수 있어 다양한 유형의 포아송 방정식을 해결하는 유연한 프레임워크를 제공한다. 셋째, DeepONet 구현을 위한 딥러닝 프레임워크의 사용은 GPU 가속의 효율적인 활용을 가능하게 하여 계산 속도와 확장성을 더욱 향상시킬 수 있다. 이러한 특징들은 DeepONet을 전통적인 수치 방법에 대한 매력적인 대안으로 제안하며, 특히 계산 자원이 제한되거나 빠른 해결 시간이 요구되는 경우에서 더욱 유용하게 활용할 수 있다.

DeepONet의 유용한 특성에도 불구하고 해결해야 할 도전과 제한 사항이 있다. 딥 신경망을 훈련하는 것은 계산적으로 집중적인 작업이며, 특히 높은 정확성이 중요한 과학 컴퓨팅 응용 분야에서 안정적인 성능을 위해 대규모 데이터 세트가 필요하다. 또한 DeepONet이 해결 연산자를 근사하는 효율적인 방법을 제공하더라도, 전통적인 방법에 대한 성능을 철저히 평가하여 그 강점, 한계 및 잠재적인 응용 분야를 이해하는 것이 중요하다. 주요 질문은 DeepONet이 정통적 수치 해결 방법과 비교하여 유사한 정확성을 달성할 수 있는지, 문제 크기에 따라 어떻게 확장되는지, 그리고 FDM 및 FEM에 비해 계산 효율성이 어떻게 되는지이다.

본 연구는 포아송 방정식의 해결기로서 DeepONet의 포괄적인 성능 평가를 수행하여 이러한 질문의 답을 찾고자 한다. 구체적으로 우리는 다음을 목표로 한다:

○ **정확성 평가**: DeepONet이 포아송 방정식을 해결하는 정확성을 평가하고, 그 해결책을 전통적인 수치 방법에서 얻은 해결책과 비교한다. 우리는 다양한 원인 함수 및 경계 조건에 따라 평균 제곱 오차(MSE) 및 상대 오차와 같은 여러 정확성 지표를 기초로 분석할 것이다.

○ **계산 효율성 검토**: DeepONet, FDM 및 FEM의 계산 효율성을 추론 시간 및 메모리 사용 측면에서 측정하고 비교한다. 이는 DeepONet이 실시간 응용 프로그램 및 대규모 문제에 적합한지 판단하는 데 도움이 될 것이다.

○ **확장성 평가**: 문제 크기가 증가함에 따라 DeepONet이 얼마나 잘 확장되는지를 조사한다. 특히 격자 해상도가 증가할 때의 성능을 평가한다. 확장성을 이해하는 것은 전통적인 방법이 종종 계산 병목 현상에 직면하는 고차원 도메인 응용에 필수적이다.

이 연구는 DeepONet과 기존의 수치 방법을 비교하여 포아송 방정식의 해결기로서의 DeepONet의 능력과 한계에 대한 포괄적인 이해를 제공하고자 한다. 또한, DeepONet이 특히 실시간 또는 자원이 제한된 환경에서 전통적인 해결 방법에 대한 실행 가능한 대안으로 작용할 수 있는 잠재적인 응용 시나리오에 대한 통찰력을 제공할 것이다.

2. 연구 방법

본 연구는 포아송 방정식의 해결기로서 DeepONet의 성능을 평가하는 것을 목표로 한다. 이 섹션에서는 전체 문제 설정, DeepONet 모델의 아키텍처, 데이터셋 준비 및 훈련 과정을 간략하게 설명한다. 또한 비교를 위해 사용되는 전통적인 기준 해결기에 대해 논의하고, 정확성, 계산 효율성 및 확장성을 평가하기 위한 평가 지표와 실험 설정을 설명한다.

2.1 문제 정의

포아송 방정식은 물리학, 공학 및 수학과 같은 다양한 분야에서 발생하는 널리 사용되는 PDE입니다. 주어진 영역 Ω에 대해 이 방정식은 다음과 같이 쓸 수 있다:

$$\nabla^2 u(x,y) = f(x,y) \text{ in } \Omega \tag{1}$$

여기서 $u(x,y)$는 해답이며, $f(x,y)$는 원인항을 나타낸다. 우리는 다양한 형태의 $f(x,y)$와 여러 경계 조건(예, Dirichlet 또는 Neumann) 하에서 $u(x,y)$를 구해본다. 계산 영역은 $\Omega = [0,1] \times [0,1]$로 정의하였다.

2.2 DeepONET의 아키텍처

DeepONet은 특정 함수보다는 연산자를 학습하도록 설계된 신경망 기반 접근 방식이다. 이를 통해 DeepONet은 다양한 입력함수에 일반화할 수 있다. 본 연구에서 DeepONet은 포아송 방정식의 원인항 $f(x,y)$에서 해답인 $u(x,y)$로의 매핑을 근사하도록 훈련한다.

DeepONet의 아키텍처는 두 가지 주요 구성 요소로 이루어져 있습니다:

- **브랜치 네트워크**(Branch network): 브랜치 네트워크는 원인항 $f(x,y)$를 입력으로 받는다. 우리는 영역에서 샘플링된 유한 집합의 점들을 사용하여 $f(x,y)$를 나타낸다. 이 샘플 값들은 여러 개의 은닉층을 가진 완전 연결 신경망을 통해 전달되어 입력 원인 항의 특징을 추출한다.
- **트렁크 네트워크**(Trunk network): 트렁크 네트워크는 공간 좌표 (x,y)를 직접 처리한다. 브랜치 네트워크와 마찬가지로 여러 개의 완전 연결 층으로 구성되어 있으며, 좌표 정보에서 특징을 인코딩한다.

브랜치 네트워크와 트렁크 네트워크의 출력은 내적(dot product)을 통해 결합되어 최종 해답인 $u(x,y)$을 생성한다. 구체적으로, 브랜치 네트워크의 출력을 $b(f)$로, 트렁크 네트워크의 출력을 $t(x,y)$로 나타낸다. 그러면 해답은 다음과 같이 계산된다:

$$u(x,y) = b(f) \cdot t(x,y) \tag{2}$$

이러한 아키텍처는 DeepONet이 f를 u로 매핑하는 연산자를 학습할 수 있도록 하여, 포아송 방정식의 해결 연산자를 효과적으로 근사하게 된다.

신경망 하이퍼파라미터는 은닉층의 수, 각 층의 뉴런 수 및 활성화 함수 등을 포함하여 일련의 예비 실험을 기반으로 선택하였다. 브랜치 및 트렁크 네트워크의 경우 다음을 사용했다:

- 4개의 은닉층, 각 층마다 128개의 뉴런
- ReLU 활성화 함수(activation function)
- Adam 옵티마이저, 학습률(learning rate, 10^{-4})

2.3 데이터셋 준비

DeepONet을 훈련하고 검증하기 위해, 포아송 방정식을 전통적인 수치 방법으로 해결하여 원인 함수 $f(x,y)$와 그에 해당하는 해답 $u(x,y)$의 합성 데이터셋을 생성하였다. 다음과 같은 다양한 유형의 원인 함수를 고려한다:

- **사인 함수**: $f(x,y) = \sin(k_x x)\sin(k_y y)$ (다양한 k_x, k_y 사용)
- **가우시안 함수**: $f(x,y) = \exp\left(-\frac{(x-x_0)^2 + (y-y_0)^2}{2\sigma^2}\right)$
- **다항식 함수**: 임의의 계수를 가진 저차원 다항식으로 생성된 f(x,y)
- **임의 함수**: 변동성을 도입하기 위해 균일 분포에서 샘플링된 f(x,y)

각 원인 유형에 대해, 우리는 정확한 해결 데이터 생성을 위해 유한 요소 방법을 사용하여 포아송 방정식을 해결하였다. 우리는 서로 다른 수준의 세부 사항을 캡처하기 위해 32×32에서 256×256까지의 총 4개의 그리드 해상도를 사용하였다. 총 약 10,000개의 좌표점(collocation points)에서 원인-해답 쌍을 생성하였다.

3. 결과

이 섹션에서는 포아송 방정식의 해결기로서 DeepONet의 성능 평가 결과를 제시하며, 이를 FDM 및 FEM과 같은 전통적인 수치적 방법과 비교한다. 우리는 성능의 주요 세 가지 측면인 정확도, 계산 효율성 및 확장성을 평가하고 논의한다.

3.1 계산의 정확도

정확도는 포아송 방정식의 해결자의 성능을 평가하는 데 중요한 척도로, 계산된 해가 실제 해에 얼마나 가까운지를 결정한다. 우리는 다양한 소스 함수 및 경계 조건을 가진 테스트 케이스에서 전통적인 수치적 해결자와 DeepONet의 해를 비교하여 정확도를 평가하였다. 구체적으로, 평균 제곱 오차(MSE), 평균 절대 오차(MAE), 상대 오차와 같은 지표를 사용하여 각 방법의 성능을 정량화하였다.

[표 1] 각 해법별 다양한 오차

솔버	평균 제곱 오차 (MSE)	평균 절대 오차 (MAE)	상대 오차 [%]
FDM	0.0023	0.0174	0.98
FEM	0.0019	0.0150	0.75
DeepONet	0.0021	0.0162	0.83

표 1에서 보듯이, DeepONet은 FDM 및 FEM과 유사한 정확도를 보인다. 평균 제곱 오차(MSE) 측면에서 DeepONet은 0.0021의 MSE를 기록하여 유한차분법(0.0023)과 유한요소법(0.0019)에 근접한 오차를 보여주고 있다. 이는 DeepONet이 높은 정밀도로 포아송 방정식의 해를 근사할 수 있음을 나타낸다.

3.2 계산 효율성

정확도 외에도 DeepONet의 계산 효율성을 평가하여, 추론 시간 및 메모리 사용량을 특히 중점으로 두고 FDM 및 FEM과 비교하였다. 표 2는 각 방법의 훈련 시간과 추론 시간, 메모리 사용량을 요약한 것이다.

[표 2] 각 해법별 계산 시간과 메모리 사용량

솔버	학습 시간 [s]	추론 시간 [s]	메모리 사용량 [MB]
FDM	-	12.5	150
FEM	-	11.2	130
DeepONet	240	0.3	80

표 2에 따르면, DeepONet의 초기 훈련 단계는 240초가 소요되었으나, 추론 시간은 평균 약 0.3초로 상당히 빨랐다. 이에 반해 유한차분법과 유한요소법은 각각 약 12.5초 및 11.2초가 소요되었습니다. 이러한 추론 시간의 큰 감소는 실시간 시뮬레이션 및 제어 시스템과 같은 빠른 해 생성이 필요한 응용에서 DeepONet 적용이 매우 적합하다고 할 수 있다.

3.3 확장성 분석

확장성은 특히 고차원 또는 대규모 문제에 적용될 해결기에서 중요한 성능 평가지표이다. DeepONet의 확장성을 평가하기 위해 도메인 크기와 해상도 증가에 따른 계산 비용의 변화를 측정하였다. 비교를 위해 FDM 및 FEM에서도 유사한 테스트를 수행하였다. 결과는 아래 표에 요약하였다.

[표 3] 각 해법별 확장성 실험 결과

격자 해상도(N×N)	FDM [s]	FEM [s]	DeepONet [s]
32 × 32	5.6	4.8	0.3
64 × 64	12.5	11.2	0.3
128 × 128	25.4	23.5	0.3
256 × 256	53.9	50.8	0.3

표 3은 FDM 및 FEM의 계산 시간이 격자 해상도 증가에 따라 상당히 증가하는 반면, DeepONet은 모든 해상도에서 거의 일정한 추론 시간(0.3초)을 유지함을 보여준다. 일정한 추론 시간은 DeepONet의 확장성을 입증하며, 도메인 해상도 또는 차원이 가변적인 응용에 매우 유용하게 활용할 수 있음을 시사한다. 이에 반해 전통적인 방법은 해상도가 두 배로 늘어날 때 계산 시간이 거의 선형적으로 증가하여, 문제 크기가 커질수록 확장성에 제약이 따름을 보여준다.

4. 토론

본 연구는 DeepONet이 포아송 방정식을 푸는 데 매우 유망한 도구임을 보여준다. 특히 계산 효율성이 중요한 상황에서 더 유용하다. DeepONet을 전통적인 수치 해석법(FDM 및 FEM)과 비교함으로써, 이 신경망 기반 접근 방식의 장점과 잠재적인 한계를 파악할 수 있었다.

4.1 DeepONet의 강점 및 장점

DeepONet의 핵심 강점은 특히 추론 속도 측면에서의 계산 효율성에 있다. 포아송 방정식을 푸는 전통적인 방법은 높은 정확도를 제공하지만, 전체 영역에 대한 반복 연산이 필요하여 상당한 계산 비용이 소요된다. 특히, 유한 요소법에서의 행렬 조립 및 반전 작업은 문제 크기가 커짐에 따라 더욱 비용이 많이 든다. 반면, DeepONet은 학습 후, 신경망 구조를 활용하여 실시간으로 해를 예측할 수 있어, 빠른 반응 시간이 필요한 적응형 시뮬레이션 환경이나 제어 시스템과 같은 실시간 응용에 적합하다.

4.2 한계 및 도전 과제

DeepONet은 여러 강점을 지니고 있지만, 주목할 만한 한계도 존재한다. 첫 번째 주요 문제는 학습 데이터 생성 비용이 높다는 점입니다. 정확한 결과를 얻기 위해, DeepONet은 대규모의 대표적인 데이터셋이 필요하다. 일반적으로 이러한 데이터셋은 포아송 방정식을 전통적인 방법으로 풀어 생성되므로, 이는 추론 중 얻은 계산 비용 절감을 상쇄할 수 있다. 또한, 대규모 데이터셋을 처리할 때 학습 과정 자체가 시간 소모적이고 자원 집약적일 수 있다. 이는 학습 자원이 부족하거나 신속한 배포가 필요한 경우, DeepONet의 즉각적인 활용성을 제한한다.

5. 결론

본 논문에서는 물리학, 공학, 계산 과학 등 다양한 분야에서 폭넓게 활용되는 포아송 방정식의 효율적 해석기로서의 DeepONet의 성능을 조사했다. 종합적인 성능측정 실험을 통해, DeepONet이 FDM 및 FEM과 같은 전통적인 수치 해석 방법에 비해 계산 시간을 현저히 단축하면서도 높은 정확도로 해를 근사할 수 있음을 알 수 있었다.

참고문헌

[1] M. Raissi, P. Perdikaris, and G. E. Karniadakis. “Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations”. Journal of Computational Physics, 378:686–707, 2019.
[2] K.O’Shea and R. Nash. “An Introduction to Convolutional Neural Networks”, arXiv:1511.08458v2. 2015.
[3] S. Goswami, A. Bora, Y. Yu, and G.E.Karniadakis. “Physics-Informed Deep Neural Operator Networks”, ArXiv:2207.05748v2. 2022.
[4] L. Lu, P. Jin, and G.E. Karniadakis, “DeepONet: Learning nonlinear operators for identifying differential equations based on the universal approximation theorem of operators“, arXiv:1910.03193v3, 2020.
[5] Y. Yang,“DeepONet for Solving PDEs: Generalization Analysis in Sobolev Training”, ArXiv:2410.04344v1, 2024.