

공학석사학위논문

물리 기반 신경망을 이용한
탄성체의 거동 해석

Analysis of Continuous Elastic Structure
Using Physics Informed Neural Networks

2022년 2월

서울대학교 대학원
생태조경·지역시스템공학부
지역시스템공학전공
정 영 준

물리 기반 신경망을 이용한 탄성체의 거동 해석

Analysis of Continuous Elastic Structure
Using Physics Informed Neural Networks

지도교수 최 원

이 논문을 공학석사 학위논문으로 제출함
2022년 1월

서울대학교 대학원
생태조경·지역시스템공학부 지역시스템공학전공
정 영 준

정 영 준의 공학석사 학위논문을 인준함
2022년 1월

위 원 장 _____ (인)

부위원장 _____ (인)

위 원 _____ (인)

국문초록

격자 생성이 어려울 때 발생하는 FEM의 연산 부하와 연산 시간이 크게 증가하여 반복적 해석을 어렵게 만드는 단점을 극복하기 위하여, FEM보다 빠르게 근사해를 제공하는 대체 모델 (Surrogate Model)이 주목받아 왔다. 대부분의 대체 모델은 지도학습 알고리즘으로 학습되며, 최근 DNN 기반으로 학습되는 경우가 많았는데, 이 방법은 많은 양의 라벨링 된 데이터가 필요하다는 단점이 있다. 그런데 충분한 양의 라벨링된 데이터를 획득할 수 없는 경우에 대하여, 물리 기반 신경망 (PINN)의 필요성이 대두되고 있다. PINN은 DNN의 일종으로, 학습 시 경계조건, 초기조건, 그리고 지배방정식을 학습하여 미분방정식의 근사해를 구하는 방법으로, 라벨링된 데이터를 필요로 하지 않아 스몰 데이터 문제에서 대체 모델을 학습하는 데에 적합하다고 평가되었다. 그러나 PINN은 모든 구간에서 미분 가능한 해를 제공하기 때문에 미분 불가능한 해를 정확히 근사하지 못하는 단점이 있었다. 본 연구에서는 그동안 PINN이 충분히 다루지 않았던 구조역학 문제에 대한 대체 모델을 학습하기 위하여 다양한 구조 문제를 PINN으로 해석하였으며, 이때 일반적인 PINN만으로 해석할 수 없는 문제를 해결하기 위하여 연산 영역 분할을 적용한 새로운 형태의 PINN인 piecewise PINN을 제안하고 이를 적용하였다. 그리고 일반적인 PINN으로도 해석할 수 있는 문제들에 대하여 piecewise PINN을 적용하여 두 방법의 정확도를 비교하였다. 이때 해석한 문제들은 1차원 보 문제, 2차원 평면 응력 문제, 그리고 2차원 판 문제였으며, PINN의 근사해는 FEM으로 얻은 근사해와 비교하여 NMSE (Normalized Mean Squared Error)와 R^2 를 계산하여 정확도를 평가하였다. 그 결과, 1차원 단순보, 캔틸레버보나 사방이 지지된 평면 응력 및 판 문제 등의 문제에 대해서는 일반적인 PINN으로도 정확한 해를 얻을 수 있었으나, 집중하중이 작용하는 1차원 연속보, 중간에 추가적인 지지 구간이 발생하는 평면 응력과 판 문제에 대해서는 일반적인 PINN으로 정확한 해를 얻을 수 없어 piecewise PINN을 적용하였고, 그 결과 piecewise PINN이 일반 PINN에 비하여 훨씬 정확한 해를 제공하는 것으로 드러났다. 또한, 일반 PINN으로도 충분히 정확한 해를 얻을 수 있었던 문제에서 piecewise PINN을 적용한 결과 일반 PINN이 더 정확도가 높은 것으로 드러났다. 마지막으로, piecewise PINN을 이용하여 중앙부가 고정된 2차원 평면 응력 문제에 대한 대체 모델을 학습하였으며, 해당 모델의 정확도를 평가하기 위하여 FEM을 통해 얻은 라벨링된 데이터로 학습한 DNN 기반의 대체 모델과 비교하였다. 결과적으로 piecewise PINN을 통해 얻은 대체 모델이 DNN 기반의 대체 모델보다 거의 모든 경우에서 우수한 성능을 보였으며, 이를 통해 piecewise PINN이 대체 모델을 학습할 때 DNN과는 반대로 라벨링된 데이터를 필요로 하지 않을 뿐만 아니라 DNN과 비교했을 때 학습 데이터 수가 더 적어도 대개 더 우수한 성능

을 보인다는 결론을 얻을 수 있었다.

주요어 : 수치해석, 대체 모델, 구조 역학, 물리 기반 신경망 (PINN), piecewise PINN, 연산 영역 분할

학 번 : 2020-29376

목 차

국 문 초 록	i
목 차	iii
List of Tables	v
List of Figures	vi
제 1 장 서론	1
1.1 연구배경 및 필요성	1
1.2 연구목적	5
제 2 장 연구사	7
2.1 PINN	7
2.2 영역 분할을 적용한 PINN	9
2.3 대체 모델	10
제 3 장 재료 및 방법	13
3.1 탄성체의 거동 문제	13
3.1.1 1차원 보 (1-D beam)	13
3.1.2 2차원 평면 응력 (2-D plane stress)	15
3.1.3 2차원 판 (2-D plate)	17
3.2 인공신경망을 이용한 미분방정식의 해석	18
3.2.1 인공신경망	18
3.2.2 PINN	22
3.2.3 Piecewise PINN	25
제 4 장 결과 및 고찰	30
4.1 탄성체의 거동 해석: 일반 PINN	30
4.1.1 개요	30
4.1.2 1차원 단순보 문제의 해석	30
4.1.3 1차원 캔틸리버보 문제의 해석	33
4.1.4 2차원 평면 응력 문제의 해석	36
4.1.5 2차원 판 (2-D plate) 문제의 해석	40

4.2 탄성체의 거동 해석: Piecewise PINN	42
4.2.1 개요	42
4.2.2 1차원 연속보 문제의 해석	43
4.2.3 중앙부가 고정된 2차원 평면 응력 문제의 해석	47
4.2.4 중앙부가 고정된 2차원 판 문제의 해석	53
4.3 PINN과 piecewise PINN의 성능 비교	57
4.4 Piecewise PINN을 활용한 대체 모델 개발	61
 제 5 장 요약 및 결론	 71
 참고문헌	 75
Abstract	81

List of Tables

Table 4.1 Normalized mean squared error and R-squared score of piecewise PINN and the general PINN in 1-D continuous beam problem	47
Table 4.2 Normalized mean squared error and and R-squared score of piecewise PINN and the general PINN in 2-D center-fixed plane-stress problem	53
Table 4.3 Normalized mean squared error and R-squared score of piecewise PINN and the general PINN in 2-D center-supported plate problem	57
Table 4.4 Normalized mean squared error and R-squared score of the general PINN and the piecewise PINNs with 2 and 3 subdomains in 1-D continuous beam problem	58
Table 4.5 Normalized mean squared error and R-squared score of the general PINN and the piecewise PINNs with 2 subdomains in 2-D center-fixed plane stress problem	60
Table 4.6 Normalized mean squared error of various surrogate models in training cases, trained using DNN-1, DNN-2, DNN-3, and piecewise PINN	69
Table 4.7 Normalized mean squared error of various surrogate models in test cases, trained using DNN-1, DNN-2, DNN-3, and piecewise PINN	69
Table 4.8 R-squared score of various surrogate models in training cases, trained using DNN-1, DNN-2, DNN-3, and piecewise PINN	70
Table 4.9 R-squared score of various surrogate models in test cases, trained using DNN-1, DNN-2, DNN-3, and piecewise PINN	70

List of Figures

Fig. 1.1 A flow diagram of the study	5
Fig. 3.1 A deflection curve of the beam	14
Fig. 3.2 The backpropagation of error in neural networks	19
Fig. 3.3 The feedforward process in neural networks	22
Fig. 3.4 Schematic of PINN solving 2-dimensional stationary problems	24
Fig. 3.5 Sampling of the training data of PINN from boundary conditions and computational domain	25
Fig. 3.6 Sampling of the training data of piecewise PINN from an interface between two subdomains	27
Fig. 3.7 An example of domain decomposition to train piecewise PINN for 1-D continuous beam problem	29
Fig. 4.1 Problem setup and boundary conditions of simple beam under two different distributed load	31
Fig. 4.2 A comparison between the PINN and FEM solutions of 1-D simple beam problem	33
Fig. 4.3 A history of boundary, PDE, and total loss during training for 1-D simple beam problem	33
Fig. 4.4 Problem setup and boundary conditions of cantilever beam under two different distributed load	34
Fig. 4.5 A comparison between the PINN and FEM solutions of 1-D cantilever beam problem	35
Fig. 4.6 A history of boundary, PDE, and total loss during training for 1-D cantilever beam problem	36
Fig. 4.7 Problem setup and boundary conditions of the 2-D plane stress	37
Fig. 4.8 A comparison between the PINN and FEM solutions of 2-D	39
Fig. 4.9 A history of boundary, PDE, and total loss during training for 2-D plane-stress problem	40
Fig. 4.10 Problem setup and boundary conditions of the 2-D plate	41
Fig. 4.11 A comparison between the PINN and FEM solutions of 2-D plate problem:	42
Fig. 4.12 A history of boundary, PDE, and total loss during training for 2-D plate problem	42
Fig. 4.13 Problem setup and boundary conditions of continuous beam under a distributed load and a point load	44
Fig. 4.14 Decomposed subdomains to solve the 1-D continuous beam problem	

using piecewise PINN	45
Fig. 4.15 A comparison among the PINN, piecewise PINN and FEM solutions of 1-D continuous beam problem	46
Fig. 4.16 A history of loss during training for 1-D continuous beam problem: (a), (b), (c) are the history of boundary, PDE, interface, and total loss during training submodel 1, 2, 3, respectively. (d) is a comparison among the total loss of each submodel during training.	47
Fig. 4.17 Problem setup and boundary conditions of 2-D center-fixed plane stress fixed at $x=0$	48
Fig. 4.18 Domain and interface decomposed to solve the 2-D center-fixed plane stress problem and 2-D center-supported plate problem using piecewise PINN	50
Fig. 4.19 A comparison among the PINN, piecewise PINN and FEM solutions of 2-D center-fixed plane stress problem: (a), (b), (c) are displacement fields in x direction, and (d), (e), (f) are displacement fields in y direction. (a) and (d) are predicted by piecewise PINN, (b) and (e) are calculated by FEM, (c) and (f) are predicted by PINN	51
Fig. 4.20 A history of loss during training for 2-D center-fixed plane stress problem: (a) and (b) are the history of loss in subdomain 1 and 2, respectively.	52
Fig. 4.21 Problem setup and boundary conditions of the center-supported 2-D plate fixed at $x=0$	54
Fig. 4.22 A comparison among the PINN, piecewise PINN and FEM	55
Fig. 4.23 A history of loss during training for 2-D center-supported plate problem: (a) and (b) are the history of loss in subdomain 1 and 2, respectively.	56
Fig. 4.24 A comparison among the FEM, the general PINN, and the piecewise PINN with 2 subdomains and 3 subdomains solutions of 1-D continuous beam problem	58
Fig. 4.25 The point-wise error among the FEM, the general PINN, and the piecewise PINN with 2 subdomains and 3 subdomains solutions in 1-D continuous beam problem	59
Fig. 4.26 Piecewise PINN solution of the 2-D plane stress problem: (a) and (b) are the displacement field in x and y direction, respectively.	60
Fig. 4.27 The architecture of piecewise PINN and deep neural networks to train surrogate model of 2-D plane stress problem	62
Fig. 4.28 Performance indexes of the models in various elastic modulus: (a) shows NMSE, and (b) shows R-squared score. The red mark denotes the	

prediction results outside of training data range, and the blue one denotes the results inside of training data range 65

Fig. 4.29 The pointwise absolute error of various surrogate models when $E = 1.5$ Pa: (a) is DNN-1. (b) is DNN-2, (c) is DNN-3, (d) is piecewise PINN. · 66

Fig. 4.30 The pointwise absolute error of various surrogate models when $E = 5$ Pa: (a) is DNN-1. (b) is DNN-2, (c) is DNN-3, (d) is piecewise PINN. 67

Fig. 4.31 The pointwise absolute error of various surrogate models when $E = 12$ Pa: (a) is DNN-1. (b) is DNN-2, (c) is DNN-3, (d) is piecewise PINN. 68

제 1 장 서론

1.1 연구배경 및 필요성

물리적 현상을 해석하기 위해서는 먼저 수식으로 그 현상을 묘사할 필요가 있으며, 이것을 수학적 모델링이라고 한다. 수학적으로 모델링할 수 있는 현상들은 여러 가지가 있으며, 예를 들면 전염병의 전파, 인구 변화, 유체의 흐름, 구조체의 거동, 열 유동 등이 있다. 이런 현상들은 보통 시간에 따라 어떤 물리량이 변하는 과정을 나타내고, 이 때문에 시간에 대한 어떤 물리량의 미분이 수식에 포함되는 것이 필연적이다. 예를 들어 특정 지역의 인구 변화를 모델링하면 시간에 따른 인구의 변화량이 수식에 포함될 것이며, 유체의 흐름을 모델링하면 유체의 속도가 수식에 포함될 것이다. 따라서 자연 현상에 대한 수학적 모델링은 미분방정식으로 주로 이루어진다 (Bender, 1978).

수학적 모델링을 통해 얻은 미분방정식의 해를 구하면 비로소 임의의 시간 혹은 지점에서 어떤 물리량의 값을 미리 알 수 있으나, 실제로는 해석적인 해를 구할 수 없는 미분방정식이 많이 존재한다 (Bender, 1978). 이러한 미분방정식들의 해를 구하기 위해서는 수치해석 (Numerical Analysis) 기법을 사용하여 엄밀해에 가까운 근사해를 구해야 한다 (Chapra and Canale, 2011). 특히, 미분방정식의 독립변수가 2개 이상이고, 2개 이상의 독립변수에 대한 미분항이 포함되는 편미분방정식은 엄밀해를 구하기가 매우 어렵기에 수치해석 기법을 사용하여 근사해를 구하는 경우가 대부분이다. 편미분방정식의 근사해를 구하는 방법으로는 FDM (Finite Difference Method), FEM (Finite Element Method) 등이 주로 사용된다 (Boresi et al., 2003).

FDM이나 FEM과 같은 방법들은 연산 영역을 격자로 나누어 근사해를 구하는 방법으로, 효율적인 연산과 안정적으로 수렴하는 근사해를 구할 수 있다는 점에서 지난 50년간 다양한 분야의 문제들에 적용되어 왔다 (Raissi et al., 2019). 그러나, FEM에도 여러 문제점이 존재한다. 먼저, 난류 유동 등 복잡한 문제를 해석하거나 복잡한 형태의 연산 영역에서 문제를 해석할 경우 시간이 오래 걸리고 연산 부하가 심하게 걸릴 수 있다는 단점이 있다 (Hoffer et al., 2021; Sun et al., 2020). 특히, 이동경계조건 (Moving Boundary Condition)이나 연산 영역의 형태가 크게 변하는 등 특수한 경우에 격자를 생성하는 것이 큰 부담이 될 수 있다 (Sun et al. 2020).

또한, 격자를 구성하여 격자의 절점에 해당하는 근사해만 구할 수 있기 때문에, 연산 영역이 커질 경우에 일정한 해상도를 유지하기 위해서는 격자 개수가 늘어나야 하며, 이는 큰 연산 부하로 이어질 수 있다 (Hoffer et al. 2021).

한편, 앞서 언급한 FEM의 단점을 보완하기 위하여, 대체 모델 (Surrogate Model)이 활발히 연구되어 왔다 (Anantharaman et al., 2021). 대체 모델은 어떤 물리적 시스템에 대하여 일정 오차범위 내의 근사해를 제공하는 일종의 근사 모델로, 다양한 문제의 조건 (e.g. 탄성계수, 레이놀즈 수 등)을 학습함으로써 임의의 조건의 문제에 대하여 FEM보다 빠르게 근사해를 제공할 수 있다. 한 번 FEM으로 해석할 시 긴 시간이 소요되는 물리적 시스템에 대하여, 수많은 조건에서 반복적인 해석이 요구되는 최적화 설계, 미지 물리현상의 모수 추정, 민감도 분석 등을 FEM으로 수행할 수 없다는 한계가 있었다. 대체 모델은 FEM보다 빠르게 근사해를 제공하기 때문에, 이러한 한계를 극복하기 위한 대안으로 제시되어 왔다 (Kim et al., 2021; Zhu et al., 2019). 대체 모델은 주로 실제 실험이나 FEM 시뮬레이션을 통해서 얻을 수 있는 라벨링된 데이터를 활용하여 지도학습으로 만들어지는 경우가 많은데, 이 때문에 최근 들어서는 딥러닝을 이용한 지도학습을 통해 대체 모델을 개발하는 것이 가장 보편적인 방법이 되어가고 있다 (Zhu et al., 2019).

심층신경망 (DNN; Deep Neural Networks)을 이용하는 기계학습 알고리즘인 딥러닝은, 이미지 인식, 음성 인식 등에 성공적으로 사용되었을 뿐만 아니라, 유체역학, 재료과학, 유전공학 등 다양한 공학 및 과학 분야의 문제를 해결하는 데에도 광범위하게 적용되고 있다 (LeCun et al., 2015; Libbrecht and Noble, 2015; Rafiei and Adeli, 2017; Raissi et al., 2019). 딥러닝은 최근 물리적 시스템의 수학적 모델링과 시뮬레이션에도 적용되고 있는데, 이는 복잡한 비선형 함수를 학습할 수 있는 심층신경망의 특성 때문이다 (Rao et al., 2021). 그러나 딥러닝 기반의 대체 모델을 학습시키기 위해서는 많은 양의 라벨링된 데이터가 필요한데, 실제 실험을 통해서 충분한 양의 데이터를 얻을 수 없는 경우가 많고, 다루고자 하는 문제에 따라 FEM 등 시뮬레이션을 통해서 충분한 양의 데이터를 얻는 것이 어려울 수 있다 (Willard et al., 2020; Laubscher and Rousseau, 2021). 또한, 딥러닝은 학습 데이터 범위 내에서 내삽 (Interpolation)에는 좋은 성능을 보일지 몰라도 데이터 범위 외에서 외삽 (Extrapolation)에는 빈약한 성능을 보인다 (Purja Pun et al., 2019; Willard et al., 2020).

이러한 단점을 해결하기 위하여 심층신경망을 사용해서 물리 현상을 설명할 수 있는 지배방정식을 해석하는 방법으로 물리 기반 신경망 (Physics-informed Neural Networks; PINN)이 주목받고 있다. PINN은 심층신경망의 일종으로, 미분방정식의 경계조건, 초기조건, 그리고 지배방정식을 학습하여 미분방정식의 근사해를 구하는 방법이다. 이 방법은 전통적인 수치해석 방법과 비교했을 때, 격자를 생성하지 않아도 근사해를 구할 수 있으며, 최근 빠르게 발전하고 있는 딥러닝 기술에 힘입어 주목받고 있는 상황이다 (Raissi et al., 2019). 또한 라벨링된 데이터를 요구하지 않는다는 점에서 많은 데이터를 얻을 수 없는 이른바 “스몰 데이터” 문제에서 딥러닝보다 대체 모델을 학습시키기 좋은 방법으로 각광받고 있다 (Sun et al., 2020).

그러나 PINN 역시 모든 미분방정식을 정확하게 해석할 수 있는 것은 아니다. 특히 인공 신경망은 높은 진동수의 모드보다 낮은 진동수의 모드를 학습하는 것을 선호하는 성향이 있는데, 이를 스펙트럴 편향 (Spectral Bias)이라고 한다 (Xu et al., 2019, Rahaman et al., 2019, Basri et al., 2019, Cao et al., 2019). 스펙트럴 편향은 PINN에도 마찬가지로 나타나는 현상으로, 이 현상 때문에 복잡도가 높은 문제에 대하여 PINN의 부정확한 해석 결과를 초래한다고 알려진 바 있다 (Wang et al., 2021). 또한 하나의 인공신경망이 불연속 함수를 반드시 근사하도록 하는 학습 방법에 대해서는 알려진 바가 없기에 불연속 함수를 모의하기도 부적절하다 (Llanas et al., 2008; Ismailov, 2020). 게다가 PINN은 항상 전 영역에서 미분 가능한 근사해를 제공하므로 (Raissi et al., 2017), 미분 불가능한 해를 모의하는 것이 부정확할 수 있다.

한편, 일반적인 PINN의 연산 속도나 수렴성을 개선하기 위한 여러 시도가 있어 왔으며, 영역 분할 (Domain Decomposition)은 이러한 시도 중 하나이다. 영역 분할은 전체 연산 영역 (Computational Domain)을 소영역 (Subdomain)으로 분할하여 각 소영역에서 신경망을 각각 학습시켜 지역적인 해를 구하되, 전체 영역에서의 해에 대한 구속조건을 학습에 반영하여 전 영역을 만족하는 해를 보간해 나가는 방법이다. 영역 분할은 PINN으로 근사해를 구할 때 복잡도를 감소시켜 스펙트럴 편향의 영향을 줄이고 근사해의 정확도를 증가시키며, 연산 시간을 단축시킬 수 있는 방법으로 보고된 바 있다 (Jagtap et al., 2020; Shukla et al., 2021; Moseley et al., 2021).

1.2 연구목적

본 연구에서는 다양한 구조역학 문제를 PINN을 사용해서 해석하고, 일반적인 기존 PINN이 해석하지 못하는 문제에 대하여 영역 분할을 적용한다. 또한 기존 PINN으로 해석할 수 있는 문제에 대하여 영역 분할을 적용하였을 때의 학습 결과에 대해 비교하고 분석한다. 이러한 결과를 바탕으로 영역 분할을 적용한 PINN을 이용하여, 임의의 조건이 주어졌을 때의 물리적 시스템의 근사해를 실시간으로 제공하는 대체 모델을 학습시키고, 일반적인 심층 신경망을 통해 지도학습으로 훈련된 대체 모델과 예측 성능을 비교한다. 본 연구의 흐름은 Fig. 1.1과 같다. 먼저 일반적인 PINN을 적용하여 구조체의 거동 문제를 해석한다. 다음으로 일반적인 PINN으로 해석할 수 없는 구조체의 거동 문제에 대하여 영역 분할을 적용한 새로운 형태의 PINN을 제시하고, 이를 적용하여 해석 결과와 학습 결과를 제시한다. 이후 동일한 문제에 대하여 일반적인 PINN과 영역 분할 PINN으로 해석한 후 두 방법의 정확도에 대하여 평가한다. 마지막으로, 일반적인 PINN이 해석하지 못하는 물리적 시스템에 대한 대체 모델을 영역 분할 PINN으로 학습하고, 지도학습으로 훈련된 일반적인 심층 신경망 기반의 대체 모델과 예측 성능을 비교한다.

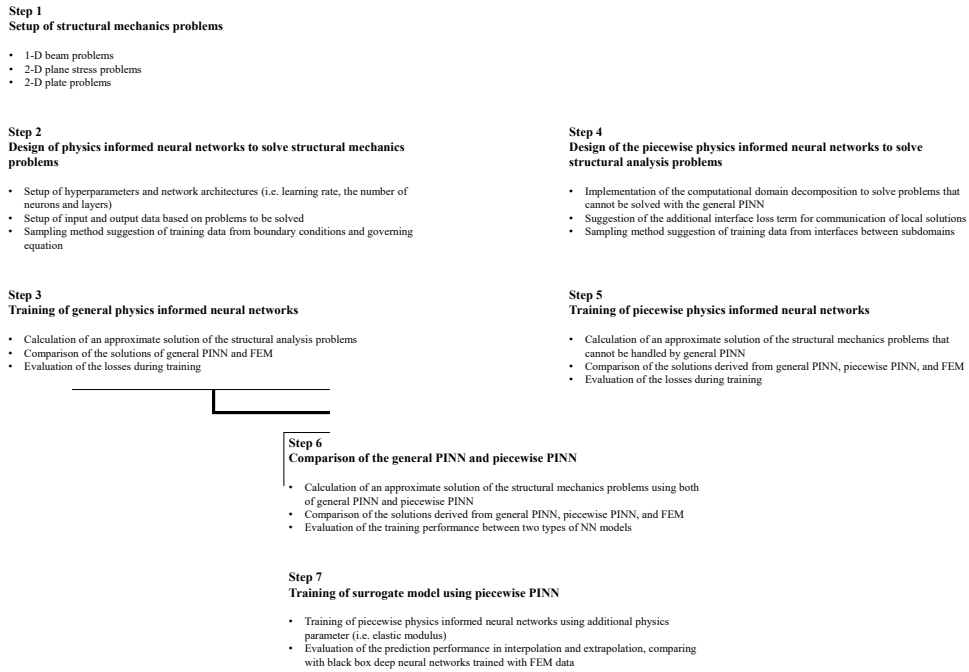


Fig. 1.1 A flow diagram of the study

제 2 장 연구사

2.1 PINN

Lagaris et al. (1998)은 인공신경망에 미분방정식의 초기조건 및 경계조건을 학습시켜 근사해를 구하는 PINN의 개념을 제시하였다. 이 연구에서는 PINN과 FEM을 통해 얻은 근사해의 정확도를 비교하였는데, 그 결과 격자의 절점에 해당하는 지점들에서는 FEM의 오차가 더 작았으나, 절점 사이의 보간점들에서는 PINN의 오차가 더 작았다고 평가되었다. 그러나 이 연구에서는 인공신경망을 학습시킬 때의 학습률이나 활성화 함수 등 다양한 조건이나 인공신경망의 층 개수, 뉴런 개수 등 구조에 대하여 고려한 바 없다는 한계점이 존재한다. 또한, PINN을 통해 미분방정식의 근사해를 구할 때 FEM에 비해 강점으로 작용할 수 있는 가능성을 제시하는데 그쳤다. 한편 다른 인공신경망 관련 연구가 그렇듯, 최근 들어 개선된 하드웨어가 개발되고 사용하기 쉬운 딥러닝 프레임워크가 공개되는 등 여건이 개선됨에 따라 개념으로만 제시되는 수준에 그쳤던 PINN 역시 Raissi et al. (2019)에 의해 최신 딥러닝 기술을 적용하여 구현되었다.

Raissi et al. (2019)은 해당 연구에서 PINN을 사용하여 순방향 문제 (Forward Problem)와 역방향 문제 (Inverse Problem)를 해결하였다. 순방향 문제는 어떤 현상에 대하여 알려진 지배방정식과 초기조건 및 경계조건이 주어져 있을 때 PINN을 사용하여 근사해를 구하는 문제를 의미하고, 역방향 문제는 반대로 어떤 현상의 지배방정식과 초기조건, 경계조건이 완전하게 알려져 있지 않은 상황에서 그 현상의 실측 데이터를 바탕으로 지배방정식, 초기조건, 경계조건을 추론하는 문제이다. Raissi et al. (2019)는 PINN을 사용하여 슈뢰딩거 방정식 (Schrodinger Equation)과 알렌-칸 방정식 (Allen-Cahn Equation)에 대하여 순방향 문제를 해결한 바 있으며, 나비에-스토크스 방정식 (Navier-Stokes Equation)과 코르테버흐-더프리스 방정식 (Korteweg-de Vries Equation)에 대하여 역방향 문제를 해결하여 알려지지 않은 지배방정식의 계수를 추정해낸 바 있다.

이후의 연구들은 PINN의 형태를 변경하고 새로운 형태를 제안하여 해석 성능을 개선하거나, 유체역학, 균열 전파 등 특정 주제의 문제의 해석에 대한 연구가 주를 이룬다. Mao et al. (2020)은 1차원 및 2차원 정상 상태에서의 오일러 방정식 (Euler

Equation)의 근사해를 PINN을 사용해 구함으로써 압축 비점성 유체의 흐름을 모의하였다. Chen et al. (2020)은 PINN을 사용해 광자 메타 물질 (Photonic Metamaterial)과 나노광학 기술에서의 산란 문제에 관련된 지배방정식의 알려지지 않은 계수를 추정하였다. Samaniego et al. (2020)은 PINN을 사용하되 함수 최적화에 에너지법을 적용하여 선형 탄성체의 거동을 모의한 바 있다. Sun et al. (2020)은 PINN을 적용하되 경계조건 강제 (Boundary Condition Enforcement)를 사용하여 나비에-스토크스 방정식을 해석하였고 이를 통하여 원형 관에서의 흐름, 혈액의 흐름 등 다양한 유체 흐름 문제를 해결한 바 있다. Goswami et al. (2020)은 위상-필드 모델 (Phase-field Model) 문제를 PINN을 사용해 해석하여 균열의 전파를 모의한 바 있으며, PINN의 전이학습 (Transfer Learning) 적용 가능성을 제시하였다. Haghighat et al. (2021)은 고체 역학 (Solid Mechanics) 문제에 PINN을 적용하여 2차원 구조체의 응력 및 처짐 분포를 모의한 바 있으며, Niaki et al. (2021)은 복합 재료 (Composite Material)의 열화학적 복원 (Thermochemical Curing) 문제를 PINN을 적용해 해석한 바 있다. Samaniego et al. (2020)은 에너지법을 적용한 PINN을 구현하여 선형 탄성 재료와 고탄성 재료 (Hyperelastic Material)의 처짐, 위상-필드 모델을 이용한 균열 전파 양상, 그리고 판의 휨 (Plate Bending)과 같은 다양한 구조해석 문제를 해석하고 결과를 제시한 바 있다. Haghighat et al. (2021)은 Raissi et al. (2019)이 제시한 PINN을 사용하여 2차원 평면-응력 문제 (Plane-stress Problem)를 해석하였고 재료에서의 처짐과 응력 분포를 모의한 바 있다. 이때 각 처짐과 응력 요소를 별개의 모델에서 학습시켜 총 5개의 PINN을 학습시킨 것이 특징적이다. Rao et al. (2021)은 PINN의 구조를 변형하여 경계조건을 강제하였고 2차원 평면-응력 문제를 해석한 바 있으며, Haghighat et al. (2021)과 마찬가지로 여러 개의 PINN을 사용하여 처짐과 응력의 개별 요소를 각각 모의하였다. 이외에도 Nabian et al. (2021)은 PINN의 학습 데이터에 중요도 샘플링 (Importance Sampling)을 적용하여 2차원 평면 응력 문제의 수렴 속도를 개선한 바 있다.

2.2 영역 분할을 적용한 PINN

PINN의 수렴 속도를 높이고 최적화 과정의 복잡도를 낮추기 위하여 영역 분할을 적용한 PINN이 연구되고 있다 (Jagtap et al. 2020; Heinlein et al., 2021). PINN

에 영역 분할을 적용하여 소영역들로 전체 영역의 근사해를 구하는 과정을 학습시킬 때 유의해야 할 점은, 대부분의 해가 연속적이기에 각 인접한 소영역들의 경계인 인터페이스에서 근사해의 값의 연속성을 보장해야 하며, 동시에 전체 영역에서의 지배방정식, 경계조건 및 초기조건 역시 만족해야 한다는 점이다.

Jagtap et al. (2020)은 cPINN (Conservative Physics-informed Neural Network)을 제시하였으며, 영역을 분할하여 각 소영역에 개별적인 PINN을 배정하고, 소영역 간의 인터페이스에서 좌우 소영역의 플럭스 연속성을 보장하는 오차항을 손실함수로 포함시켜 각 소영역의 PINN을 학습시켜 전체 영역의 근사해를 모의하도록 하였다. 이후 Jagtap and Karniadakis (2020)은 cPINN에서 더 발전된 형태인 XPINN (Extended Physics-informed Neural Networks)을 제시하였다. XPINN은 cPINN이 아무 편미분방정식이나 적용될 수 없다는 한계를 극복하여 모든 종류의 편미분방정식에 적용될 수 있으며, 영역 분할에 제약조건이 있는 cPINN과 달리 시간과 공간 모두에서 임의의 형태로 분할할 수 있다는 이점이 존재한다. Shukla et al. (2021)은 cPINN과 XPINN을 다수의 CPU와 GPU로 분산시켜 동시에 학습시키는 접근법을 보여주었으며, 단일 CPU 혹은 GPU에서 학습시킨 cPINN 및 XPINN보다 연산 속도가 증가함을 보고한 바 있다.

이외에도 Kharazmi et al. (2021)은 변분법 (Variational Formulation)과 페트로프-갈러킨 방법을 적용하여 연산 영역을 분할한 hp-VPINN (hp-Variational Physics-informed Neural Network)을 제안하였고, 다양한 문제에서 그 적용 결과를 제시하였다. Moseley et al. (2021)은 연산 영역이 큰 문제에서 발생하는 스펙트럴 편향을 해소하기 위한 FBPINN (Finite Basis Physics Informed Neural Network)을 제안하였다. 이는 기존 방법과 다르게 영역을 겹치는 영역이 있게 분할하며, 이때 발생하는 인터페이스 영역에서 윈도우 함수를 적용하여 cPINN이나 XPINN 방법을 적용했을 때 소영역의 근사해 간의 오차로 인해 발생하는 불연속성 문제를 해결하고 전체 영역에서 연속성을 보장하였다.

2.3 대체 모델

대체 모델은 어떤 문제에 대한 실제 해가 구하기 어려울 경우에 그에 근사한 해를 구하는 모델로서, 본 연구에서는 FEM에 대한 대체 모델만을 다루었다.

최근에는 심층 신경망을 사용하는 딥러닝 기반의 대체 모델이 많이 사용되고 있

다. Abueidda et al. (2021)은 FEM 시뮬레이션을 통해 획득한 데이터를 바탕으로 RNN (Recurrent Neural Networks)을 학습시켜 소성체 재료의 거동을 예측하는 대체 모델을 개발한 바 있다. Mai et al. (2021)은 DNN을 사용하여 비선형적 형상을 가지는 트러스 구조물의 최적화에 관련된 대체 모델을 학습하였고, FEM보다 연산 부하가 작을뿐더러 수렴성도 더 좋았다고 보고하였다. Liang et al. (2018)은 FEM 시뮬레이션 데이터를 바탕으로 DNN을 학습시켜 대동맥의 응력 분포를 모의하는 대체 모델을 개발하였으며, 해당 모델은 폰-미세스 응력에 대한 최대 오차가 0.891% 정도로 우수한 성능을 보여주었다. Abueidda et al. (2019)는 마찬가지로 FEM으로 획득한 데이터를 바탕으로 CNN (Convolutional Neural Networks)을 학습시켜 2차원 복합 재료의 구조적 성질을 추정하는 모델을 제안한 바 있다. Lee et al. (2021)은 센서에서 획득한 데이터를 바탕으로 DNN을 학습시켜 트러스 구조물의 손상 정도를 정량적으로 평가하는 대체 모델을 개발하였으며, 센서에서 발생하는 노이즈가 포함된 데이터에 대해서도 훌륭하게 손상 정도를 예측한다고 보고한 바 있다.

한편, 딥러닝 기반의 대체 모델을 학습시키기 위해서는 많은 양의 라벨링된 데이터가 필요하다는 한계를 극복하기 위하여, “스몰 데이터” 문제에서 주로 PINN을 이용한 대체 모델이 각광받고 있다. Haghighat et al. (2021)은 2차원 평면 응력 문제에 대하여 탄성계수와 항복 응력에 대하여 학습시킨 PINN 기반 대체 모델을 민감도 분석에 적용하였다. Niaki et al. (2021)은 재료의 열전달계수에 따른 열화학적 회복 과정을 모의하는 대체 모델을 PINN으로 학습시켰고 이를 이용하여 실시간으로 해석 결과를 나타냄으로써 FEM보다 연산 시간에서 우위를 점하였다. Pun et al. (2019)는 원자의 포텐셜 문제를 PINN을 통하여 학습시킨 대체 모델로 나타낸 바 있으며, 그 결과 해당 문제에서 PINN이 내삽은 물론 외삽에서도 일반적인 심층 신경망보다 더 좋은 성능을 보였음을 제시하였다. Sun et al. (2020)은 원형 관 및 혈관에서의 유체의 흐름을 PINN을 통해 모의하였으며, 다양한 레이놀즈 수 (Reynolds Number)에 대하여 대체 모델을 학습시켜 임의의 레이놀즈 수가 주어졌을 때 유체의 흐름을 예측하였다.

제 3 장 재료 및 방법

3.1 탄성체의 거동 문제

3.1.1 1차원 보 (1-D beam)

보의 두께가 보의 경간 길이보다 아주 작은 얇은 보가 있다고 가정하자. 대부분의 얇은 보의 처짐은 매우 작으며 따라서 보의 회전각 역시 매우 작다. Fig. 3.1에서 볼 수 있듯이 보의 회전각 θ 가 매우 작으면 ds 와 dx 가 같다고 가정할 수 있고, 또 θ 와 $\tan\theta$ 가 같다고 가정할 수 있기에 $\theta \approx \tan\theta = \frac{dv}{dx}$ 인 관계가 성립한다. 한편,

보의 곡률은 $\frac{d\theta}{dx}$ 와 같으며 이는 $\frac{d^2v}{dx^2}$ 와 같다. 만약 보의 재료가 선형 탄성적이라고 가정하면 보의 거동은 후크 법칙 (Hooke's Law)를 따르게 되고, 이때 보의 곡률은 $\frac{M}{EI}$ 와 같게 된다 (Goodno and Gere, 2017). 이때 M 은 보의 굽힘모멘트이며, EI 는 보의 굽힘강성을 의미한다. 이를 통해 식 (3.1)과 같이 굽힘모멘트와 보의 처짐에 관한 미분방정식을 얻을 수 있으며, 식 (3.1)의 전단력, 굽힘모멘트, 분포하중에 대한 관계로부터 처짐과 분포하중에 관한 미분방정식인 식 (3.2)를 얻을 수 있다.

Fig. 3.1 A deflection curve of the beam

$$\begin{aligned}\frac{d^2u}{dx^2} &= \frac{M}{EI} \\ \frac{dM}{dx} &= V \\ \frac{dV}{dx} &= w\end{aligned}\tag{3.1}$$

$$EI \frac{\partial^4 u}{\partial x^4} - w = 0\tag{3.2}$$

where, u is a displacement of the beam, M is a bending moment of the beam, E is Young's modulus, I is a moment of inertia, V is a shear force, w is a distributed load.

따라서 1차원 보에 임의의 하중이 작용할 때 임의 지점에서의 처짐에 대한 지배 방정식은 식 (3.2)와 같다. 이를 풀기 위해서는 경계조건이 필요하며, 경계조건은 지지점의 종류에 따라 달라지게 된다. 예를 들어 단순보의 양 끝 지지점 같은 경우에는 처짐과 굽힘모멘트가 0이므로 식 (3.3)과 같은 경계조건이 주어지고, 연속보에서

양 끝 지지점이 아닌 가운데에 있는 지지점의 경우에는 단순히 처짐만 0이므로 식 (3.4)와 같은 경계조건이 주어진다. 캔틸리버보의 고정 지지점의 경우에는 처짐과 기울기가 0이므로 식 (3.5)와 같이 경계조건이 설정되고, 자유단의 경우에는 굽힘모멘트와 전단력이 0이므로 식 (3.6)과 같은 경계조건이 설정된다.

$$u(a) = u_{xx}(a) = 0 \quad (3.3)$$

$$u(a) = 0 \quad (3.4)$$

$$u(a) = u_x(a) = 0 \quad (3.5)$$

$$u_{xx}(a) = u_{xxx}(a) = 0 \quad (3.6)$$

where, a is a boundary point, $u(a)$ is a displacement of the beam at $x=a$, and u_x, u_{xx}, u_{xxx} are the first, second, and third derivatives of u .

3.1.2 2차원 평면 응력 (2-D plane stress)

어떤 재료의 응력 요소들이 식 (3.7)처럼 z 방향 응력 성분이 모두 0일 때 평면 응력 (Plane Stress) 상태라고 정의한다. 이때 평형 방정식 (Equilibrium Equation) 은 식 (3.8)과 같고, 변형률-변위 관계는 식 (3.9)와 같다. 재료가 선형 탄성적이라고 가정하면 후크 법칙에 의하여 식 (3.10)을 얻을 수 있으며, 이는 식 (3.10)과 같다. 이때 변형률-응력 관계 (식 (3.11))와 식 (3.10)을 조합하면 변위와 체적력에 관련된 지배방정식 (식 (3.12))을 얻을 수 있다 (Chou and Pagano, 1967).

$$\begin{aligned} \sigma_x &= \sigma_x(x, y) \\ \sigma_y &= \sigma_y(x, y) \\ \tau_{xy} &= \tau_{xy}(x, y) \\ \tau_{xz} &= \tau_{yz} = \sigma_z = 0 \end{aligned} \quad (3.7)$$

$$\begin{aligned} \frac{\partial \sigma_x}{\partial x} + \frac{\partial \tau_{xy}}{\partial y} + f_x &= 0 \\ \frac{\partial \tau_{xy}}{\partial x} + \frac{\partial \sigma_y}{\partial y} + f_y &= 0 \end{aligned} \quad (3.8)$$

$$\begin{aligned}
\epsilon_x &= \frac{\partial u}{\partial x} \\
\epsilon_y &= \frac{\partial v}{\partial y} \\
\gamma_{xy} &= \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}
\end{aligned} \tag{3.9}$$

$$\begin{aligned}
\frac{\partial^2 \epsilon_x}{\partial y^2} + \frac{\partial^2 \epsilon_y}{\partial x^2} &= \frac{\partial^2 \gamma_{xy}}{\partial x \partial y} \\
\nabla^2(\sigma_x + \sigma_y) &= -(\nu + 1) \left(\frac{\partial f_x}{\partial x} + \frac{\partial f_y}{\partial y} \right)
\end{aligned} \tag{3.9}$$

$$\begin{aligned}
\epsilon_x &= \frac{1}{E}(\sigma_x - \nu \sigma_y) \\
\epsilon_y &= \frac{1}{E}(\sigma_y - \nu \sigma_x) \\
\gamma_{xy} &= \frac{1}{G} \tau_{xy} \\
G &= \frac{E}{2(1 + \nu)}
\end{aligned} \tag{3.10}$$

where, σ_x, σ_y are normal stresses in x and y direction and τ_{xy} a shearing stress, $\epsilon_x, \epsilon_y, \gamma_{xy}$ are normal strains in x and y direction and a shearing strain, f_x, f_y are body forces in x and y direction. u, v are displacements in x and y direction.

식 (3.9)와 식 (3.10)에서 변형률 항을 소거하고 식 (3.8)에 대입하면 식 (3.11)과 같은 x, y 방향 처짐과 체적력에 대한 지배방정식을 얻을 수 있다.

$$\begin{aligned}
G \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) + (\lambda + G) \frac{\partial}{\partial x} \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \right) + f_x &= 0 \\
G \left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} \right) + (\lambda + G) \frac{\partial}{\partial y} \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \right) + f_y &= 0 \\
\lambda &= \frac{\nu E}{(1 + \nu)(1 - \nu)}
\end{aligned} \tag{3.12}$$

where, ν is the Poisson's ratio and E is the Young's modulus of the structure.

3.1.3 2차원 판 (2-D plate)

3차원 물체의 z 방향 두께가 매우 작아 무시할 수 있을 때 이를 2차원 판으로 단순화할 수 있다. 2차원 판의 굽힘모멘트와 비틀림모멘트의 관계식은 식 (3.13)와 같으며, 모멘트와 하중의 관계는 식 (3.14)와 같다. 식 (3.13)와 식 (3.14)를 합치면 식 (3.15)과 같이 2차원 판에서 하중과 처짐에 관한 지배방정식을 얻을 수 있다 (Timoshenko and Woinowsky-Kreiger, 1959; Ugural, 2009).

$$\frac{\partial^2 M_x}{\partial x^2} + 2 \frac{\partial M_{xy}}{\partial x \partial y} + \frac{\partial^2 M_y}{\partial y^2} \quad (3.13)$$

$$\begin{aligned} M_x &= -D \left(\frac{\partial^2 w}{\partial x^2} + \nu \frac{\partial^2 w}{\partial y^2} \right) \\ M_y &= -D \left(\frac{\partial^2 w}{\partial y^2} + \nu \frac{\partial^2 w}{\partial x^2} \right) \\ M_{xy} &= -M_{yx} = D(1 - \nu) \frac{\partial^2 w}{\partial x \partial y} \end{aligned} \quad (3.11)$$

$$\begin{aligned} \frac{\partial^4 w}{\partial x^4} + 2 \frac{\partial^4 w}{\partial x^2 \partial y^2} + \frac{\partial^4 w}{\partial y^4} - \frac{q}{D} &= 0 \\ D &= \frac{Eh^3}{12(1 - \nu^2)} \end{aligned} \quad (3.12)$$

M_x, M_y, M_{xy} are bending moments in x, y direction and a shearing moment, D is the flexural rigidity of the plate, defined in Eq. (3.15). ν is Poisson's ratio, h is a depth of the plate, w is a displacement in z direction.

3.2 인공신경망을 이용한 미분방정식의 해석

3.2.1 인공신경망

인공신경망 (Artificial Neural Networks)은 생물의 신경망에서 영감을 받아 만들어진 학습 알고리즘으로, 패턴 인식, 예측, 최적화 등 다양한 분야에서 사용되고 있는 방법이다 (Jain et al., 1996). 인공신경망은 입력층 (Input Layer), 은닉층 (Hidden Layer), 그리고 출력층 (Output Layer)으로 구성되고, 각 층은 개개의 뉴런 (Neuron)들로 구성되어 있다. 입력층에서 들어온 데이터를 바탕으로 신경망의 출력층의 출력 데이터가 계산되면, 이를 정답 데이터와 비교하여 오차를 계산 후 오차

를 줄이는 방향으로 각 뉴런의 가중치와 편향을 업데이트한다. 이때 출력 데이터 계산 시의 방향과 반대로 업데이트가 이루어지기 때문에 이를 역전파 (Backpropagation)이라고 일컫는다 (LeCun et al., 2015; Fig. 3.2).

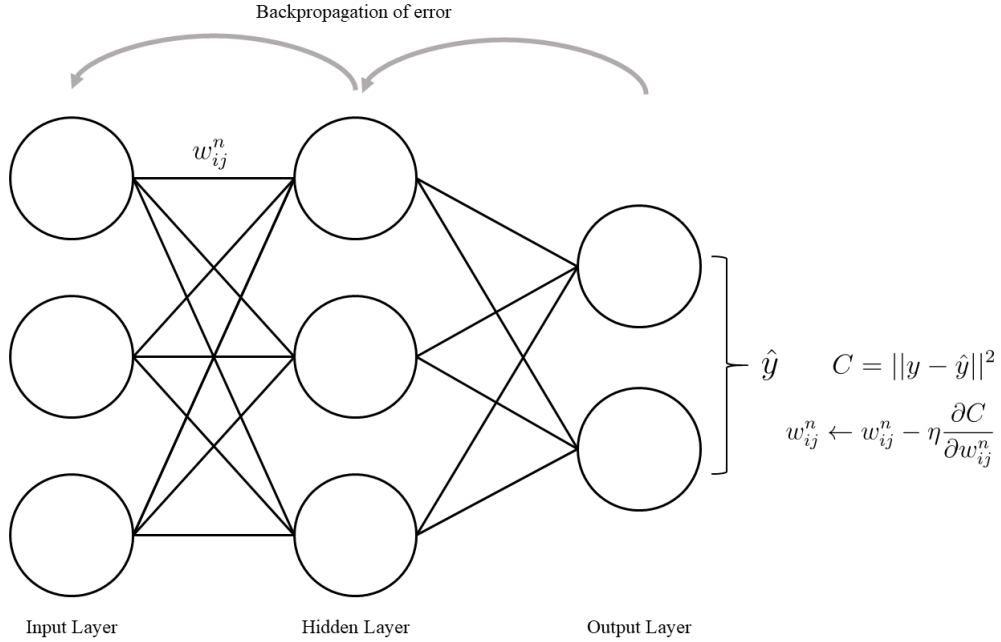


Fig. 3.2 The backpropagation of error in neural networks

각 뉴런에는 고유한 가중치 (Weight)와 편향 (Bias)가 있어, 이전 층의 뉴런으로부터 전달된 값에 가중치를 곱한 후 편향을 더해서 비선형 활성화 함수 (Nonlinear activation function)에 대입하여 다음 층으로 전달하며, 이를 순전파 (Feedforward)라고 한다 (Nielsen, 2015; Fig. 3.3). 특정 뉴런의 값은 식 (3.16)과 같이 나타낼 수 있다. l 번째 층의 j 번째 뉴런은, 직전 층인 $l-1$ 번째 층의 모든 뉴런과 가중치를 곱하고 편향을 더하여 활성화함수 σ 에 대입한 것과 같다. 이때 w_{jk}^l 은 $l-1$ 번째 층의 k 번째 뉴런과 l 번째 층의 j 번째 뉴런 사이의 가중치이며, b_j^l 은 l 번째 층의 j 번째 뉴런의 편향을 나타낸다.

$$a_j^l = \sigma(\sum_k w_{jk}^l a_k^{l-1} + b_j^l) \quad (3.16)$$

N 번째 층의 가중치 벡터를 w^N , 편향 벡터를 b^N , 신경망의 목적함수를 C 라고 하면 식 (3.17)과 같이 나타낼 수 있다. 신경망의 편향과 가중치는 경사하강법을 사용하여 최적화하게 되는데, 경사하강법은 어떤 함수의 기울기를 구하여 극값에 도달할 때까지 계속해서 기울기의 반대 방향으로 이동시키는 방법을 말한다. 경사하강법을 사용해 각 층의 편향과 가중치의 값을 목적함수의 크기를 작게 하는 방향으로 업데이트하는 과정이 식 (3.18), (3.19)에 나타나 있으며, 목적함수의 N 번째 층 편향에 대한 기울기는 식 (3.20), 가중치에 대한 기울기는 식 (3.21)과 같이 $N+1$ 번째 층의 값을 통해 계산할 수 있다. 이처럼 신경망의 출력 값을 구할 때와 반대 방향으로 출력층에서 입력층으로 차례차례 업데이트하기 때문에 이 과정을 역전파라고 말한다.

$$s^N = w^N a^{N-1} + b^N, \quad a^N = \sigma(s^N), \quad \frac{\partial C}{\partial s^N} = \delta^N \quad (3.17)$$

$$b^N \leftarrow b^N - \eta \frac{\partial C}{\partial b^N} \quad (3.18)$$

$$w^N \leftarrow w^N - \eta \frac{\partial C}{\partial w^N} \quad (3.19)$$

where, w^N and b^N are weights and biases of N th layer, σ is an activation function, C is a cost function, η is the learning rate.

$$\begin{aligned} \frac{\partial C}{\partial b^N} &= \frac{\partial C}{\partial s^N} \frac{\partial s^N}{\partial b^N} = \frac{\partial C}{\partial a^N} \frac{\partial a^N}{\partial s^N} \frac{\partial s^N}{\partial b^N} \\ &= \frac{\partial C}{\partial s^{N+1}} \frac{\partial s^{N+1}}{\partial a^N} \frac{\partial a^N}{\partial s^N} \frac{\partial s^N}{\partial b^N} = \delta^{N+1} w^{N+1} \sigma'(s^N) \end{aligned} \quad (3.20)$$

$$\begin{aligned} \frac{\partial C}{\partial w^N} &= \frac{\partial C}{\partial s^N} \frac{\partial s^N}{\partial w^N} = \frac{\partial C}{\partial a^N} \frac{\partial a^N}{\partial s^N} \frac{\partial s^N}{\partial w^N} \\ &= \frac{\partial C}{\partial s^{N+1}} \frac{\partial s^{N+1}}{\partial a^N} \frac{\partial a^N}{\partial s^N} \frac{\partial s^N}{\partial w^N} = \delta^{N+1} w^{N+1} \sigma'(s^N) a^{N-1} \end{aligned} \quad (3.21)$$

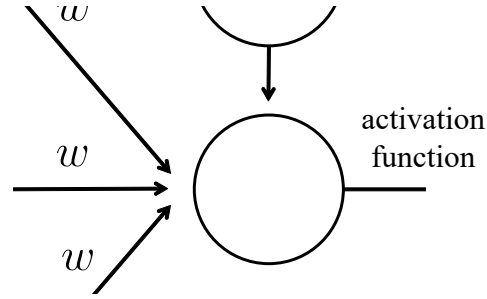


Fig. 3.3 The feedforward process in neural networks

3.2.2 PINN

PINN은 심층 신경망의 일종으로서, 다음과 같은 미분방정식 (식 (3.22); 식 (3.23))를 만족하는 해 \mathbf{u} 의 근사해 $\hat{\mathbf{u}}$ 를 구하기 위하여 사용된다.

$$N[\mathbf{u}(\mathbf{x})] = 0, \mathbf{x} \in D \quad (3.22)$$

$$B[\mathbf{u}(\mathbf{x})] = 0, \mathbf{x} \in \partial D \quad (3.23)$$

where, N is a differential operator, B is a boundary operator, D is a computational domain.

PINN이 기존의 심층 신경망과 가장 다른 점은, 학습에 있어서 식 (3.24), 식 (3.25)와 같이 물리현상을 지배하는 미분방정식을 적용할 수 있어 신경망을 통해 얻은 근사해가 지배방정식과 초기조건 및 경계조건을 만족한다는 점이다. 근사해 $\hat{\mathbf{u}}$ 를 신경망을 통하여 나타내면 식 (3.24)과 같고, 신경망의 학습을 위한 목적함수인 손실 (Loss)는 식 (3.25)와 같이 나타낼 수 있다. PINN의 손실은 경계조건에 대한 손실과 지배방정식에 대한 손실의 합으로 나타낼 수 있다.

$$\hat{\mathbf{u}} = NN(\mathbf{x}; \boldsymbol{\theta}) \quad (3.24)$$

$$L(\mathbf{x}; \boldsymbol{\theta}) = L_B + \lambda L_F = \|B[\hat{\mathbf{u}}(\mathbf{x}; \boldsymbol{\theta})]\|^2 + \lambda \|N[\hat{\mathbf{u}}(\mathbf{x}; \boldsymbol{\theta})]\|^2 \quad (3.25)$$

where, NN is a neural network, $\boldsymbol{\theta}$ is parameters of neural network (weights, biases), L_B is a loss regard to boundary conditions, L_F is a loss regard to governing differential equations, λ is a penalty parameter.

결국 PINN의 학습을 통하여 주어진 미분방정식의 근사해를 구하는 과정은 식 (3.26)와 같은 비구속 최적화 문제로 바꿀 수 있다. 손실의 값을 최소화하는 신경망의 최적 모수 $\boldsymbol{\theta}^*$ 를 신경망의 역전파를 통하여 찾아 나가게 된다. 예를 들어 2차원 정상 (Stationary) 문제를 해결하기 위한 PINN의 구성도를 Fig. 3.4과 같이 나타낼 수 있다.

$$\boldsymbol{\theta}^* = \operatorname{argmin}_{\boldsymbol{\theta}} (\|B[\hat{\mathbf{u}}(\mathbf{x}; \boldsymbol{\theta})]\|_{\partial D}^2 + \lambda \|N[\hat{\mathbf{u}}(\mathbf{x}; \boldsymbol{\theta})]\|_D^2) \quad (3.26)$$



Fig. 3.4 Schematic of PINN solving 2-dimensional stationary problems

PINN을 학습시키기 위한 데이터의 추출 방식을 Fig. 3.5에 나타내었다. 먼저 경계조건의 학습을 위하여 ∂D 에서 임의의 개수의 데이터를 추출한다. 추출한 데이터를 신경망에 입력하고 출력되는 값과 경계조건에 주어진 경계값 간의 오차를 계산한다. 다음으로 지배방정식의 학습을 위하여 D 에서 임의의 개수의 데이터를 추출

하여, 그 데이터를 신경망에 입력하고 신경망의 미분값을 계산한 뒤 지배방정식에 대입하여 오차를 계산한다.

예를 들어 식 (3.27)과 같은 미분방정식을 PINN을 통해 해석한다고 가정해보자. 이 경우 L_B , L_F 는 각각 식 (3.28), 식 (3.29)과 같이 계산된다.

$$\begin{aligned} \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} &= 0, \\ u(0, y) &= 0, \quad u(1, y) = 10, \quad u(x, 0) = 0, \quad u(x, 1) = 0 \end{aligned} \quad (3.27)$$

$$\begin{aligned} L_B &= \sum ((\hat{u}(0, y) - u(0, y))^2 + (\hat{u}(1, y) - u(1, y))^2 \\ &\quad + (\hat{u}_y(x, 0) - u_y(x, 0))^2 + (\hat{u}_y(x, 1) - u_y(x, 1))^2) \end{aligned} \quad (3.28)$$

$$L_F = \sum ((\hat{u}_{xx}(x, y) + \hat{u}_{yy}(x, y) - 0)^2) \quad (3.29)$$

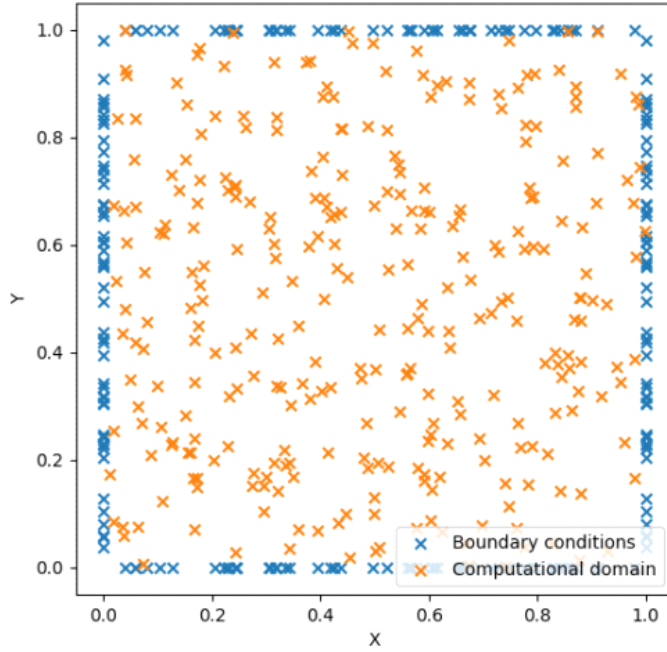


Fig. 3.5 Sampling of the training data of PINN from boundary conditions and computational domain

3.2.3 Piecewise PINN

PINN는 FDM, FEM 등 고전적인 수치해석 방법과 달리 격자망을 구성할 필요

가 없어 연산 영역의 형상이 복잡하거나 큰 경우에 이점을 발휘할 수 있으며, 격자 망에 해당하는 해만 구하는 FDM, FEM 등과 달리 전체 연산 영역에 대하여 근사해를 제공한다. 그러나 앞에서 밝혔듯 일반적인 PINN 만으로는 해석하지 못하는 문제도 더러 있으며, 본 연구에서는 그런 문제를 해결하기 위하여 piecewise PINN이라는 새로운 형태의 PINN을 제안한다.

Piecewise PINN의 학습 방식은 다음과 같다. 먼저 전체 연산 영역을 소영역 (Subdomain)들로 분해한 뒤, 각 소영역에 개별적인 PINN을 할당한다. 이후 각 소영역의 경계조건과 지배방정식에 대한 손실을 구하여 PINN을 학습하며, 여기에 소영역들 간의 연속성을 담보하기 위한 손실 (Interface Loss; L_I)이 추가된다. 어떤 소영역 S 와 인접한 소영역과의 경계 (Interface)를 생각해 보면, 그 경계에서의 근사해는 소영역 S 의 PINN뿐만 아니라 인접한 소영역의 PINN으로도 계산할 수 있다. 각 소영역의 신경망으로 구한 경계에서의 근사해와 1차, 2차, 3차, ..., n 차 미분계수 간의 오차를 손실로 두고 이를 0으로 만드는 방향으로 최적화를 진행한다면 각 소영역 간의 연속성과 미분 가능성을 보장할 수 있다.

Piecewise PINN을 학습시키기 위한 데이터의 추출 방식을 Fig. 3.6에 나타내었다. 경계조건과 지배방정식의 충족을 위한 학습 방식은 PINN의 그것과 동일하다. 추가된 것은, 소영역 간의 연속성과 미분가능성을 성립시키기 위한 인터페이스 손실 (Interface Loss) L_I 이다. 이는 인접한 소영역들의 경계인 인터페이스에서 임의의 개수의 데이터를 추출하고, 그 데이터를 각 소영역의 신경망에 입력한 뒤 출력되는 값과 1차, 2차, 3차, ..., n 차 미분계수의 차이로 정의된다 (식 (3.30); 식 (3.31)). 이때 n 의 값은 해석하고자 하는 문제의 종류나 차원에 따라 달라질 수 있다.

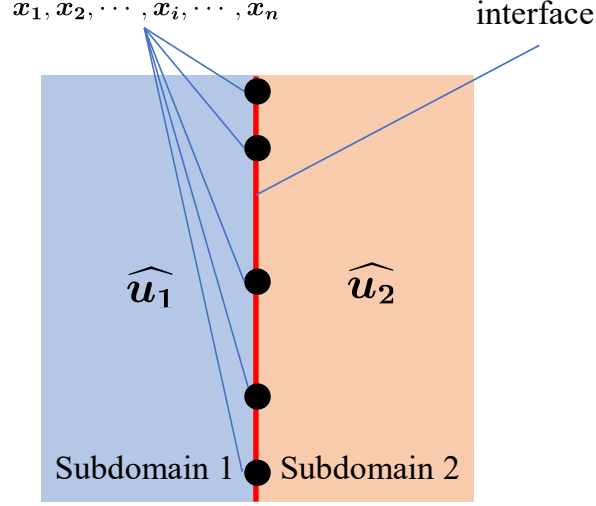


Fig. 3.6 Sampling of the training data of piecewise PINN from an interface between two subdomains

$$F[\mathbf{u}(\mathbf{x})] = (\mathbf{u}, \frac{\partial \mathbf{u}}{\partial \mathbf{x}}, \frac{\partial^2 \mathbf{u}}{\partial \mathbf{x}^2}, \frac{\partial^3 \mathbf{u}}{\partial \mathbf{x}^3}, \dots) \quad (3.30)$$

$$L_I = \sum \|F[\hat{\mathbf{u}}(\mathbf{x}^+; \boldsymbol{\theta})] - F[\hat{\mathbf{u}}(\mathbf{x}^-; \boldsymbol{\theta})]\|^2 \quad (3.31)$$

where, I is a set of interfaces, which means boundary of two adjacent subdomains, \mathbf{x}^+ and \mathbf{x}^- are the values of the argument on the right and left hand of an interface between adjacent subdomains.

함수 자체와 그 미분값들이 주어진 영역 전체에서 연속이어야 하는 경우에는 식 (3.31)과 같이 손실을 설정하여야 하나, 만약 어떤 지점을 경계로 함수값이나 그 미분값이 특정 값만큼 차이나 불연속이 되는 경우가 있을 수 있다. 그 예시로 1차원 보 문제에서 어떤 점 x 에 집중하중 P 가 주어지는 경우를 생각해 보면, 집중하중이 주어지는 점 x 좌우에서 전단력이 P 만큼 차이나게 되고 전단력은 x 에서 불연속이 된다. 이런 경우에는 L_I 를 계산할 때 경계 좌우로 함수값 또는 미분값의 차이와 P 의 차이를 계산하여 손실에 반영하였다. 따라서 식 (3.31)를 더 일반적인 경우에 대해 다시 쓰면 식 (3.32)과 같다.

$$L_I = \sum \|F[\hat{\mathbf{u}}(\mathbf{x}^+; \theta)] - F[\hat{\mathbf{u}}(\mathbf{x}^-; \theta)] - P\|^2 \quad (3.32)$$

where, P is a vector regard to the difference between \mathbf{x}^+ and \mathbf{x}^- . If the n th derivative of the neural network is continuous at the interface, the $(n+1)$ th element of P is zero, and if it is discontinuous, it has nonzero value.

Piecewise PINN을 본 연구에서 해석하고자 한 다양한 탄성체 거동 해석 문제에 적용한 방법은 다음과 같다. 먼저 1차원 연속보 문제에서 양 끝점을 제외한 추가 지지점을 경계로 소영역으로 분할하였다 (Fig. 3.7). 분할한 영역 좌우에서 함수값과 1차 미분값, 2차 미분값의 오차를 손실에 추가하여 학습을 수행하였다. 그리고 집중하중이 발생하는 경우 그것이 작용하는 위치를 경계로 소영역으로 분할하였으며, 마찬가지로 분할한 영역 좌우에서 함수값과 1차 미분값, 2차 미분값의 차이를 손실에 추가한 뒤, 3차 미분값의 차이와 집중하중 값의 오차를 손실에 추가하였다. 이때 소영역 1과 2 사이의 인터페이스 손실은 식 (3.33)와 같고, 소영역 2와 3 사이의 인터페이스 손실은 식 (3.34)와 같이 계산할 수 있다.

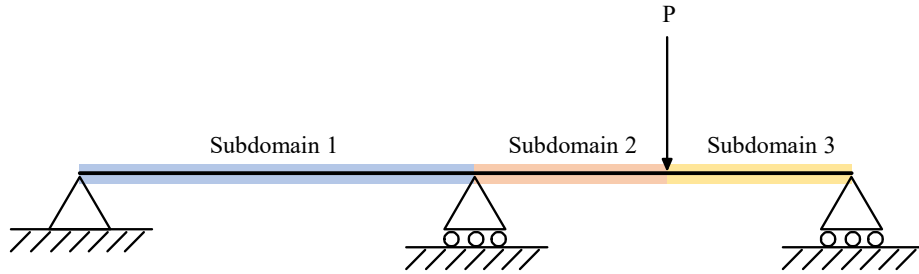


Fig. 3.7 An example of domain decomposition to train piecewise PINN for 1-D continuous beam problem

$$L_I^{1,2} = (\hat{u}^1(x) - \hat{u}^2(x))^2 + (\hat{u}_x^1(x) - \hat{u}_x^2(x))^2 + (\hat{u}_{xx}^1(x) - \hat{u}_{xx}^2(x))^2 \quad (3.33)$$

$$L_I^{2,3} = (\hat{u}^2(x) - \hat{u}^3(x))^2 + (\hat{u}_x^2(x) - \hat{u}_x^3(x))^2 + (\hat{u}_{xx}^2(x) - \hat{u}_{xx}^3(x))^2 + (\hat{u}_{xxx}^2(x) - \hat{u}_{xxx}^3(x) - P)^2 \quad (3.34)$$

제 4 장 결과 및 고찰

4.1 탄성체의 거동 해석: 일반 PINN

4.1.1 개요

가장 먼저 탄성체의 거동 문제 중 일반적인 PINN만으로 해석할 수 있는 것들을 해석하였다. 순서대로 1차원 단순보, 1차원 캔틸리버보, 2차원 평면 응력, 2차원 판 순으로 문제의 구성과 해석 및 학습 결과를 나타내었다.

모든 PINN 모델은 Python 3.9.7 상에서 Pytorch 1.9.1, CUDA 10.2, cuDNN 7.6.5, Numpy 1.20.3을 사용하여 작성되었으며, Nvidia Quadro RTX 5000을 사용하여 학습되었다. 1차원 보에 대한 근사해의 정확도를 평가하기 위하여, COMSOL Multiphysics를 이용하여 얻은 FEM 기반의 결과와 PINN을 통해 얻은 근사해 사이의 결과를 비교하였다. 이 때 PINN의 정확도를 평가하기 위해 NMSE (Normalized Mean Squared Error)와 R^2 를 사용하였으며, 그 계산 방법은 식 (4.1)과 같다.

$$\begin{aligned} NMSE &= \frac{\sqrt{\sum \|y - \hat{y}\|^2}}{\sqrt{\sum \|\hat{y}\|^2}} \\ R^2 &= 1 - \frac{\sum (\hat{y} - y)^2}{\sum (\hat{y} - \bar{y})^2} \end{aligned} \quad (4.1)$$

where, y is a result from FEM, \bar{y} is a mean of the FEM data, and \hat{y} is a prediction from PINN.

4.1.2 1차원 단순보 문제의 해석

가장 먼저 1차원 보의 처짐을 해석하기 위하여 일반 PINN을 적용하였다. 신경망의 구조는 뉴런 1개로 구성된 입력층과 마찬가지로 뉴런 1개로 구성된 출력층 그리고 뉴런 20개로 구성된 은닉층 4개로 구성하였다. 옵티마이저 (Optimizer)로는 각 파라미터마다 다른 크기의 업데이트를 적용하는 Adam 옵티마이저를 사용하였으며, 층 간의 활성화 함수로는 쌍곡탄젠트 (Hyberbolic tangent) 함수를 사용하였다. 학습률 (Learning rate)로는 0.0002를 적용하였으며 총 2000번의 에폭 (Epochs) 동안 학습을 진행하였다. 학습 데이터는 경계조건에 대하여 100개, 지배방정식에 대하여

10000개의 데이터를 추출하여 사용하였다. 경계조건을 보다 잘 만족하도록 하기 위하여, L_B 에 가중치 100을 곱하여 학습을 수행하였다.

본 연구에서 해석하고자 한 단순보 문제는 다음과 같다 (Fig. 4.1; 식 (4.2)). 지배방정식은 앞서 언급하였던 1차원 보의 처짐에 관련된 4차 미분방정식을 사용하였고, 보의 양 끝점은 각각 $x=-1$ 과 $x=1$ 이고 두 점 모두에서 지지를 받아 처짐과 휨 모멘트가 0이라고 설정하였다. 하중은 2개의 불균일한 분포하중을 설정하였다. 보 재료의 단면 2차모멘트와 탄성계수는 각각 1 m^4 , 1 Pa 이라고 가정하였다.

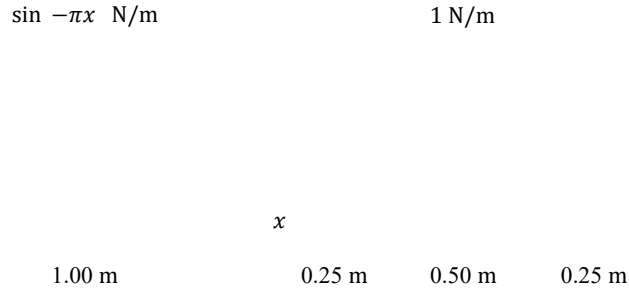


Fig. 4.1 Problem setup and boundary conditions of simple beam under two different distributed load

$$u(-1) = u(1) = 0, \quad u_{xx}(-1) = u_{xx}(1) = 0 \quad (4.2)$$

where, $u(a)$ is a displacement of the beam at $x=a$, f is a composite load, and

$$u_{xx} \text{ denotes } \frac{\partial^2 u}{\partial x^2}$$

단순보 문제의 PINN을 통해 얻은 해석 결과와 FEM을 통해 얻은 해석 결과를 Fig. 4.2에 나타내었다. 학습을 마친 후 FEM과 결과를 비교했을 때 0.0053의 NMSE과 0.9999의 R^2 를 보여 거의 차이가 없는 것으로 판단되었다. 또한 학습 과정 중 경계조건에 대한 손실, 지배방정식에 대한 손실 및 총 손실을 각각 Fig. 4.3에 도시하였다. 학습의 극초반 과정에서는 경계조건에 대한 손실이 지배방정식에 대한 손실보다 훨씬 크게 나타났지만, 이후로는 정반대의 양상이 나타나 총 손실의 대부분을 지배방정식에 대한 손실이 차지하였다.

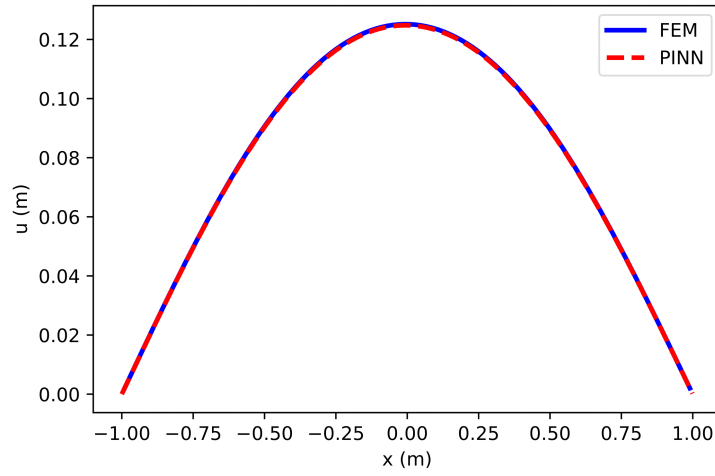


Fig. 4.2 A comparison between the PINN and FEM solutions of 1-D simple beam problem

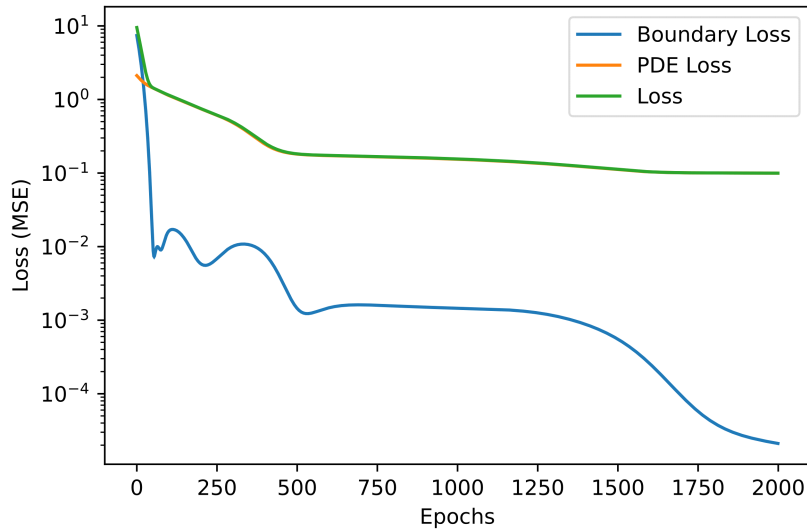


Fig. 4.3 A history of boundary, PDE, and total loss during training for 1-D simple beam problem

4.1.3 캔틸리버보 문제의 해석

본 연구에서 해석하고자 한 캔틸리버보 문제는 다음과 같다. (Fig. 4.4; 식 (4.3)) 지배방정식은 마찬가지로 1차원 보의 처짐에 관련된 4차 미분방정식을 사용하였고, 보의 양 끝점은 각각 $x=-1$ 과 $x=1$ 이고 왼쪽 끝 점에서 지지를 받는다. 왼쪽 끝 점에서는 처짐과 처짐각이 0, 오른쪽 끝 점에서는 모멘트와 전단력이 0이라고 설정하였다. 하중은 2개의 불균일한 분포하중을 설정하였다. 보 재료의 단면 2

차모멘트와 탄성계수는 각각 1 m^4 , 1 Pa 이라고 가정하였다.

캔틸리버 보 문제의 PINN을 통해 얻은 해석 결과와 FEM을 통해 얻은 해석 결과를 Fig. 4.5에 나타내었다. 학습을 마친 후 FEM과 결과를 비교했을 때 약 0.0141의 NMSE와 0.9999의 R^2 를 보여 단순보 문제보다 큰 오차를 보였지만 역시 거의 차이가 없는 것으로 판단되었다. 지또한 학습 과정 중 경계조건에 대한 손실, 지배방정식에 대한 손실 및 총 손실을 각각 Fig. 4.6에 도시하였다. 학습의 전 과정에서 지배방정식에 대한 손실이 경계조건에 대한 손실보다 크게 나타나, 총 손실의 대부분을 지배방정식에 대한 손실이 차지하였다. 또한 1000 에폭 이후로 경계조건에 대한 손실에서 지속적인 진동이 발생하였다.

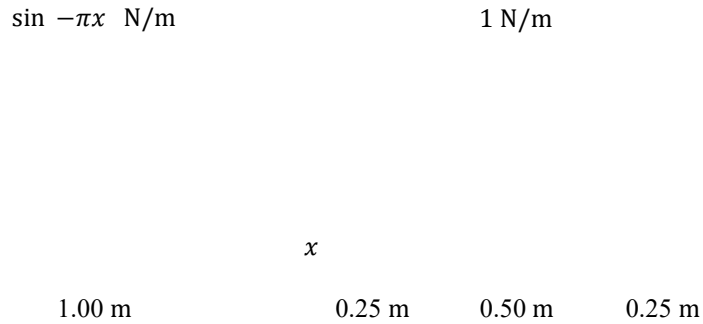


Fig. 4.4 Problem setup and boundary conditions of cantilever beam under two different distributed load

$$u(-1) = u_x(-1) = 0, \quad u_{xx}(1) = u_{xxx}(1) = 0 \quad (4.3)$$

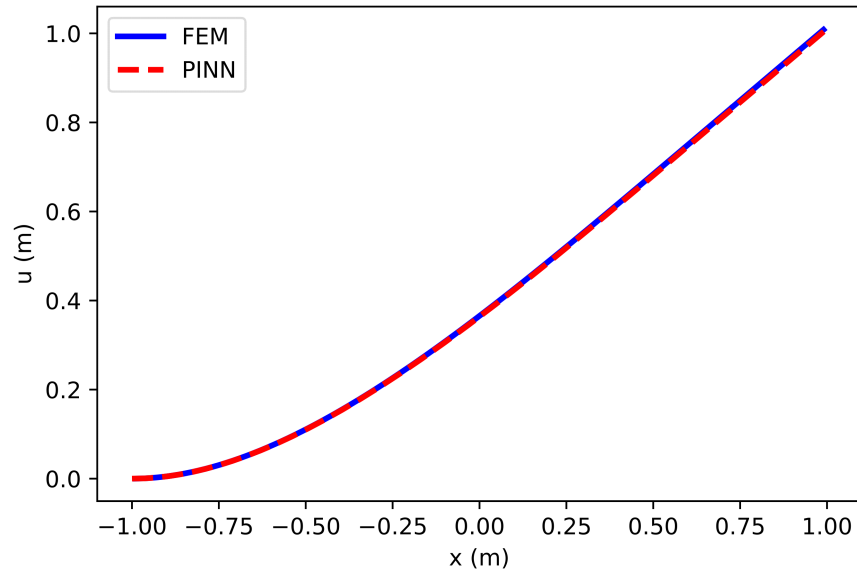


Fig. 4.5 A comparison between the PINN and FEM solutions of 1-D cantilever beam problem

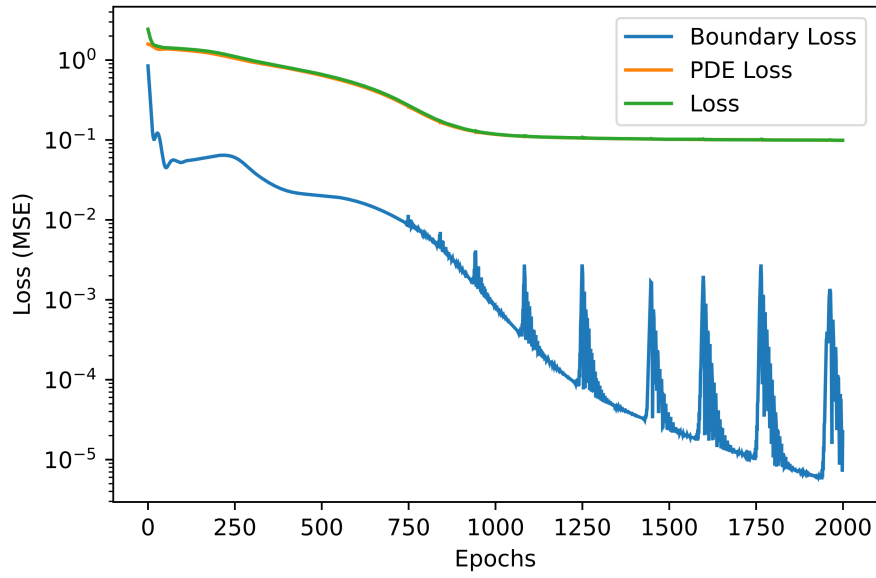


Fig. 4.6 A history of boundary, PDE, and total loss during training for 1-D cantilever beam problem

4.1.4 2차원 평면 응력 문제의 해석

앞서 서술하였던 2차원 평면 응력 문제에서 처짐에 대한 지배방정식 (식 3.10)을 적용하여, 일반 PINN을 사용해 2차원 평면 응력 문제를 해석하였다 (Fig. 4.7). x 좌표와 y 좌표 모두 -1에서 1사이에서 정의된 정사각형 형상으로, 네 변이 모두 고정되어 처짐이 발생하지 않는 조건에서 x 방향 및 y 방향의 변위장 (Displacement Field)을 계산하였다 (식 (4.4)). 이때, 체적력 (Body Force)는 다음과 같이 정의하였다 (식 (4.5)). 또한 포아송 비 ν 와 탄성계수 E 는 각각 0과 2 Pa로 설정하였다.

해석하고자 하는 문제가 1차원에서 2차원으로 변화하였기에, PINN의 형태 역시 변화하였다. 입력층은 x 와 y 2개의 뉴런으로 구성되고, 출력층은 u 와 v 2개의 뉴런으로 구성되었다. 은닉층은 1차원 문제와 마찬가지로 3개의 층으로 구성되었으며, 각 층은 20개의 뉴런으로 구성되었다. 또한, 총 40000번의 에폭 (Epochs) 동안 학습을 진행하였다. 학습 데이터는 경계조건에 대하여 100개, 지배방정식에 대하여 10000개의 데이터를 추출하여 사용하였다. 경계조건을 보다 잘 만족하도록 하기 위하여, L_B 에 가중치 100을 곱하여 학습을 수행하였으며, 이외의 다른 모든 조건은 1차원 문제를 해석할 때와 동일하게 적용하였다.

2.00

2.00

2.00

2.00

Fig. 4.7 Problem setup and boundary conditions of the 2-D plane stress

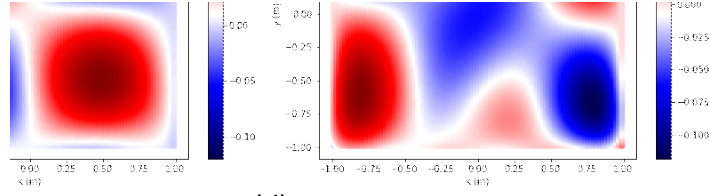
$$\begin{aligned}
u(x, -1) &= u(x, 1) = u(-1, y) = u(1, y) = 0 \\
v(x, -1) &= v(x, 1) = v(-1, y) = v(1, y) = 0
\end{aligned} \tag{4.4}$$

$$\begin{aligned}
f_x &= -10y \cos\left(\frac{\pi}{2}y\right) \sin(\pi x), \\
f_y &= -10x \sin\left(\frac{\pi}{2}y\right) \cos(\pi x)
\end{aligned} \tag{4.5}$$

PINN으로 2차원 평면 응력 문제를 해석한 결과와 FEM으로 같은 문제를 해석한 결과를 Fig. 4.8에 도시하였다. 해석 결과는 u 의 경우 0.0282의 NMSE와 0.9999의 R^2 를 가지는 것으로 드러났으며 v 의 경우 약 0.0329의 NMSE와 0.9988의 R^2 를 가지는 것으로 드러났다. 학습 중 손실의 변화를 Fig. 4.9에 제시하였다. 1차원일 때와 다르게 대부분의 학습 과정에서 경계조건에 대한 손실이 지배방정식에 대한 손실보다 더 크게 나타났다.

(a)

(b)



(c)

(d)

Fig. 4.8 A comparison between the PINN and FEM solutions of 2-D plane stress problem: (a) and (b) are displacement fields in x and y direction predicted by PINN; (c) and (d) are displacement fields in x and y direction calculated by FEM.

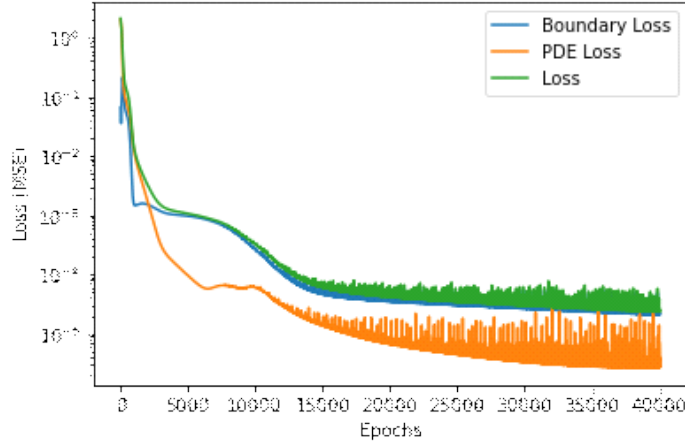


Fig. 4.9 A history of boundary, PDE, and total loss during training for 2-D plane-stress problem

4.1.5 2차원 판 문제의 해석

앞서 서술하였던 2차원 판의 지배방정식을 적용하여 (식 (3.16)), 일반 PINN을 사용해 2차원 평면 응력 문제를 해석하였다 (Fig. 4.10). x 좌표와 y 좌표 모두 -1에서 1사이에서 정의된 정사각형 형상으로, 네 변이 모두 단순 지지되어 처짐이 발생하지 않는 조건에서 판의 z 방향 처짐을 계산하였다 (식 (4.6)). 이때, 판의 z 방향으로 작용하는 분포하중은 식 (4.7) 와 같고, 판의 굽힘강성 D 는 1로 정의하였다.

PINN의 입력층은 x 와 y 2개의 뉴런으로 구성되고, 출력층은 w 1개의 뉴런으로 구성되었다. 은닉층은 1차원 문제와 마찬가지로 3개의 층으로 구성되었으며, 각 층은 40개의 뉴런으로 구성되었다. 또한, 총 40000번의 에폭 (Epochs) 동안 학습을 진행하였다. 학습 데이터는 경계조건에 대하여 100개, 지배방정식에 대하여 10000개의 데이터를 추출하여 사용하였다. 경계조건을 보다 잘 만족하도록 하기 위하여, L_B 에 가중치 100을 곱하여 학습을 수행하였으며, 이외의 다른 모든 조건은 1차원 보 문제와 2차원 평면응력 문제를 해석할 때와 동일하게 적용하였다.

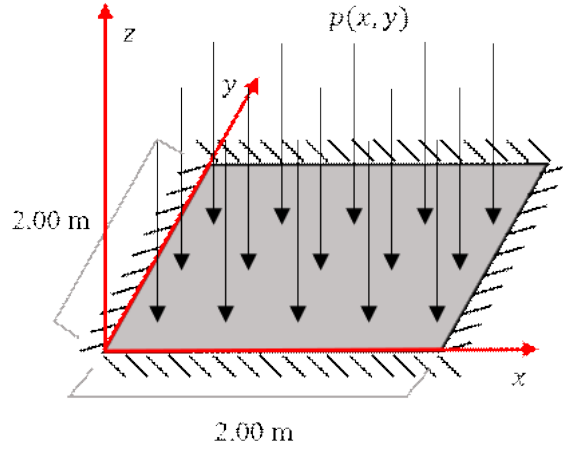


Fig. 4.10 Problem setup and boundary conditions of the 2-D plate

$$\begin{aligned}
 w(x, -1) = w(x, 1) = w(-1, y) = w(1, y) &= 0 \\
 w_{yy}(x, -1) = w_{yy}(x, 1) = w_{xx}(-1, y) = w_{xx}(1, y) &= 0
 \end{aligned} \tag{4.6}$$

$$p(x, y) = -10 \cos\left(\frac{\pi}{2}y\right) \sin(\pi x) \tag{4.7}$$

PINN으로 2차원 판 문제를 해석한 결과와 FEM으로 같은 문제를 해석한 결과를 Fig. 4.11에 도시하였다. 해석 결과는 FEM과 비교해서 0.0156의 NMSE와 0.9999의 R^2 를 보여 큰 차이가 없는 것으로 드러났다. 학습 중 손실의 변화를 Fig. 4.12에 제시하였다. 대부분의 학습 과정에서 경계조건에 대한 손실과 지배방정식에 대한 손실이 비슷한 크기로 나타나는 것을 확인할 수 있었다.

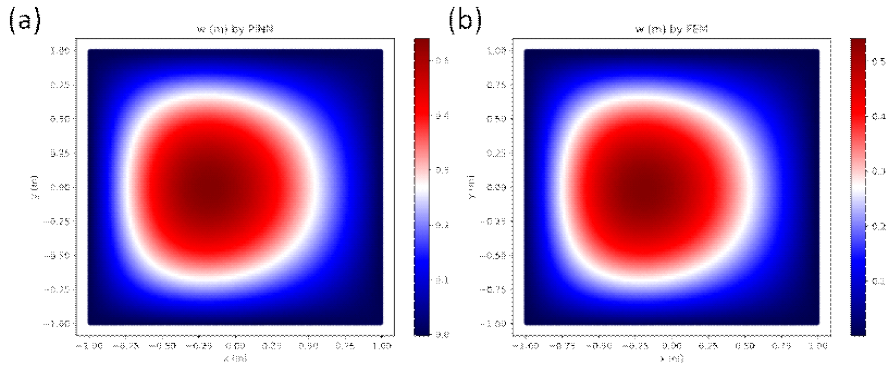


Fig. 4.11 A comparison between the PINN and FEM solutions of 2-D plate problem:
 (a) is a displacement in z direction predicted by PINN,
 (b) is a displacement in z direction calculated by FEM

Fig. 4.12 A history of boundary, PDE, and total loss during training for 2-D plate problem

4.2 탄성체의 거동 해석: Piecewise PINN

4.2.1 개요

다음으로는 일반적인 PINN만으로 정확한 해를 얻을 수 없는 문제들을 영역 분할을 적용한 piecewise PINN을 활용하여 해석하였다. 순서대로 1차원 연속보, 중앙부가 고정된 2차원 평면 응력, 중앙부가 고정된 2차원 판 문제를 해석하였다. 각 문제의 구성과 해석 및 학습 결과를 차례대로 나타내었다. 각 문제에 piecewise PINN

에서 얻은 해석 결과와 FEM에서 얻은 해석 결과를 비교하였으며, 일반 PINN으로 동일 문제를 해석했을 때의 결과도 같이 도시하여 해당 문제에 piecewise PINN을 적용해야 하는 당위성을 제시하였다.

4.2.2 연속보 문제의 해석

부정정 구조인 연속보 문제를 해결하기 위하여 Piecewise PINN를 적용하였다. 총 5000번의 에폭 동안 학습을 진행하였으며 다른 모든 학습 조건은 일반 PINN와 동일하게 설정하였다. 소영역 사이의 경계인 인터페이스에서 100개씩의 데이터를 추출하여 학습에 사용하였다.

본 연구에서 해석하고자 한 연속보 문제는 다음과 같다. (Fig. 4.13; 식 (4.8)) 지배방정식은 앞서와 마찬가지로 1차원 보의 처짐에 관련된 4차 미분방정식을 사용하였다. 보의 양 끝점은 각각 $x=-1$ 과 $x=1$ 이고 추가적으로 $x=0$ 에서 지지를 받는다는 조건을 추가하였다. 양 끝점에서는 처짐과 모멘트가 0이라고 설정하였고 중간 지지점에서는 처짐만이 0이라고 설정하였다. 하중은 1개의 불균일한 분포하중과 1개의 집중하중을 설정하였으며, 분포하중은 보 전체에 작용하고 집중하중은 $x=0.5$ 에서 작용한다고 설정하였다. 마찬가지로 보 재료의 단면 2차모멘트와 탄성계수는 각각 1 m^4 , 1 Pa 이라고 가정하였다.

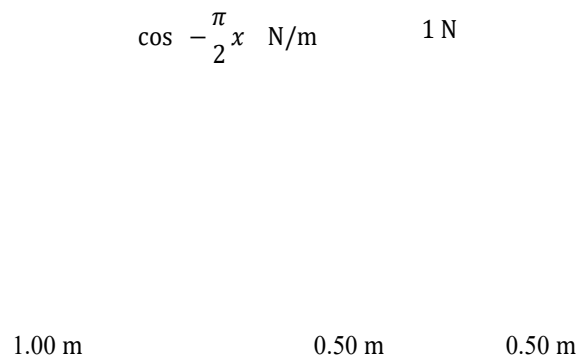


Fig. 4.13 Problem setup and boundary conditions of continuous beam under a distributed load and a point load

$$u(-1) = u(0) = u(1) = 0, \quad u_{xx}(-1) = u_{xx}(1) = 0 \quad (4.8)$$

연속보 문제는 전체 연산 영역에 대하여 한 개의 해를 구하는 일반적인 PINN과 연산 영역을 소영역으로 분할하여 각 소영역의 해를 구하는 piecewise PINN의 2가지 방법을 사용하여 해석하였다. 이때 영역의 분할 결과는 Fig. 4.14와 같았다. 우선 $x=0.5$ 에 집중하중이 가해지기 때문에 그 좌우에서 전단력의 불연속을 구현하기 위하여 해당 지점에서 영역을 분할하였고, 또 $x=0$ 에서 추가적으로 영역을 분할하여 모든 소영역이 정정 문제가 되도록 하였다.

해석 결과를 비교해보면, 일반 PINN은 1.2345의 NMSE와 0.0772의 R^2 를 가지는 것으로 나타나 연속보 문제를 잘 해석하지 못하는 것으로 드러났다. 반면 piecewise PINN을 통해 얻은 해는 FEM을 통해 얻은 해와 비교했을 때 0.0045의 NMSE와 0.9999의 R^2 를 가져 거의 차이가 없다는 것을 확인할 수 있었다. (Fig. 4.15; Table 4.1). 이를 통해 영역을 소영역으로 분할하지 않을 시 1차원 부정정 문제를 해결하지 못할뿐더러, 집중하중과 같은 조건도 구현하지 못함을 알 수 있었다.

학습 중 손실의 변화를 살펴보면, 3개의 소영역 모두에서 L_I 가 초반에 가장 크게 나타났으나 학습이 진행됨에 따라 앞서 서술한 일반 PINN에서의 학습처럼 총 손실에서 차지하는 비중은 L_F 가 가장 컸고 그 다음으로 L_I , L_B 순으로 크게 나타났다 (Fig. 4.16). 3개의 소영역에서 학습 중 손실의 변화를 비교해보면 1번 소영역의 손실이 가장 작았으며 2, 3번 소영역의 손실 변화는 유사한 양상을 보였다.

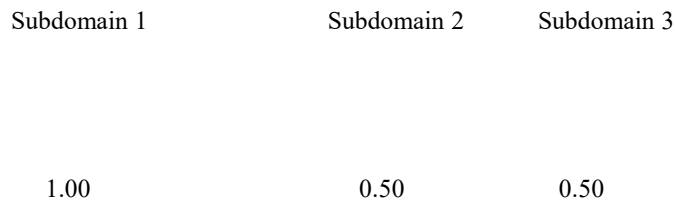


Fig. 4.14 Decomposed subdomains to solve the 1-D continuous beam problem using piecewise PINN

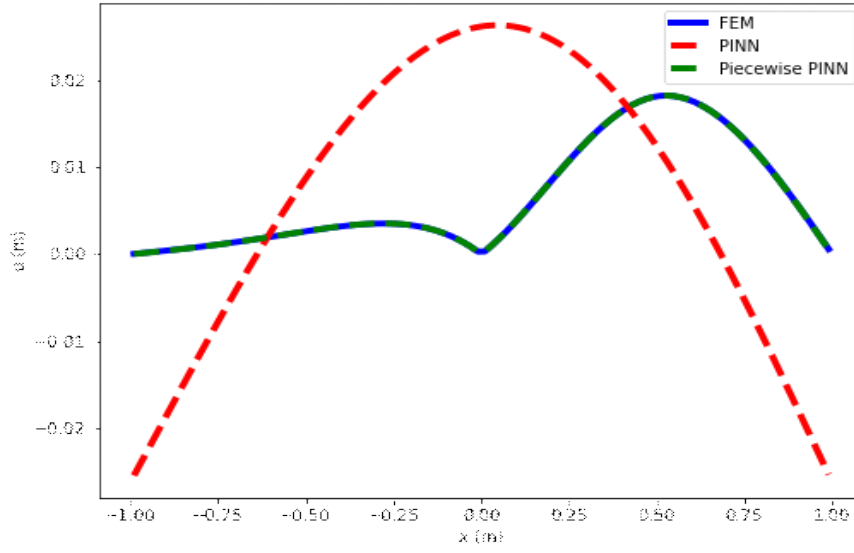


Fig. 4.15 A comparison among the PINN, piecewise PINN and FEM solutions of 1-D continuous beam problem

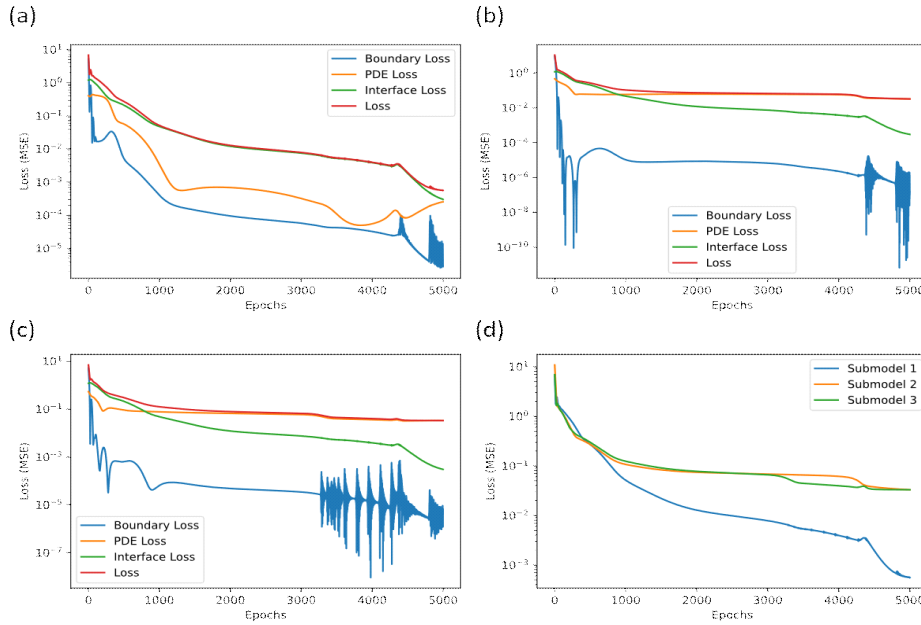


Fig. 4.16 A history of loss during training for 1-D continuous beam problem: (a), (b), (c) are the history of boundary, PDE, interface, and total loss during training submodel 1, 2, 3, respectively. (d) is a comparison among the total loss of each submodel during training.

Table 4.1 Normalized mean squared error and R-squared score of piecewise PINN and the general PINN in 1-D continuous beam problem

	Piecewise PINN	General PINN
NMSE	0.0045	1.2345
R^2	0.9999	0.0772

4.2.3 중앙부가 고정된 평면 응력 문제의 해석

중앙부가 고정된 평면 응력 문제를 해석하기 위해서 piecewise PINN를 적용하였다. 총 40000번의 에폭 동안 학습을 진행하였으며 다른 모든 학습 조건은 2차원 평면 응력 문제를 해석할 때의 일반 PINN와 동일하게 설정하였다. 소영역 사이의 경계인 인터페이스에서 100개의 데이터를 추출하여 학습에 사용하였다.

해석하고자 한 평면 응력 문제는 Fig. 4.17와 같다. x 좌표와 y 좌표 모두 -1에서 1사이에서 정의된 정사각형 형상으로, 네 변과 $x=0$ 에서 모두 고정되어 처짐이 발생하지 않는 조건에서 x 방향 및 y 방향의 변위장을 계산하였다 (식 (4.9)). 이때, 체적력은 네 변이 고정된 평면 응력 문제를 해석할 때 사용했던 것과 같이 사용하였다 (식 (4.5)). 또한 포아송 비 ν 와 탄성계수 E 는 각각 0과 2 Pa로 설정하였다.

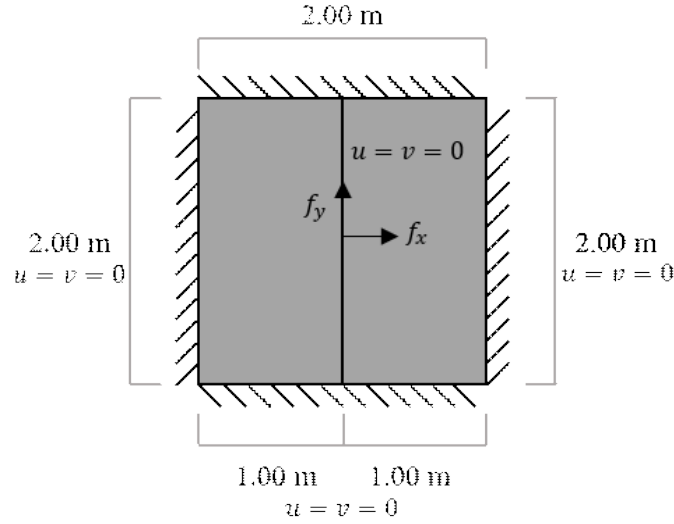


Fig. 4.17 Problem setup and boundary conditions of 2-D center-fixed plane stress fixed at $x=0$

$$\begin{aligned}
 u(x, -1) = u(x, 1) = u(-1, y) = u(0, y) = u(1, y) = 0 \\
 v(x, -1) = v(x, 1) = v(-1, y) = v(0, y) = v(1, y) = 0
 \end{aligned} \quad (4.9)$$

평면 응력 문제 역시 일반적인 PINN과 piecewise PINN의 2가지 방법을 사용하여 해석하였다. 이때 영역의 분할 결과는 Fig. 4.18와 같았다. $x=0$ 에서 추가적인 고정이 발생했기 때문에 이 지점을 경계로 2개의 직사각형 모양 소영역으로 분할하였다.

해석 결과를 비교해보면, 일반 PINN은 u 의 경우 0.1628의 NMSE와 0.8731의 R^2 를, v 의 경우 1.1402의 NMSE와 0.4236의 R^2 를 가지는 것으로 나타나 연속보 문제를 잘 해석하지 못하는 것으로 드러났으며, piecewise PINN을 통해 얻은 해는 u 의 경우 0.0672의 NMSE와 0.9954의 R^2 를, v 의 경우 0.1610의 NMSE와 0.9697의 R^2 를 가지는 것으로 나타나 일반 PINN보다 훨씬 정확도가 높은 해석 결과를 보이는 것을 확인할 수 있었다 (Fig. 4.19; Table 4.2). 이를 통해 1차원 보 문제에 이어 2차원 평면 응력 문제에서도 영역을 소영역으로 분할하지 않을 시 특정 문제를 정확하게 해석할 수 없음을 알 수 있었다.

학습 중 손실의 변화를 살펴보면, 3개의 소영역 모두에서 총 손실에서 차지하는 비중은 L_B 가 가장 컸고 그 다음으로 L_F , L_I 순으로 크게 나타났다 (Fig. 4.20). 2개

의 소영역에서 손실의 크기를 비교했을 때는 비슷한 것으로 드러났다.

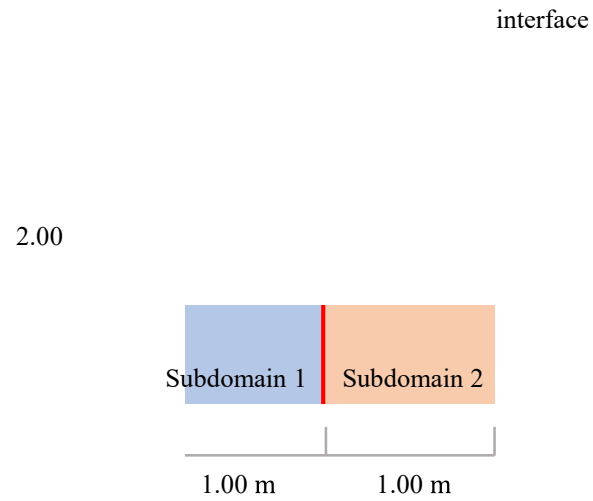


Fig. 4.18 Domain and interface decomposed to solve the 2-D center-fixed plane stress problem and 2-D center-supported plate problem using piecewise PINN

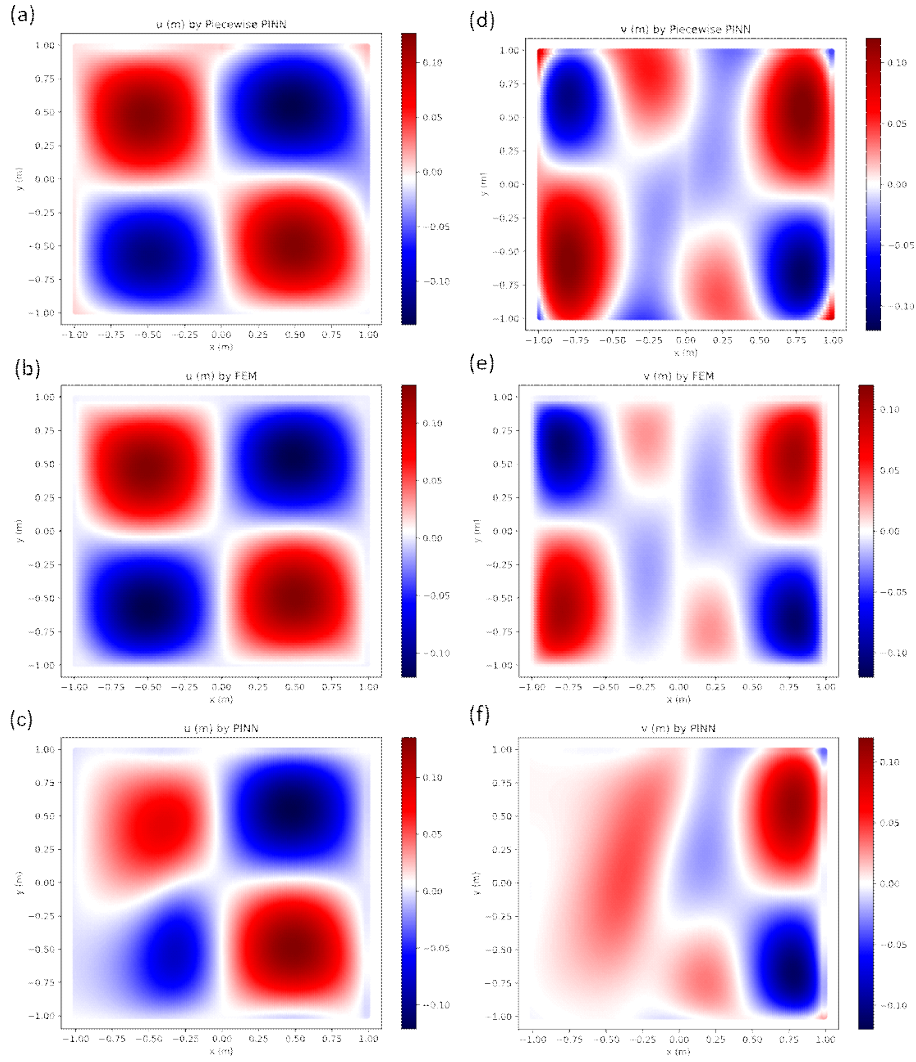


Fig. 4.19 A comparison among the PINN, piecewise PINN and FEM solutions of 2-D center-fixed plane stress problem: (a), (b), (c) are displacement fields in x direction, and (d), (e), (f) are displacement fields in y direction. (a) and (d) are predicted by piecewise PINN, (b) and (e) are calculated by FEM, (c) and (f) are predicted by PINN

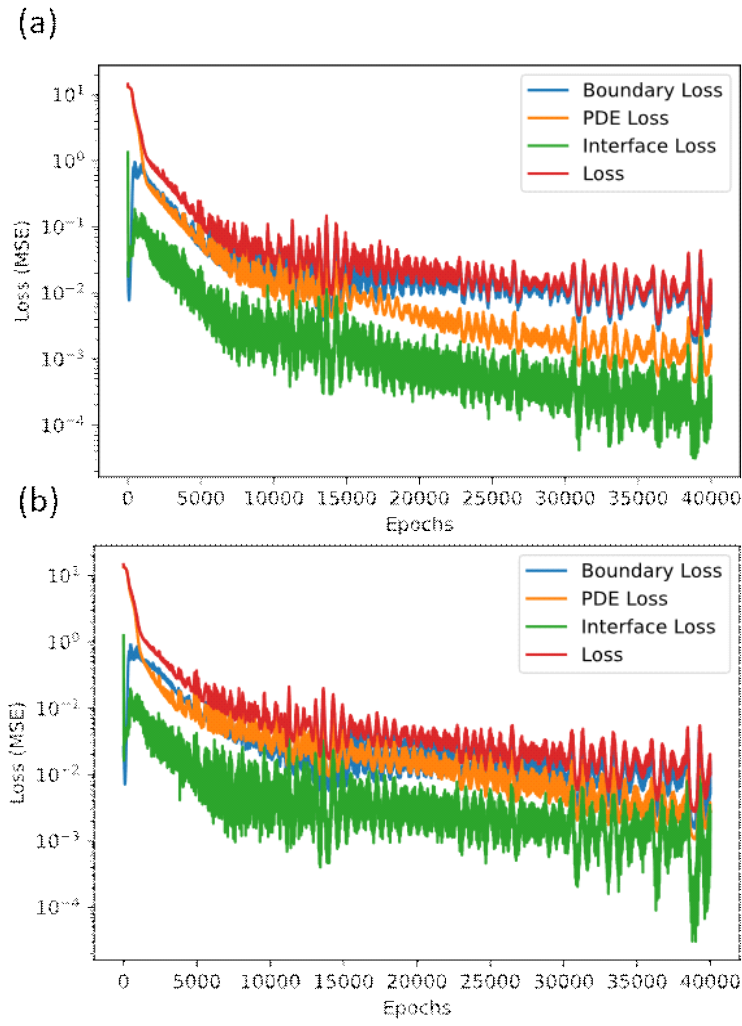


Fig. 4.20 A history of loss during training for 2-D center-fixed plane stress problem: (a) and (b) are the history of loss in subdomain 1 and 2, respectively.

Table 4.2 Normalized mean squared error and and R-squared score of piecewise PINN and the general PINN in 2-D center-fixed plane-stress problem

	Piecewise PINN		General PINN	
	u	v	u	v
NMSE	0.0672	0.1610	0.1628	1.1402
R^2	0.9954	0.9697	0.8731	0.4236

4.2.4 중앙부가 고정된 2차원 판 문제의 해석

다음으로, 중앙부가 고정된 평면 응력 문제를 해석하기 위해서 piecewise PINN를 적용하였다. 총 40000번의 에폭 동안 학습을 진행하였으며 다른 모든 학습 조건은 2차원 판 문제를 해석할 때의 일반 PINN와 동일하게 설정하였다. 소영역 사이의 경계인 인터페이스에서 100개의 데이터를 추출하여 학습에 사용하였다.

해석하고자 한 판 문제는 Fig. 4.21와 같다. x 좌표는 -1에서 1 사이, y 좌표는 -1에서 1사이에서 정의된 직사각형 형상으로, 네 변과 $x=0$ 에서 모두 고정되어 처짐이 발생하지 않는 조건에서 z 방향의 처짐을 계산하였다 (식 (4.10)). 이때, 판의 z 방향으로 작용하는 분포하중은 식 (4.7)과 같고, 판의 굽힘강성 D 는 1로 정의하였다.

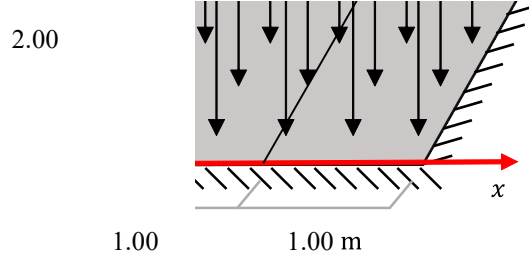


Fig. 4.21 Problem setup and boundary conditions of the center-supported 2-D plate fixed at $x=0$

$$\begin{aligned}
 w(x, -1) = w(x, 1) = w(-1, y) = w(0, y) = w(1, y) = 0 \\
 w_{yy}(x, -1) = w_{yy}(x, 1) = w_{xx}(-1, y) = w_{xx}(1, y) = 0
 \end{aligned} \tag{4.10}$$

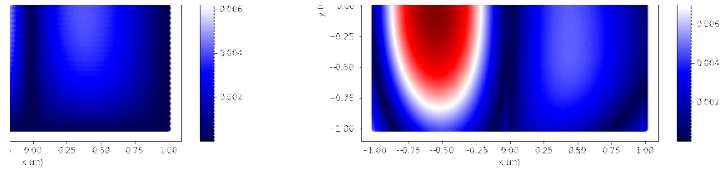
판 문제 역시 일반적인 PINN과 piecewise PINN의 2가지 방법을 사용하여 해석하였다. 이때 영역의 분할 결과는 Fig. 4.18와 같았다. $x=0$ 에서 추가적인 고정이 발생했기 때문에 이 지점을 경계로 2개의 직사각형 모양 소영역으로 분할하였다.

해석 결과를 비교해보면, 일반 PINN은 FEM과 비교했을 때 0.6931의 NMSE와 0.1903의 R^2 를 가져 중간이 지지된 판 문제를 정확하게 해석하지 못하는 것으로 드러났다. 반면 piecewise PINN을 통해 얻은 해는 FEM을 통해 얻은 해와 비교했을 때 0.1396의 NMSE와 0.9490의 R^2 를 가져 일반 PINN보다 훨씬 정확도가 높았다 (Fig. 4.22; Table 4.3). 이를 통해 2차원 판 문제에서도 영역을 소영역으로 분할하지 않을 시 특정 문제를 정확하게 해석할 수 없음이 드러났다.

학습 중 손실의 변화를 살펴보면, 3개의 소영역 모두에서 총 손실에서 차지하는 비중은 L_F 가 가장 컸고 그 다음으로 L_B , L_I 순으로 크게 나타났다 (Fig. 4.23). 2개의 소영역에서 손실의 크기를 비교했을 때는 비슷한 것으로 드러났다.

(a)

(b)



(c)

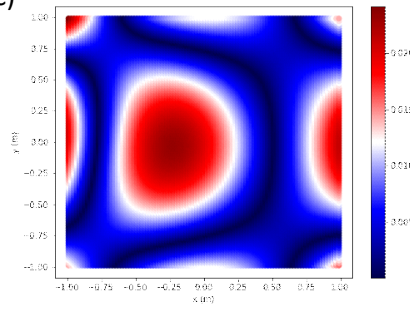


Fig. 4.22 A comparison among the PINN, piecewise PINN and FEM solutions of 2-D center-supported plate problem: (a), (b), (c) are displacement in z direction, predicted by piecewise PINN, FEM, and the general PINN.

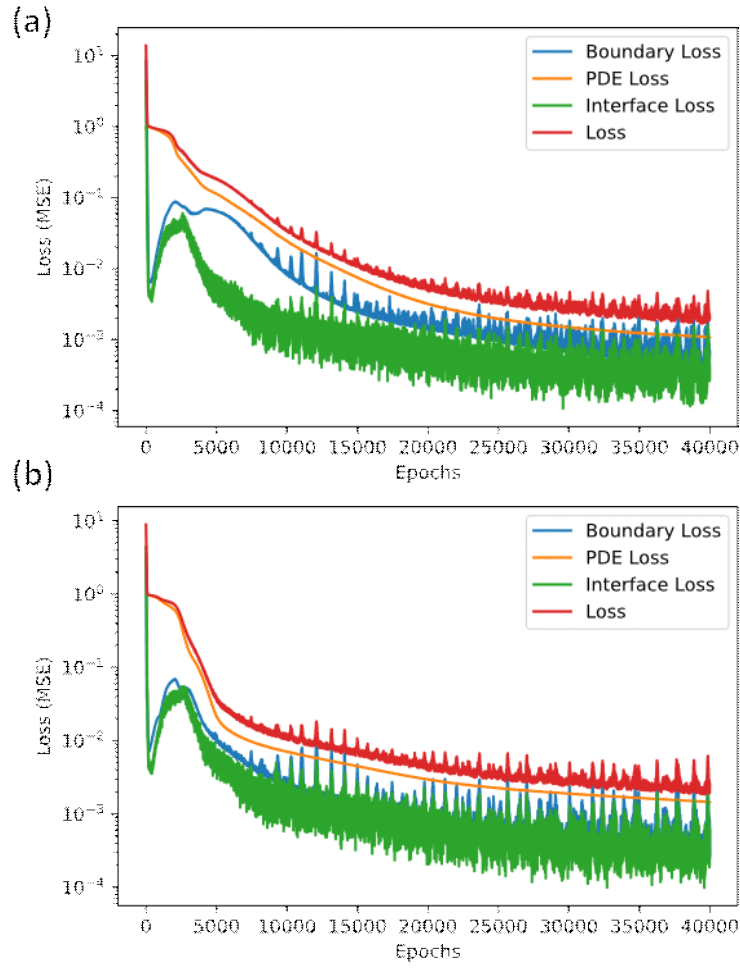


Fig. 4.23 A history of loss during training for 2-D center-supported plate problem: (a) and (b) are the history of loss in subdomain 1 and 2, respectively.

Table 4.3 Normalized mean squared error and R-squared score of piecewise PINN and the general PINN in 2-D center-supported plate problem

	Piecewise PINN	General PINN
NMSE	0.1396	0.6931
R^2	0.9490	0.1003

4.3 PINN과 piecewise PINN의 성능 비교

다음으로는 앞서 4.1에서 일반 PINN으로 해석한 문제 중 일부에 대하여 piecewise PINN을 적용해 얻은 근사해와 일반 PINN으로 얻은 근사해의 정확도를 비교하였다. 이때 학습조건은 일반 PINN으로 학습할 때와 동일하게 두었으며, 학습이 모두 끝난 후 FEM과 비교하여 NMSE를 계산하였다.

1차원 단순보 문제 (Fig. 4.1)에 대하여 piecewise PINN을 적용하였는데, 영역 분할을 하지 않았을 때, 2개의 소영역과 3개의 소영역으로 나누었을 때의 결과를 각각 비교하였다. 2개의 소영역으로 나누었을 때에는 $x=0.0$ 을 기준으로, 3개의 소영역으로 나누었을 때에는 $x=-0.5$ 와 $x=0.5$ 를 기준으로 나누었다. 각 경우에서의 근사해를 Fig. 4.24에 제시하였으며, 마찬가지로 각 경우에서의 지점별 절대오차를 Fig. 4.25에 제시하였다. 영역 분할을 하지 않았을 때에는 NMSE가 0.0053이고 R^2 는 0.9999였고, 2개의 소영역으로 분할했을 때는 NMSE가 0.0080이고 R^2 는 0.9999이었으며, 3개의 소영역으로 분할했을 때는 NMSE가 0.0264이고 R^2 는 0.9962인 것으로 나타났다 (Table 4.4).

Table 4.4 Normalized mean squared error and R-squared score of the general PINN and the piecewise PINNs with 2 and 3 subdomains in 1-D continuous beam problem

	General PINN	Piecewise PINN - 2 subdomains	Piecewise PINN - 3 subdomains
NMSE	0.0053	0.0080	0.0264
R^2	0.9999	0.9999	0.9962

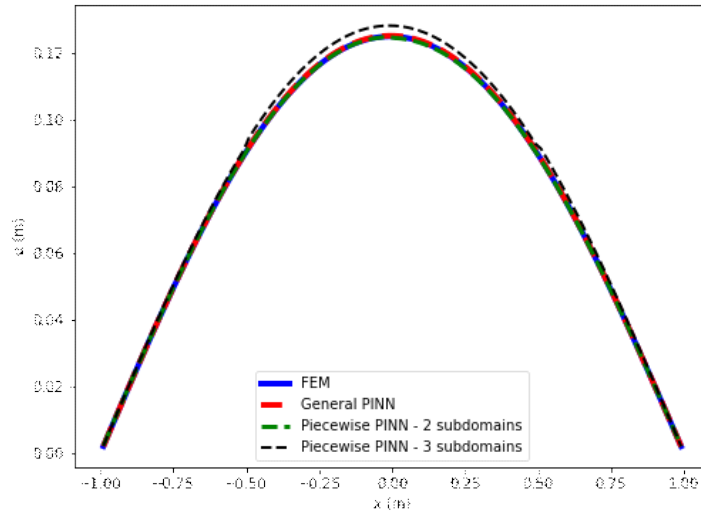


Fig. 4.24 A comparison among the FEM, the general PINN, and the piecewise PINN with 2 subdomains and 3 subdomains solutions of 1-D continuous beam problem

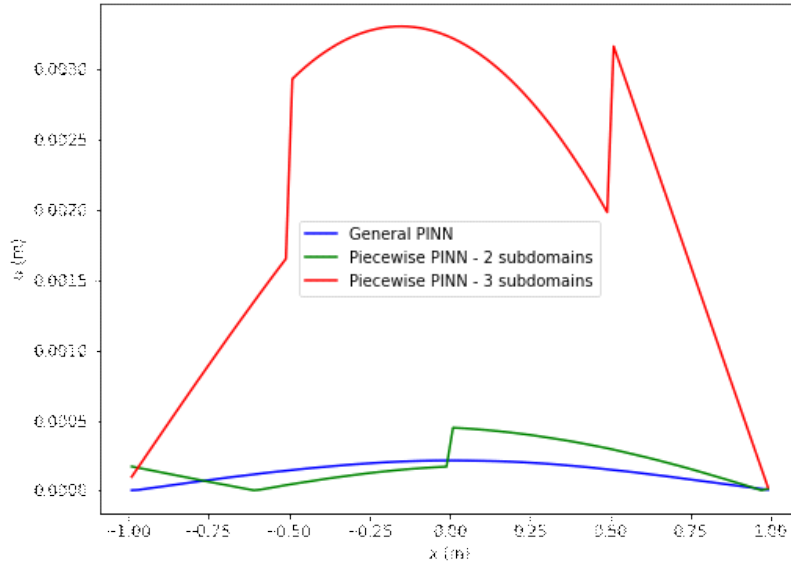


Fig. 4.25 The point-wise error among the FEM, the general PINN, and the piecewise PINN with 2 subdomains and 3 subdomains solutions in 1-D continuous beam problem

다음으로 2차원 평면 응력 문제 (Fig. 4.7)에 대하여 piecewise PINN을 적용하였는데, 영역 분할을 하지 않았을 때와 2개의 소영역으로 분할했을 때의 결과를 각각 비교하였다. 2개의 소영역으로 분할할 때에는 $x = 0.0$ 을 기준으로 2개의 직사각형 영역으로 나누었다. 이때 해석 결과를 Fig. 4.26에 도시하였다. 영역 분할을 하지 않았을 때 x 방향 처짐에 대한 NMSE가 0.0282이고 R^2 는 0.9999였고 y 방향 처짐에 대한 NMSE가 0.0329이고 R^2 는 0.9988인 것으로 나타났으며, 영역 분할을 했을 때는 NMSE가 각각 0.1318, 0.2639 인 것으로 나타났으며 R^2 는 각각 0.8051, 0.9150인 것으로 드러났다 (Table 4.5).

Table 4.5 Normalized mean squared error and R-squared score of the general PINN and the piecewise PINNs with 2 subdomains in 2-D center-fixed plane stress problem

	General PINN		Piecewise PINN - 2 subdomains	
	u	v	u	v
NMSE	0.0282	0.0329	0.1318	0.2639
R^2	0.9999	0.9988	0.8051	0.9150

(a)

(b)

Fig. 4.26 Piecewise PINN solution of the 2-D plane stress problem: (a) and (b) are the displacement field in x and y direction, respectively.

결과적으로, 1차원 보와 2차원 평면 응력 문제에 대하여 일반적인 PINN으로도 충분히 해석이 가능한 문제에서는 영역 분할을 하는 것이 오히려 근사해의 오차를 키운다는 결론을 얻을 수 있었다.

4.4 Piecewise PINN을 활용한 대체 모델 개발

마지막으로, piecewise PINN을 사용하여 탄성체의 거동 문제를 해석할 수 있는 대체 모델을 개발하였다. 적용 대상은 2차원 평면 응력 문제로서, 그 조건은 Fig. 4.17과 식 (4.9), (4.11)에 나타내었다.

$$f_x = -1, f_y = -1 \quad (4.11)$$

이 모델에서는 2차원 평면 응력 문제에 일정한 하중이 작용할 때 임의의 탄성계수에 따라 변위장이 어떻게 변화하는지 확인하고자 하였다. Piecewise PINN의 성능을 평가하기 위해 Piecewise PINN과 단순한 심층 신경망을 각각 학습시켜서 임의의 탄성계수에서의 예측 정확도를 비교하였다. Piecewise PINN과 심층 신경망의 신경망 구조는 Fig. 4.27과 같으며, 기존 신경망에 입력 자료로 사용되는 x, y 좌표 외에도 탄성계수 E 가 포함된 것이 특징이다.

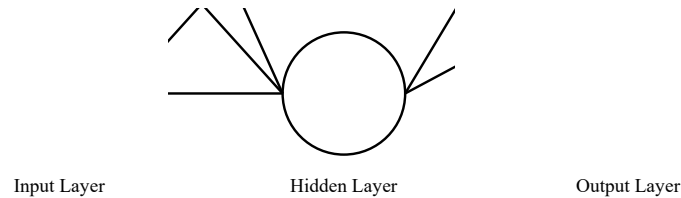


Fig. 4.27 The architecture of piecewise PINN and deep neural networks to train surrogate model of 2-D plane stress problem

이때 사용된 학습 데이터는 탄성계수가 2, 4, 6, 8, 10 Pa일 때로 선정하였으며, 심층 신경망의 학습 시에는 FEM을 통해 얻은 해당 탄성계수일 때의 x 방향, y 방향 변위장 데이터를 사용하여 지도학습을 수행하였다. 한편 심층 신경망을 학습시킬 때의 데이터는 piecewise PINN의 학습 데이터를 추출하는 방식을 그대로 적용하였고, 경계조건에서는 총 500개의 데이터를 추출하였으며 연산 영역 전체에 대하여 무작위하게 1000, 10000, 100000개의 데이터를 추출하여 학습시켰다. 이때 piecewise PINN이 x, y 좌표 값만 필요로 하는 것과 달리 심층 신경망은 해당 좌표 값에 해당

하는 x , y 방향 변위 값까지 필요로 하였다. Piecewise PINN과 심층 신경망의 신경망 구조는 동일하게 입력층과 출력층, 3개의 은닉층으로 구성되었으며 각 층은 20개의 뉴런으로 구성되었다. 또한, 심층 신경망과 piecewise PINN 모두 $x = 0.0$ 을 기준으로 2개의 소영역으로 나누어 학습을 수행하였다.

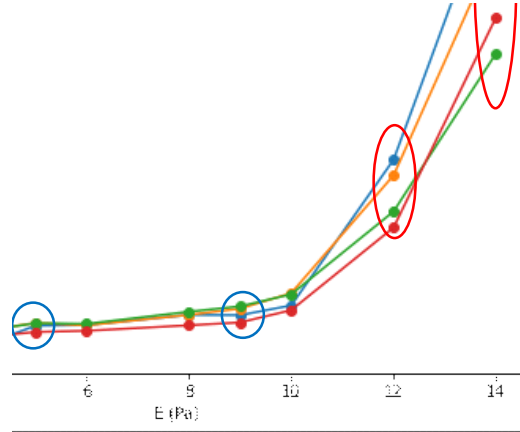
예측 성능을 비교하기 위하여 앞서와 마찬가지로 FEM으로 얻은 근사해와 비교하여 NMSE를 계산하였으며, 탄성계수가 1.5, 12, 14 Pa일 때의 외삽(Extrapolation) 정확도와 5, 9 Pa일 때의 내삽(Interpolation) 정확도를 각각 평가하였다. 이때 각 모델에 대하여 학습 데이터에서의 NMSE가 비슷한 수준이 될 때까지 학습시켜 유사한 학습 조건에서의 예측 정확도를 평가하였다. NMSE는 x , y 방향 변위에 대한 오차를 합쳐서 계산하였다.

탄성계수에 따른 각 모델의 NMSE와 R^2 는 Fig. 4.28과 Table 4.6-4.9와 같다. Fig. 4.28에서 내삽에 해당하는 부분을 파란색 원으로, 외삽에 해당하는 부분을 빨간색 원으로 표기하였다. 또한 같은 그림에서 1000개, 10000개, 100000개의 연산 영역 데이터를 사용하여 학습시킨 심층 신경망의 결과를 각각 DNN-1, DNN-2, DNN-3으로 나타냈고 piecewise PINN의 결과를 PINN으로 나타내었다. 우선 학습 데이터에 해당하는 2, 4, 6, 8, 10 Pa에서의 NMSE를 보면 네 개의 모델이 해당 학습 데이터에 대하여 비슷한 수준으로 학습이 된 것을 확인할 수 있다 (Table 4.6). 그리고 내삽에 해당하는 5, 9 Pa에서의 오차를 보면 piecewise PINN이 근소한 차이로 가장 낮은 하나 그다지 유의미한 차이가 아닌 것으로 판단되었다. 마지막으로 외삽에 해당하는 1.5, 12, 14 Pa에서의 오차를 보면 심층 신경망의 경우 학습에 사용된 데이터가 많을수록 오차가 줄어드는 것으로 확인되었다 (Table 4.7). 그럼에도 불구하고 1.5, 12 Pa에서는 piecewise PINN이 가장 오차가 작았으며 14 Pa에서는 DNN_3 다음으로 piecewise PINN이 가장 오차가 작았다. 이를 통해 piecewise PINN은 라벨링된 데이터를 학습에 요구하지 않는다는 장점 외에도, 동일한 개수의 데이터를 학습한 일반적인 심층 신경망보다 외삽에 훨씬 강점을 보이며, 경우에 따라 10배 더 많은 데이터를 학습한 심층 신경망보다도 더 좋은 외삽 성능을 보이는 것으로 관찰되었다.

탄성계수가 1.5, 5, 12 Pa일 때 각 모델별 모의 결과의 지점별 절대오차를 Fig. 4.29, Fig. 4.30, Fig. 4.31에 각각 나타내었다. 탄성계수가 1.5 Pa일 때의 모델 별 절대오차 분포를 살펴보면, 심층 신경망에서는 학습 데이터 개수에 상관없이 오차가

전 영역에 분포되어 있었으며, 각 소영역의 중앙부에서 가장 크게 발생하였다. 반면 piecewise PINN에서는 소영역의 꼭짓점 부근에서 주변보다 큰 오차가 발생한 것으로 드러났다. 탄성계수가 5 Pa , 12 Pa 일 때는 다른 모델들도 꼭짓점 부근에서 주변보다 큰 오차가 발생했지만, piecewise PINN에 비하면 더 넓은 영역에 오차가 분포하는 것을 확인할 수 있었다. 이는 piecewise PINN을 통해 학습한 대체 모델이 심층 신경망을 통해 학습한 대체 모델과 비교했을 때 대부분의 영역에서 더 정확하고 고른 근사해를 구하며, 소영역의 꼭짓점 등 일부 영역에서 발생하는 주변보다 훨씬 큰 오차가 NMSE의 대부분을 차지하는 것으로 해석할 수 있었다.

(a)



(b)

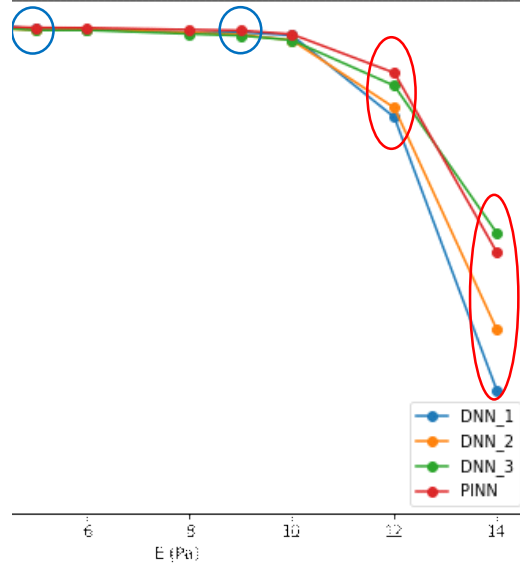


Fig. 4.28 Performance indexes of the models in various elastic modulus: (a) shows NMSE, and (b) shows R-squared score. The red mark denotes the prediction results outside of training data range, and the blue one denotes the results inside of training data range

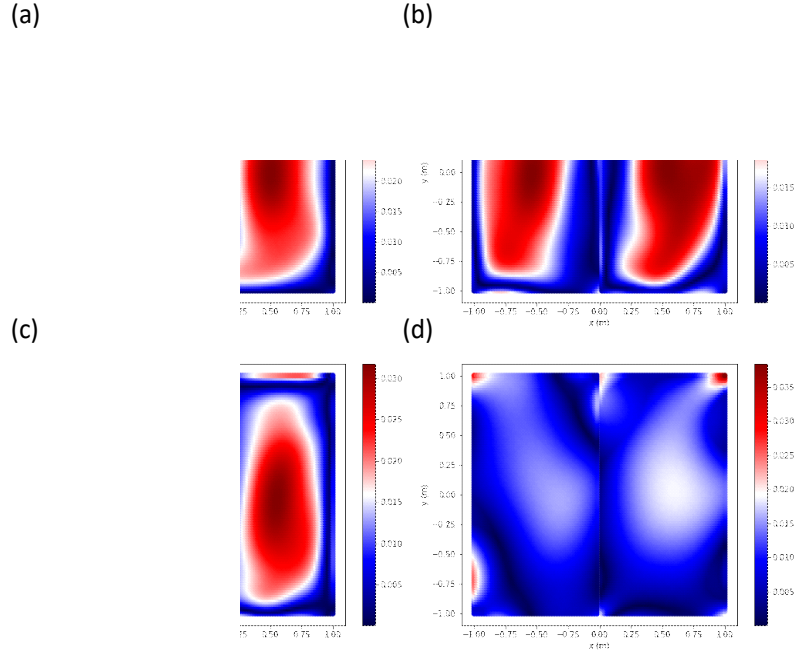


Fig. 4.29 The pointwise absolute error of various surrogate models when $E = 1.5$ Pa: (a) is DNN-1, (b) is DNN-2, (c) is DNN-3, (d) is piecewise PINN.

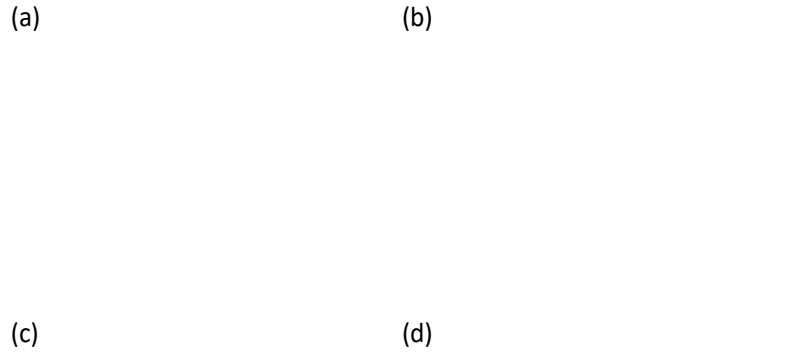


Fig. 4.30 The pointwise absolute error of various surrogate models when $E = 5$ Pa: (a) is DNN-1, (b) is DNN-2, (c) is DNN-3, (d) is piecewise PINN.

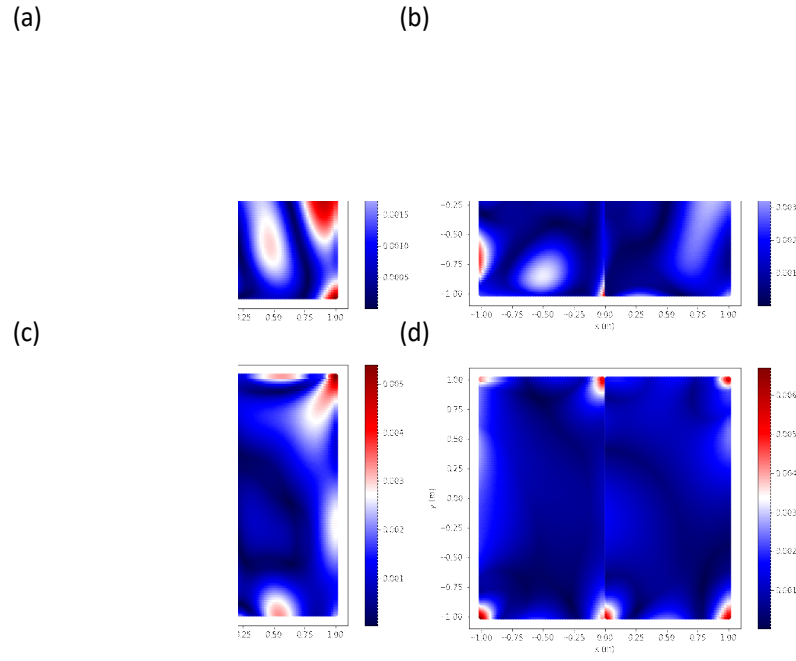


Fig. 4.31 The pointwise absolute error of various surrogate models when $E = 12$ Pa: (a) is DNN-1, (b) is DNN-2, (c) is DNN-3, (d) is piecewise PINN.

Table 4.6 Normalized mean squared error of various surrogate models in training cases, trained using DNN-1, DNN-2, DNN-3, and piecewise PINN

E (Pa)	DNN-1	DNN-2	DNN-3	Piecewise PINN
2	0.0278	0.0233	0.0269	0.0250
4	0.0258	0.0323	0.0321	0.0293
6	0.0348	0.0346	0.0353	0.0321
8	0.0394	0.0395	0.0409	0.0347
10	0.0437	0.0494	0.0488	0.0416

Table 4.7 Normalized mean squared error of various surrogate models in test cases, trained using DNN-1, DNN-2, DNN-3, and piecewise PINN

E (Pa)	DNN-1	DNN-2	DNN-3	Piecewise PINN
1.5	0.1935	0.1681	0.1697	0.0926
5	0.0345	0.0357	0.0356	0.0316
9	0.0393	0.0423	0.0433	0.0360
12	0.1103	0.1032	0.0867	0.0793
14	0.2367	0.2072	0.1587	0.1752

Table 4.8 R-squared score of various surrogate models in training cases, trained using DNN-1, DNN-2, DNN-3, and piecewise PINN

E (Pa)	DNN-1	DNN-2	DNN-3	Piecewise PINN
2	0.9975	0.9982	0.9976	0.9979
4	0.9978	0.9966	0.9967	0.9971
6	0.996	0.996	0.9957	0.9965
8	0.9949	0.9949	0.9944	0.9959
10	0.9939	0.9922	0.9922	0.9943

Table 4.9 R-squared score of various surrogate models in test cases, trained using DNN-1, DNN-2, DNN-3, and piecewise PINN

E (Pa)	DNN-1	DNN-2	DNN-3	Piecewise PINN
1.5	0.835	0.8695	0.8597	0.9666
5	0.9961	0.9957	0.9957	0.9966
9	0.995	0.9942	0.9938	0.9956
12	0.9653	0.9686	0.9763	0.9808
14	0.87	0.8912	0.9249	0.9182

제 5 장 요약 및 결론

본 연구에서는 FEM의 한계를 극복하기 위한 대체 모델을 학습하기 위한 방법으로 PINN을 채택하였는데, 이 방법은 많은 라벨링된 데이터를 필요로 하는 DNN 기반의 대체 모델과 달리 라벨링된 데이터를 필요로 하지 않는다는 장점이 있다. 이에 여태까지 주로 FDM, FEM 등의 격자 기반 수치해석 방법으로 해석되어 왔던 탄성체의 거동 문제를 PINN을 통해 해석하였고, 미분 불가능한 지점이 발생하여 일반적인 PINN으로 해석할 수 없었던 문제들에 영역 분할을 적용하는 새로운 구조인 piecewise PINN을 제안하였다. 그리고 piecewise PINN을 통해 임의의 탄성계수가 주어졌을 때 평면 응력 문제에서 처짐의 분포를 모의하는 대체 모델을 학습시키고, 단순 DNN 지도학습 기반의 대체 모델과 정확도를 비교하여 piecewise PINN의 대체 모델에 대한 적용성을 증명하고자 하였다.

먼저 일반적인 PINN을 구현하여 1차원 단순보 문제와 1차원 캔틸레버보 문제를 해석하였으며, FEM과 비교했을 때 각각 0.0053, 0.0141의 NMSE와 0.9999, 0.9999의 R^2 를 가져 거의 차이가 없는 것으로 드러났다. 다음으로 사방이 고정된 2차원 평면 응력 문제를 해석하였고, x 방향 처짐에서 FEM과 비교하여 얻은 NMSE는 0.0282, R^2 는 0.9999였고, y 방향 처짐에 대해서는 0.0329의 NMSE와 0.9988의 R^2 를 가져 마찬가지로 거의 차이가 없는 근사해를 구한 것으로 판단되었다 이후 사방이 단단 지지된 2차원 판 문제를 해석하였으며, 이때 z 방향 처짐의 NMSE는 0.0156, R^2 는 0.9999로 FEM과 큰 차이가 없이 처짐을 모의한 것으로 판단되었다.

다음으로는 일반적인 PINN을 적용해서는 정확하게 풀 수 없었던 1차원 연속보 문제, 중앙이 지지된 2차원 평면 응력 문제, 중앙이 지지된 2차원 판 문제에 대하여 영역 분할 기반의 piecewise PINN을 적용하여 해석하였다. 먼저 1차원 연속보의 경우에는 piecewise PINN에서 얻은 해의 NMSE는 0.0045, R^2 는 0.9999이었으며, 이때 동일 문제에 대해 일반 PINN을 적용하여 얻은 해의 NMSE는 1.2345, R^2 는 0.0772로 piecewise PINN을 적용하지 않을 시 정확한 해를 구할 수 없음을 알 수 있었다. 또한 중앙부가 고정된 2차원 평면 응력 문제에 대하여 일반 PINN으로 해석하였을 때 x 방향 처짐에서 FEM과 비교하여 얻은 NMSE는 0.1628, R^2 는 0.8731이었고 y 방향 처짐에 대해서는 1.1402의 NMSE와 0.4236의 R^2 를 얻었다. 반면 piecewise

PINN을 적용하였을 때는 x 방향 처짐에서 0.0672의 NMSE와 0.9954의 R^2 , y 방향 처짐에서 0.1610의 NMSE와 0.9697의 R^2 를 보여 마찬가지로 piecewise PINN이 해당 문제에 대하여 일반 PINN보다 더 정확한 해를 얻을 수 있음을 보여주었다. 마지막으로, 중앙부가 지지된 2차원 판 문제에서는 일반 PINN의 NMSE가 0.6931이고 R^2 는 0.1003이었으며, piecewise PINN의 NMSE는 0.1396이고 R^2 는 0.9490으로 드러나 piecewise PINN이 해당 문제를 해석할 때 훨씬 정확한 해를 얻을 수 있음을 나타내었다.

다음으로는 일반 PINN으로 충분히 정확한 해를 얻을 수 있었던 문제 중 일부에 임의로 구간을 분할한 뒤 piecewise PINN을 적용하여, 어떤 방법이 더 정확한 근사해를 제공하는지 비교해보았다. 1차원 보와 2차원 평면 응력 문제에서 모두 일반적인 PINN이 더 나은 정확도를 보이는 것으로 나타났다. 결과적으로 연속보 문제나 중앙부가 지지된 판 문제처럼 piecewise PINN을 적용하지 않으면 정확한 근사해를 찾기 어려운, 미분 불가능한 지점이 발생하는 문제에 대해서만 piecewise PINN을 적용해야 할 것으로 사료된다.

마지막으로 piecewise PINN을 사용하여 다양한 탄성계수가 주어졌을 때 평면 응력 문제에서 처짐을 모의하는 대체 모델을 학습하였다. 해당 모델의 성능을 평가하기 위하여, 단순히 DNN을 이용하여 지도학습으로 학습시킨 케이스들과 비교하였는데, 이때 학습 데이터의 추출 방식은 PINN을 학습시킬 때의 그것과 동일했으며, DNN은 FEM을 통해 얻은 라벨링된 데이터를 필요로 하는 반면 piecewise PINN은 그렇지 않다는 차이점이 존재하였다. 이때 전체 영역에서 1000개, 10000개, 100000개를 추출하여 총 3개의 DNN 기반 대체 모델을 학습하였으며, $E=2, 4, 6, 8, 10$ Pa 일 때에 대하여 학습하였다. 모든 케이스에 대하여 학습 데이터에서 비슷한 수준의 정확도를 보일 때 까지 학습시켜 동일한 데이터에 대해 비슷한 수준으로 학습된 서로 다른 모델들에 대하여 임의의 데이터에 대한 예측 성능을 평가하였다. 결과적으로 $E=5, 9$ Pa에서 내압 성능을 평가하였을 때는 근소하지만 piecewise PINN이 모든 경우에서 우세하였으며, $E=1.5, 12, 14$ Pa에서 외압 성능을 평가하였을 때는 $E=14$ Pa에서 100000개의 데이터로 학습시킨 DNN에서 정확도가 열세였던 것을 제외하면 모든 지점에서 우세하였다. 또 각 모델 별로 절대 오차의 분포를 살펴보면, DNN은 오차가 상대적으로 골고루 분포했던 반면, piecewise PINN은 특정 지점에서만 큰 오차가 발생하여, 상대적으로 piecewise PINN을 통해 학습시킨 대체 모델

이 더 넓은 영역에서 정확한 해를 제공하는 것으로 드러났다.

이를 통하여, 대체 모델을 학습할 때 piecewise PINN이 단순 DNN과 비교하였을 때 가지는 이점으로 (1) 라벨링된 데이터를 전혀 필요로 하지 않고, (2) 심지어 데이터의 개수가 적어도 대부분 우세한 예측 성능을 보인다는 것을 확인할 수 있었는데, 이는 단순히 학습 데이터에 피팅하는 DNN과 달리 물리 현상을 학습하는 piecewise PINN의 구조 덕분인 것으로 판단되었다.

참 고 문 헌

1. Anantharaman, R., Ma, Y., Gowda, S., Laughman, C., Shah, V., Edelman, A., & Rackauckas, C. (2020). Accelerating simulation of stiff nonlinear systems using continuous-time echo state networks. *arXiv preprint arXiv:2010.04004*.
2. Bender, E. A. (2000). *An introduction to mathematical modeling*. Courier Corporation.
3. bin Waheed, U., Haghighat, E., Alkhalifah, T., Song, C., & Hao, Q. (2021). PINNeik: Eikonal solution using physics-informed neural networks. *Computers & Geosciences*, 104833.
4. Boresi, A. P., Chong, K. P., & Saigal, S. (2003). *Approximate solution methods in engineering mechanics*. John Wiley & Sons.
5. Cao, Y., Fang, Z., Wu, Y., Zhou, D. X., & Gu, Q. (2019). Towards understanding the spectral bias of deep learning. *arXiv preprint arXiv:1912.01198*.
6. Chapra, S. C., & Canale, R. P. (2011). *Numerical methods for engineers (Vol. 1221)*. New York: Mcgraw-hill.
7. Chen, Y., Lu, L., Karniadakis, G. E., & Dal Negro, L. (2020). Physics-informed neural networks for inverse problems in nano-optics and metamaterials. *Optics express*, 28(8), 11618–11633.
8. Chou, P. C., & Pagano, N. J. (1992). *Elasticity: tensor, dyadic, and engineering approaches*. Courier Corporation.
9. Gere, J. M., & Goodno, B. J. (2013). *Mechanics of Materials*, 8th Eds. Cengage Learning, 387–399.
10. Goswami, S., Anitescu, C., Chakraborty, S., & Rabczuk, T. (2020). Transfer learning enhanced physics informed neural network for phase-field modeling of fracture. *Theoretical and Applied Fracture Mechanics*, 106, 102447.
11. Haghighat, E., Bekar, A. C., Madenci, E., & Juanes, R. (2021). A nonlocal physics-informed deep learning framework using the peridynamic differential

- operator. *Computer Methods in Applied Mechanics and Engineering*, 385, 114012.
12. Haghighat, E., & Juanes, R. (2021). Sciann: A keras/tensorflow wrapper for scientific computations and physics-informed deep learning using artificial neural networks. *Computer Methods in Applied Mechanics and Engineering*, 373, 113552.
 13. Hoffer, J. G., Geiger, B. C., Ofner, P., & Kern, R. (2021). Mesh-Free Surrogate Models for Structural Mechanic FEM Simulation: A Comparative Study of Approaches. *Applied Sciences*, 11(20), 9411
 14. Jagtap, A. D., & Karniadakis, G. E. (2020). Extended physics-informed neural networks (xpinns): A generalized space-time domain decomposition based deep learning framework for nonlinear partial differential equations. *Communications in Computational Physics*, 28(5), 2002–2041.
 15. Jagtap, A. D., Kharazmi, E., & Karniadakis, G. E. (2020). Conservative physics-informed neural networks on discrete domains for conservation laws: Applications to forward and inverse problems. *Computer Methods in Applied Mechanics and Engineering*, 365, 113028.
 16. Kharazmi, E., Zhang, Z., & Karniadakis, G. E. (2021). hp-VPINNs: Variational physics-informed neural networks with domain decomposition. *Computer Methods in Applied Mechanics and Engineering*, 374, 113547.
 17. Kim, Y., Choi, Y., Widemann, D., & Zohdi, T. (2021). A fast and accurate physics-informed neural network reduced order model with shallow masked autoencoder. *Journal of Computational Physics*, 110841.
 18. Laubscher, R., & Rousseau, P. (2021). Application of a mixed variable physics-informed neural network to solve the incompressible steady-state and transient mass, momentum, and energy conservation equations for flow over in-line heated tubes. *Applied Soft Computing*, 108050.
 19. LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436–444.
 20. Libbrecht, M. W., & Noble, W. S. (2015). Machine learning applications in

- genetics and genomics. *Nature Reviews Genetics*, 16(6), 321–332.
21. Lucor, D., Agrawal, A., & Sergent, A. (2021). Physics-aware deep neural networks for surrogate modeling of turbulent natural convection. *arXiv preprint arXiv:2103.03565*.
 22. Mao, Z., Jagtap, A. D., & Karniadakis, G. E. (2020). Physics-informed neural networks for high-speed flows. *Computer Methods in Applied Mechanics and Engineering*, 360, 112789.
 23. Moseley, B., Markham, A., & Nissen-Meyer, T. (2021). Finite Basis Physics-Informed Neural Networks (FBPINNs): a scalable domain decomposition approach for solving differential equations. *arXiv preprint arXiv:2107.07871*.
 24. Nabian, M. A., Gladstone, R. J., & Meidani, H. (2021). Efficient training of physics informed neural networks via importance sampling. *Computer Aided Civil and Infrastructure Engineering*.
 25. Niaki, S. A., Haghighat, E., Campbell, T., Poursartip, A., & Vaziri, R. (2021). Physics-informed neural network for modelling the thermochemical curing process of composite-tool systems during manufacture. *Computer Methods in Applied Mechanics and Engineering*, 384, 113959.
 26. Nielsen, M. A. (2015). *Neural networks and deep learning (Vol. 25)*. San Francisco, CA: Determination press.
 27. Pun, G. P., Batra, R., Ramprasad, R., & Mishin, Y. (2019). Physically informed artificial neural networks for atomistic modeling of materials. *Nature communications*, 10(1), 1–10.
 28. Raissi, M., Perdikaris, P., & Karniadakis, G. E. (2019). 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.
 29. Rafiei, M. H., & Adeli, H. (2017). A novel machine learning based algorithm to detect damage in high rise building structures. *The Structural Design of Tall and Special Buildings*, 26(18), e1400.

30. Rahaman, N., Baratin, A., Arpit, D., Draxler, F., Lin, M., Hamprecht, F., ... & Courville, A. (2019, May). On the spectral bias of neural networks. In *International Conference on Machine Learning* (pp. 5301–5310). PMLR.
31. Rao, C., Sun, H., & Liu, Y. (2021). Physics-Informed Deep Learning for Computational Elastodynamics without Labeled Data. *Journal of Engineering Mechanics*, 147(8), 04021043.
32. Samaniego, E., Anitescu, C., Goswami, S., Nguyen-Thanh, V. M., Guo, H., Hamdia, K., Zhuang, X., & Rabczuk, T. (2020). An energy approach to the solution of partial differential equations in computational mechanics via machine learning: Concepts, implementation and applications. *Computer Methods in Applied Mechanics and Engineering*, 362, 112790.
33. Shukla, K., Jagtap, A. D., & Karniadakis, G. E. (2021). Parallel physics-informed neural networks via domain decomposition. *arXiv preprint arXiv:2104.10013*.
34. Sun, L., Gao, H., Pan, S., & Wang, J. X. (2020). Surrogate modeling for fluid flows based on physics-constrained deep learning without simulation data. *Computer Methods in Applied Mechanics and Engineering*, 361, 112732.
35. Timoshenko, S., & Woinowsky-Krieger, S. (1959). *Theory of plates and shells*.
36. Ugural, A. C. (2009). *Stresses in beams, plates, and shells*. CRC press.
37. Wang, S., Wang, H., & Perdikaris, P. (2021). On the eigenvector bias of Fourier feature networks: From regression to solving multi-scale PDEs with physics-informed neural networks. *Computer Methods in Applied Mechanics and Engineering*, 384, 113938.
38. Willard, J., Jia, X., Xu, S., Steinbach, M., & Kumar, V. (2020). Integrating physics-based modeling with machine learning: A survey. *arXiv preprint arXiv:2003.04919*, 1(1), 1–34.
39. Xu, Z. Q. J., Zhang, Y., Luo, T., Xiao, Y., & Ma, Z. (2019). Frequency principle: Fourier analysis sheds light on deep neural networks. *arXiv preprint arXiv:1901.06523*.

40. Zhu, Y., Zabaras, N., Koutsourelakis, P. S., & Perdikaris, P. (2019). Physics-constrained deep learning for high-dimensional surrogate modeling and uncertainty quantification without labeled data. *Journal of Computational Physics*, 394, 56-81.

Abstract

Analysis of Continuous Elastic Structure using Physics Informed Neural Networks

Jeong, Youngjoon

Major in Rural Systems Engineering

Department of Landscape Architecture and

Rural Systems Engineering

The Graduate School

Seoul National University

To overcome the disadvantages of making repetitive interpretation difficult due to a large increase in the computational load of FEM that occur when mesh generation is difficult, a surrogate model that provides approximation faster than FEM has attracted attention. Most surrogate models are learned with supervised learning algorithms, and have often been learned based on DNN recently, which has the disadvantage of requiring a large amount of labeled data. However, in the case where a sufficient amount of labeled data cannot be obtained, the need for a physics informed neural network (PINN) is emerging. PINN is a type of DNN, which is a method of learning boundary conditions, initial conditions, and governing equations to obtain approximate solutions of differential equations, and was evaluated as suitable for learning surrogate models in small data problems because it does not require labeled data. However, PINN had the disadvantage of not accurately approximating the

non-differentiable solution because it provides a fully differentiable solution. In this study, various structural mechanics problems were interpreted with PINNs to make surrogate models for structural mechanics problems that PINNs have not sufficiently dealt with. Also, piecewise PINN, a new type of PINN, which computational domain was decomposed, was proposed and applied. In addition, the accuracy of the two methods was compared by applying piecewise PINN to problems that can also be solved with general PINN. The target problems were one-dimensional beam problem, two-dimensional plane stress problem, and two-dimensional plate problem, and the performance of PINN was evaluated by calculating NMSE (Normalized Mean Square Error) and R^2 compared to the data from FEM. As a result, an accurate solution could be obtained with a general PINN for problems such as one-dimensional simple beam, cantilever beam, or plane stress and plate problem supported in four directions. However, piecewise PINN provided more accurate solution compared with a general PINN in one-dimensional continuous beam problem with concentrated load, and a plane stress and plate problem with an additional support. In addition, the application of piecewise PINN in a problem in which a sufficiently accurate solution could be obtained even with a general PINN showed that the general PINN was more accurate. Finally, surrogate model for the center-fixed two-dimensional plane stress problem was developed using piecewise PINN, and compared it with a DNN-based surrogate model learned with labeled data from FEM to evaluate the accuracy of that model. Consequently, surrogate model based on piecewise PINN showed superior performance in almost all cases over DNN-based surrogate models, which showed that piecewise PINN did not require labeled data as opposed to DNN when learning surrogate models, even requiring less training data than DNN, while showing better prediction performance in intrapolation or extrapolation.

keywords : Numerical analysis, Surrogate model, Structural mechanics,

Physics informed neural networks (PINN), piecewise PINN, Domain decomposition

Student Number : 2020-29376