Homework #5, 繳交期限 2024/12/20
物理資訊神經網路，PINN

系所：工海所

姓名：劉樺

學號：R13525030

物理資訊神經網路透過將損失函數(Loss function)定義為偏微分方程(PDE)之殘差與初始/邊界條件(IC/BC)之殘差，透過深度學習(數據回歸)的架構訓練出一套模型，並可用來預測該問題之解。

本作業嘗試透過物理資訊神經網路，求取一維波動方程式之解，該問題與初始邊界條件如下：

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

$$E(0,t) = E(1,t) = 0,\ t \in [0,1]$$

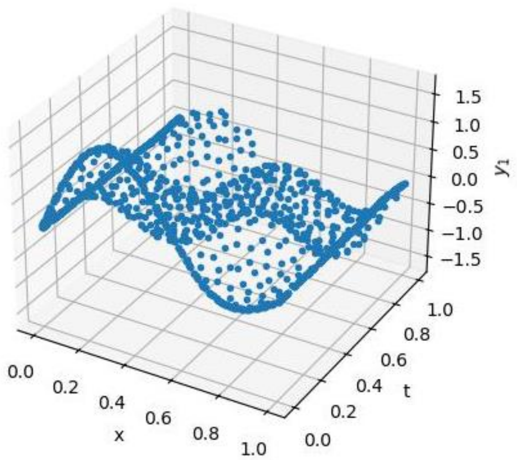
$$E(x,0) = \sin(\pi x) + \sin(A\pi x),\ \mathrm{x} \in [0,1]$$

$$\left.\frac{\partial E(x,t)}{\partial t}\right|_{t=0} = 0,\ \mathrm{x} \in [0,1]$$

根據 d'Alembert solution，跟問題之分析解為：

$$E(x,t) = \sin(\pi x)\cos(c\pi t) + \sin(A\pi x)\cos(Ac\pi t)$$

透過 PINN 求得解之結果如下：



Homework #5, 繳交期限 2024/12/20
物理資訊神經網路，PINN

系所：工海所

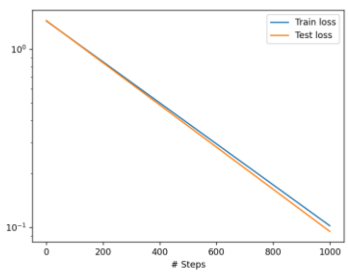
姓名：劉樺

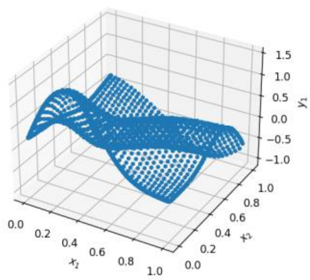
學號：R13525030

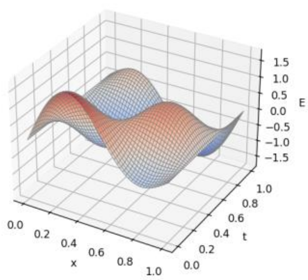
Basic

成功跑出神經網路回歸之解與分析解

```
Training model...

Step      Train loss                                        Test loss                                         Test metric
0         [6.32e-03, 4.84e-02, 1.06e+00, 2.33e-01]    [7.05e-03, 4.84e-02, 1.06e+00, 2.33e-01]    []
1000      [1.60e-03, 3.46e-02, 2.18e-02, 2.33e-03]    [2.09e-03, 3.46e-02, 2.18e-02, 2.33e-03]    []

Best model at step 1000:
  train loss: 6.04e-02
  test loss: 6.09e-02
  test metric: []

'train' took 14.994987 s
```







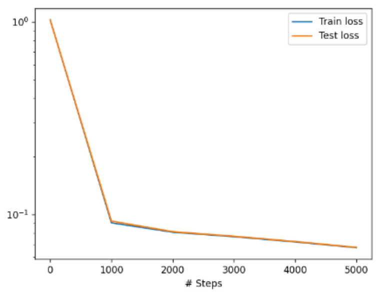

```python
#Construct PDE data
'''Homework!!! Setting the point of training'''
data=dde.data.TimePDE(geomtime, pde, [bc, ic_1, ic_2], num_domain=1000, num_boundary=800, num_initial=200, solution=func, num_test=1000)
#All the value are suggest range from 100~1000

# Construct Neural Network
'''Homework!!! Setting layer size'''
'''Homework!!! Adding hidden layer and different width of layer'''
layer_size=[2]+[32]*3+[1] #first XXXX is the width of neural, second XX is layer of hidden layer
activation='tanh'
initializer='Glorot uniform'
net=dde.nn.FNN(layer_size, activation, initializer)

# Construct Model
'''Homework!!! Setting learning rate'''
model=dde.Model(data,net)
model.compile('adam', lr=1e-4) #suggest range 1e-4~1e-2

#Training Model
'''Homework!!! Setting iteration(number of training) '''
losshistory, train_state=model.train(iterations=10000) #suggest range 1000~50000
```

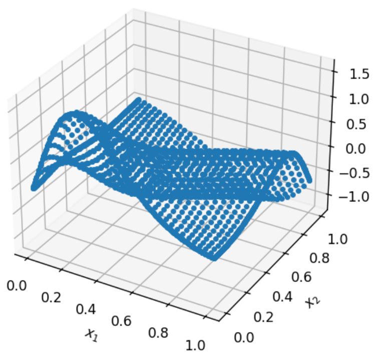
Intermediate:

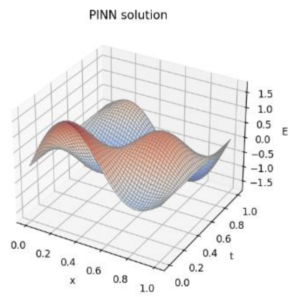
增加不同神經元之寬度與層數，進行更深度之學習

將 iteration 增加為 5000，並增加了網路的深度（隱藏層數）和每層的神經元數量

```python
# Construct Neural Network
'''Homework!!! Setting layer size'''
'''Homework!!! Adding hidden layer and different width of layer'''
layer_size=[2]+[64]*5+[1] #first XXXX is the width of neural, second XX is layer of hidden layer
activation='tanh'
initializer='Glorot uniform'
net=dde.nn.FNN(layer_size, activation, initializer)

# Construct Model
'''Homework!!! Setting learning rate'''
model=dde.Model(data,net)
model.compile('adam', lr=1e-4) #suggest range 1e-4~1e-2

#Training Model
'''Homework!!! Setting iteration(number of training) '''
losshistory, train_state=model.train(iterations=5000) #suggest range 1000~50000

```








```
Training model...

Step      Train loss                                  Test loss                                   Test metric
0         [7.25e-04, 1.90e-03, 1.01e+00, 1.09e-02]    [7.83e-04, 1.90e-03, 1.01e+00, 1.09e-02]    []
1000      [3.38e-03, 3.40e-02, 5.21e-02, 3.54e-04]    [4.91e-03, 3.40e-02, 5.21e-02, 3.54e-04]    []
2000      [9.01e-04, 3.74e-02, 4.38e-02, 4.13e-04]    [1.55e-03, 3.74e-02, 4.38e-02, 4.13e-04]    []
3000      [8.70e-04, 3.78e-02, 4.01e-02, 4.48e-04]    [1.39e-03, 3.78e-02, 4.01e-02, 4.48e-04]    []
4000      [1.20e-03, 3.77e-02, 3.59e-02, 5.28e-04]    [1.52e-03, 3.77e-02, 3.59e-02, 5.28e-04]    []
5000      [1.04e-03, 3.75e-02, 3.03e-02, 8.05e-04]    [1.34e-03, 3.75e-02, 3.03e-02, 8.05e-04]    []
6000      [1.16e-03, 3.78e-02, 2.40e-02, 1.08e-03]    [1.62e-03, 3.78e-02, 2.40e-02, 1.08e-03]    []
7000      [1.16e-03, 3.81e-02, 1.79e-02, 9.00e-04]    [1.83e-03, 3.81e-02, 1.79e-02, 9.00e-04]    []
8000      [1.55e-03, 3.73e-02, 1.27e-02, 5.86e-04]    [2.18e-03, 3.73e-02, 1.27e-02, 5.86e-04]    []
9000      [1.10e-03, 3.57e-02, 8.73e-03, 3.94e-04]    [1.45e-03, 3.57e-02, 8.73e-03, 3.94e-04]    []
10000     [2.78e-03, 3.34e-02, 6.63e-03, 5.70e-04]    [2.34e-03, 3.34e-02, 6.63e-03, 5.70e-04]    []

Best model at step 10000:
  train loss: 4.34e-02
  test loss: 4.29e-02
  train loss: 4.34e-02
  train loss: 4.34e-02
  test loss: 4.29e-02
  test metric: []

'train' took 955.473315 s
```

Homework #5, 繳交期限 2024/12/20
物理資訊神經網路，PINN

系所：工海所

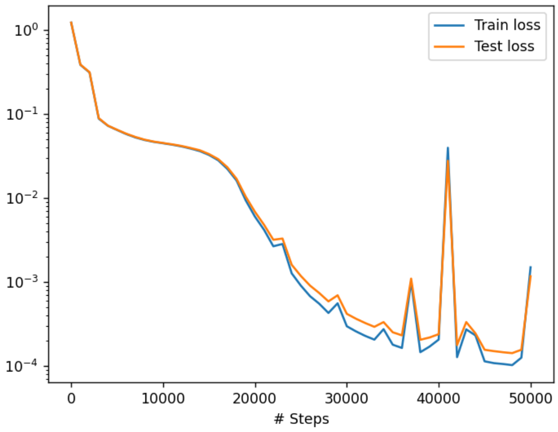
姓名：劉梓

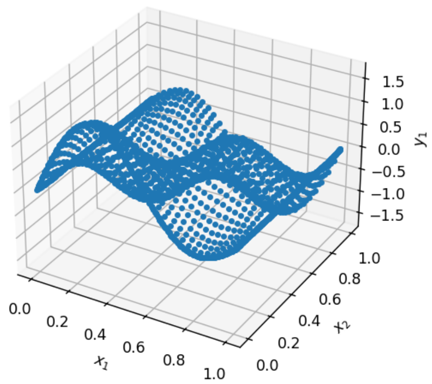
學號：R13525030

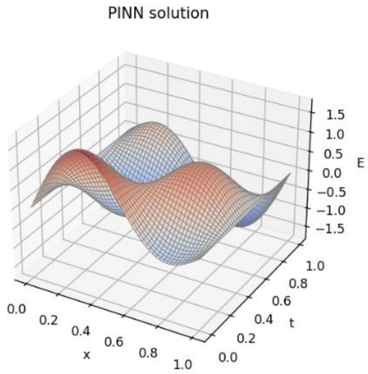
Advanced

劃出分上圖分析解之圖形，注意網格(mesh)的顏色需為灰色

將 iteration 增加為 50000








```
Training model...

Step      Train loss                                    Test loss                                     Test metric
0         [1.07e-03, 6.09e-02, 1.08e+00, 7.93e-02]     [1.17e-03, 6.09e-02, 1.08e+00, 7.93e-02]     []
1000      [6.25e-03, 6.01e-02, 3.07e-01, 1.10e-02]     [9.60e-03, 6.01e-02, 3.07e-01, 1.10e-02]     []
2000      [6.63e-03, 4.00e-02, 2.59e-01, 5.96e-03]     [9.35e-03, 4.00e-02, 2.59e-01, 5.96e-03]     []
3000      [5.47e-03, 3.71e-02, 4.32e-02, 2.27e-03]     [6.69e-03, 3.71e-02, 4.32e-02, 2.27e-03]     []
4000      [2.39e-03, 3.85e-02, 2.88e-02, 2.54e-03]     [2.71e-03, 3.85e-02, 2.88e-02, 2.54e-03]     []
5000      [1.86e-03, 3.72e-02, 2.30e-02, 2.29e-03]     [2.28e-03, 3.72e-02, 2.30e-02, 2.29e-03]     []
6000      [1.77e-03, 3.65e-02, 1.72e-02, 2.06e-03]     [2.40e-03, 3.65e-02, 1.72e-02, 2.06e-03]     []
7000      [1.59e-03, 3.65e-02, 1.29e-02, 1.51e-03]     [2.27e-03, 3.65e-02, 1.29e-02, 1.51e-03]     []
8000      [1.24e-03, 3.64e-02, 1.04e-02, 8.68e-04]     [1.71e-03, 3.64e-02, 1.04e-02, 8.68e-04]     []
9000      [9.80e-04, 3.60e-02, 9.05e-03, 5.08e-04]     [1.32e-03, 3.60e-02, 9.05e-03, 5.08e-04]     []
10000     [8.62e-04, 3.52e-02, 8.32e-03, 3.69e-04]     [1.19e-03, 3.52e-02, 8.32e-03, 3.69e-04]     []
11000     [8.08e-04, 3.44e-02, 7.60e-03, 3.04e-04]     [1.20e-03, 3.44e-02, 7.60e-03, 3.04e-04]     []
12000     [8.39e-04, 3.32e-02, 6.75e-03, 3.03e-04]     [1.36e-03, 3.32e-02, 6.75e-03, 3.03e-04]     []
13000     [8.46e-04, 3.19e-02, 5.76e-03, 2.86e-04]     [1.44e-03, 3.19e-02, 5.76e-03, 2.86e-04]     []
14000     [1.17e-03, 2.99e-02, 4.80e-03, 3.08e-04]     [1.99e-03, 2.99e-02, 4.80e-03, 3.08e-04]     []
15000     [9.19e-04, 2.75e-02, 3.81e-03, 2.74e-04]     [1.73e-03, 2.75e-02, 3.81e-03, 2.74e-04]     []
16000     [9.27e-04, 2.39e-02, 3.03e-03, 2.88e-04]     [1.81e-03, 2.39e-02, 3.03e-03, 2.88e-04]     []
17000     [1.10e-03, 1.83e-02, 2.49e-03, 2.79e-04]     [2.13e-03, 1.83e-02, 2.49e-03, 2.79e-04]     []
18000     [2.09e-03, 1.16e-02, 2.12e-03, 2.95e-04]     [2.88e-03, 1.16e-02, 2.12e-03, 2.95e-04]     []
19000     [7.67e-04, 6.60e-03, 1.88e-03, 1.23e-04]     [1.77e-03, 6.60e-03, 1.88e-03, 1.23e-04]     []
20000     [6.75e-04, 3.84e-03, 1.41e-03, 8.88e-05]     [1.52e-03, 3.84e-03, 1.41e-03, 8.88e-05]     []
21000     [7.55e-04, 2.37e-03, 9.76e-04, 7.21e-05]     [1.37e-03, 2.37e-03, 9.76e-04, 7.21e-05]     []
22000     [4.54e-04, 1.56e-03, 5.88e-04, 6.19e-05]     [9.74e-04, 1.56e-03, 5.88e-04, 6.19e-05]     []
23000     [1.25e-03, 1.06e-03, 3.61e-04, 1.71e-04]     [1.70e-03, 1.06e-03, 3.61e-04, 1.71e-04]     []
24000     [2.85e-04, 7.03e-04, 2.29e-04, 5.34e-05]     [6.21e-04, 7.03e-04, 2.29e-04, 5.34e-05]     []
25000     [2.25e-04, 4.81e-04, 1.51e-04, 4.92e-05]     [4.99e-04, 4.81e-04, 1.51e-04, 4.92e-05]     []
26000     [1.87e-04, 3.38e-04, 1.09e-04, 4.52e-05]     [4.15e-04, 3.38e-04, 1.09e-04, 4.52e-05]     []
27000     [1.79e-04, 2.40e-04, 9.07e-05, 4.20e-05]     [3.67e-04, 2.40e-04, 9.07e-05, 4.20e-05]     []
```