HW5_PartA_starter_code

March 19, 2024

[1]:
```
# import libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns  #for visualization
import torch
import torch.nn as nn
from mpl_toolkits.axes_grid1 import make_axes_locatable
np.random.seed(1)
torch.manual_seed(1)
device = torch.device('cuda:0' if torch.cuda.is_available() else 'cpu')
print(device)
```

```
cuda:0
```

[2]:
```
### Instructions for the students using Google Colab
### If you are using Google Colab, you will need to first mount the google␣
 ↪drive. You can do this by running the following code after uncommenting it.
from google.colab import drive
drive.mount('/content/drive', force_remount=False)

### You also need to upload the data to your Google drive.
### Make sure you are using the same google account for both Google Colab and␣
 ↪Google Drive.

### For example, if the dataset data.npy is located in /content/drive/My Drive/
 ↪CME216/HW5/PartA, then you can use the following code to load the data.
# data_path = '/content/drive/MyDrive/CME216/HW5/PartA'
# data = np.load(f'{data_path}/data.npy')
### OR else.....
### you can cd to the directory where the data is located using the following␣
 ↪command
# %cd /content/drive/MyDrive/CME216/HW5/PartA
### And then load the data. Make sure that the notebook is running in the same␣
 ↪directory where the data is located.
# data = np.load('data.npy')
```

```
Mounted at /content/drive
```

0.1 Q1: Data wrangling

[3]:
```
# load data
from torch.utils.data import TensorDataset, DataLoader

X = np.load("/content/drive/MyDrive/ME343/HW5A/u0_data.npy")
Y = np.load("/content/drive/MyDrive/ME343/HW5A/uT_data.npy")

# split data into training, validation and test sets
X_flatten = X.reshape(len(X), -1)
Y_flatten = Y.reshape(len(Y), -1)

train_split, val_split = int(len(X)*0.8), int(len(X)*0.9)
X_train, X_test, X_validation = X_flatten[:train_split], X_flatten[train_split:
 ↪val_split], X_flatten[val_split:]
Y_train, Y_test, Y_validation = Y_flatten[:train_split], Y_flatten[train_split:
 ↪val_split], Y_flatten[val_split:]

# create corresponding PyTorch tensors
print(X_flatten.dtype)
train_dataset = TensorDataset(torch.from_numpy(X_train), torch.
 ↪from_numpy(Y_train))
validation_dataset = TensorDataset(torch.from_numpy(X_validation), torch.
 ↪from_numpy(Y_validation))
test_dataset = TensorDataset(torch.from_numpy(X_test), torch.from_numpy(Y_test))

# create Pytorch dataloaders
train_loader = DataLoader(train_dataset)
validation_loader = DataLoader(validation_dataset)
test_loader = DataLoader(test_dataset,shuffle=True)
```

```
float64
```

0.2 Q2: Model construction

[8]:
```
# Define a simple fully connected network class with following arguments:
# input_dim: input dimension
# output_dim: output dimension
# n_layers: number of layers (counted as number of hidden layers + 1 output␣
 ↪layer)
# n_units: number of neurons in each layer
# activation: type of activation function (NOTE: there should be no activation␣
 ↪function in the output layer)

class FullyConnectedNetwork(torch.nn.Module):
    def __init__(self, input_dim, output_dim, n_layers, n_units, activation):
```

```
        super(FullyConnectedNetwork, self).__init__()  #we create a temporaty␣
↪object of superclass(parent class), and then call the constructor to␣
↪initialize
        self.layers = torch.nn.ModuleList()
        self.layers.append(torch.nn.Linear(input_dim,n_units))
        for _ in range( n_layers-1):
            self.layers.append(torch.nn.Linear(n_units,n_units))
        self.layers.append(torch.nn.Linear(n_units,output_dim))
        self.act = activation

    def forward(self, x):
        for layer in self.layers[:-1]:  #no activation in last layers
            x = self.act(layer(x))
        x = self.layers[-1](x)
        return x
```

0.3 Helper functions

You may use the following functions in train_and_plot function below

[10]:
```
# function to compute relative L2 error in percentage
def rel_l2_error(pred, true):
    """A helper function to compute the relative L2 error in percentage

    Args:
        pred (torch.Tensor): Predicted values
        true (torch.Tensor): True values

    Returns:
        torch.Tensor: Relative L2 error in percentage
    """
    return (torch.norm(pred - true) / torch.norm(true))*100


# prediction plotting function
def prediction_plots(n_plots, indices, u0, uT, output):
    """A helper function to plot the predictions of the model

    Args:
        n_plots (int): Number of plots to display
        indices (list): List of indices to plot
        u0 (np.array): Initial condition (3D numpy array of shape (n_samples,␣
↪D, D))
        uT (np.array): Target condition (3D numpy array of shape (n_samples, D,␣
↪D))
        output (np.array): Model output (3D numpy array of shape (n_samples, D,␣
↪D))
```

```
    """
    fig, axs = plt.subplots(n_plots, 4, figsize=(20, 5*n_plots))
    for i, idx in enumerate(indices):
        im = axs[i, 0].imshow(u0[idx, :, :], cmap='viridis')
        divider = make_axes_locatable(axs[i, 0])
        cax = divider.append_axes("right", size="5%", pad=0.05)
        fig.colorbar(im, cax=cax, orientation='vertical')
        im = axs[i, 1].imshow(uT[idx, :, :], cmap='viridis')
        divider = make_axes_locatable(axs[i, 1])
        cax = divider.append_axes("right", size="5%", pad=0.05)
        fig.colorbar(im, cax=cax, orientation='vertical')
        im = axs[i, 2].imshow(output[idx, :, :], cmap='viridis')
        divider = make_axes_locatable(axs[i, 2])
        cax = divider.append_axes("right", size="5%", pad=0.05)
        fig.colorbar(im, cax=cax, orientation='vertical')
        im = axs[i, 3].imshow(output[idx, :, :]-uT[idx, :, :], cmap='viridis')
        divider = make_axes_locatable(axs[i, 3])
        cax = divider.append_axes("right", size="5%", pad=0.05)
        fig.colorbar(im, cax=cax, orientation='vertical')
    axs[0, 0].set_title('Initial condition (u(x, 0))')
    axs[0, 1].set_title('Target condition (u(x, T))')
    axs[0, 2].set_title('DNN prediction (u_pred(x, T))')
    axs[0, 3].set_title('Error')
    for ax in axs.flatten():
        ax.axis('off')
    plt.show()
```

0.4 Q3: Training function

[11]:
```
# write train_and_plot function here...

# 10 points. Training routine: Write a function called train and plot which␣
 ↪takes as argument
# following things:
# • model (model - an object instantiation of the class FullyConnectedNetwork)
# • optimizer (type of optimizer for performing gradient updates)
# • max epochs (maximum number of epochs)
# • batch size (number of samples in each batch)
import random
random.seed(42)

def train_and_plot(model, optimizer, max_epoches, batch_size):
    ␣
 ↪print("-------------------------------------------------------------------------------
    print("Current Optimizer is", optimizer.__class__.__name__)
    train_loader = DataLoader(train_dataset, batch_size=batch_size,shuffle=True)
```

```
   validation_loader = DataLoader(validation_dataset,␣
↪batch_size=batch_size,shuffle=True)
   criterion = torch.nn.MSELoss()
   training_loss_list = []
   validation_loss_list =[]
   total_train_loss = 0.0
   model.to(device)
   for epoch in range(max_epoches):
       model.train()
       total_train_loss = 0.0
       for X_batch, Y_batch in  train_loader:
           X_batch, Y_batch = X_batch.to(device).float(), Y_batch.to(device).
↪float()   # change dtype to float32 to match the model
           # print(X_batch.device)
           # print(next(model.parameters()).device)
           #print(X_batch.dtype)
           # print(Y_batch.dtype)
           outputs = model(X_batch)
           training_loss = criterion(outputs,Y_batch)
           total_train_loss += training_loss.item()  #converting tensor with␣
↪single value to float, will raise an error if there's multiple items

           optimizer.zero_grad()
           training_loss.backward()
           optimizer.step()

       average_train_loss = total_train_loss / len(train_loader)

       if (epoch + 1)%100 == 0:
           model.eval()   #ensure that all layers are in inference mode.
           with torch.no_grad():  #temporarily set all reuire_gradient to false
               total_validation_loss = 0.0
               total_RE = 0.0

               for X_batch, Y_batch in validation_loader:
                   X_batch, Y_batch = X_batch.to(device).float(), Y_batch.
↪to(device).float()
                   outputs = model(X_batch)
                   validation_loss = criterion(outputs, Y_batch)
                   total_validation_loss += validation_loss.item()
                   total_RE +=  rel_l2_error(outputs, Y_batch)
               average_valid_loss = total_validation_loss /␣
↪len(validation_loader)


               training_loss_list.append(average_train_loss)
               validation_loss_list.append(average_valid_loss)
```

```
            MRE = total_RE/ len(validation_loader)


            print(f"Current Epoch: {epoch+1}/{max_epoches}") # f stands for␣
↪"formatted string literal"
            print(f"The training loss is {average_train_loss}") #loss␣
↪itself is a tensor
            print(f"The validation loss is {average_valid_loss}") #loss␣
↪itself is a tensor
            print(f"The mean relative error (in percentage) for the␣
↪validation set: {MRE}")

   plt.figure()
   tick_interval = 100
   epoches = range(100, max_epoches + 1, tick_interval)
   plt.plot(epoches, training_loss_list, label = "Training Loss")
   plt.plot(epoches, validation_loss_list, label = "Validation Loss")
   plt.title("Loss history of training and validation test")
   plt.xlabel("Epoch")
   plt.ylabel("Loss value")


   plt.grid()
   plt.legend()
   plt.show()

   n_plots = 10
   random_indices = random.sample(range(len(test_loader)), n_plots)
   test_inputs, test_outputs, predict_outputs = [], [], []   # Python lists␣
↪are versatile and can hold elements of different types
   model.eval()
   with torch.no_grad():
       for input, output in test_loader:
           predict = model(input.to(device).float())
           test_inputs.append(input.cpu().numpy().reshape(26,26))
           test_outputs.append(output.cpu().numpy().reshape(26,26))
           predict_outputs.append(predict.cpu().numpy().reshape(26,26))
   prediction_plots(n_plots, random_indices, np.array(test_inputs), np.
↪array(test_outputs), np.array(predict_outputs))
```

0.5 Q4: Optimizers

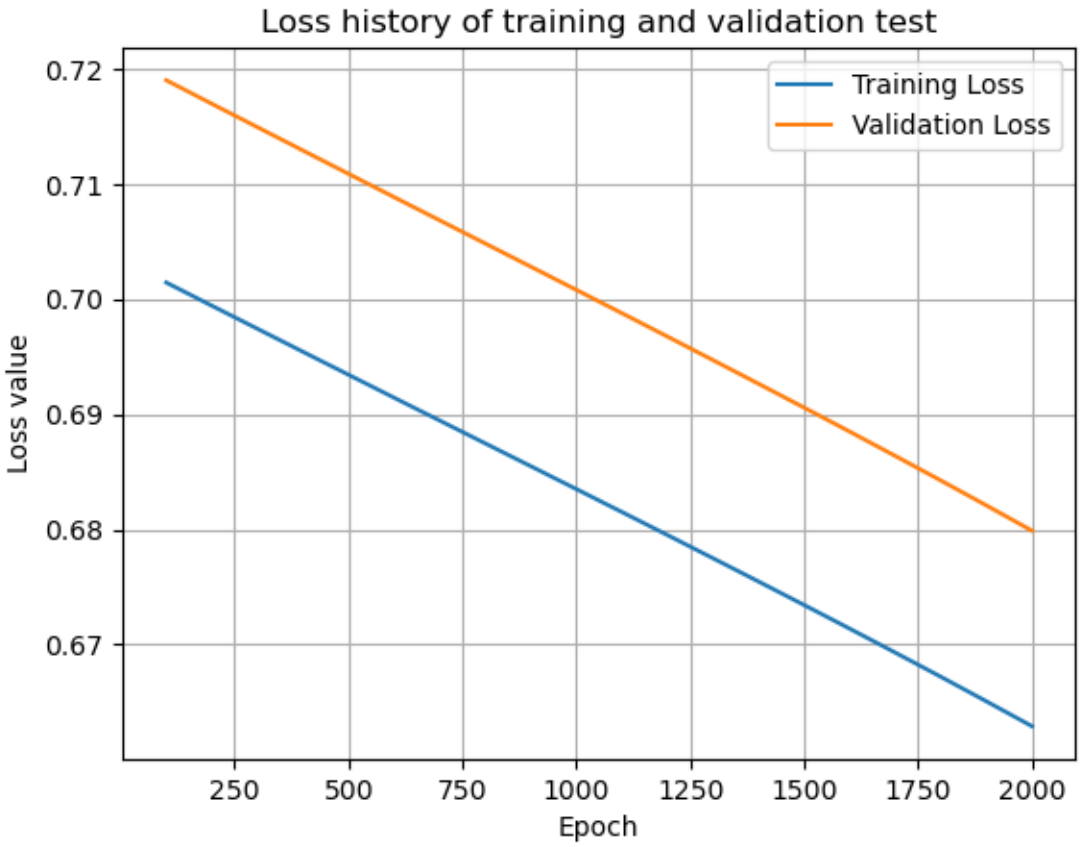
[ ]:
```
# write your code here...
#full batch gradient descent
```

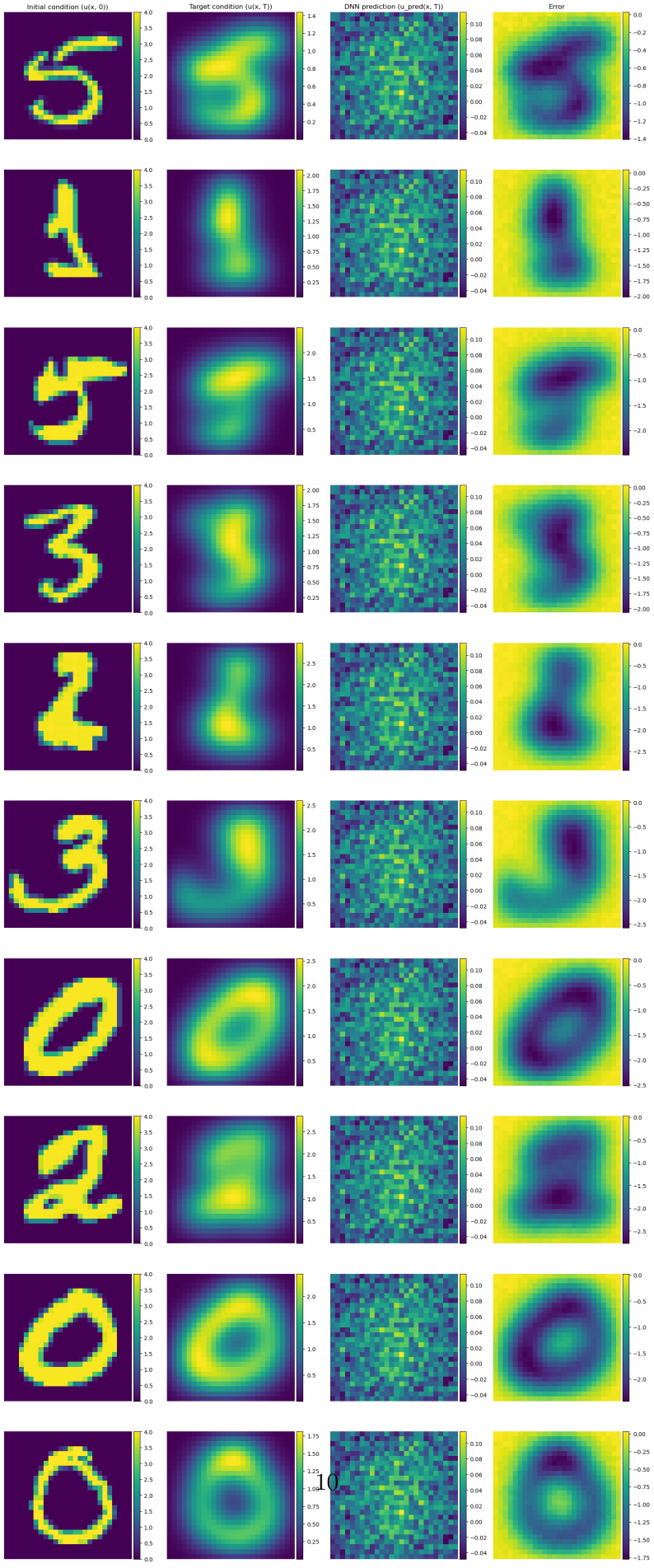
```
model = FullyConnectedNetwork(input_dim=676, output_dim=676, n_layers=8,␣
 ↪n_units=700, activation=torch.nn.ReLU())
optimizer_SGD = torch.optim.SGD(model.parameters(), lr=1e-3)
train_and_plot(model=model, optimizer=optimizer_SGD, max_epoches=2000,␣
 ↪batch_size=2000)
```

```
--------------------------------------------------------------------------------
-------------------------
Current Optimizer is SGD
Current Epoch: 100/2000
The training loss is 0.7014355361461639
The validation loss is 0.7189988493919373
The mean relative error (in percentage) for the validation set:
100.02494812011719
Current Epoch: 200/2000
The training loss is 0.69942307472229
The validation loss is 0.7169565558433533
The mean relative error (in percentage) for the validation set: 99.8827896118164
Current Epoch: 300/2000
The training loss is 0.6974198073148727
The validation loss is 0.7149235010147095
The mean relative error (in percentage) for the validation set:
99.74107360839844
Current Epoch: 400/2000
The training loss is 0.6954229921102524
The validation loss is 0.7128968834877014
The mean relative error (in percentage) for the validation set: 99.599609375
Current Epoch: 500/2000
The training loss is 0.6934282779693604
The validation loss is 0.7108722925186157
The mean relative error (in percentage) for the validation set:
99.45806884765625
Current Epoch: 600/2000
The training loss is 0.6914356350898743
The validation loss is 0.7088499069213867
The mean relative error (in percentage) for the validation set:
99.31649780273438
Current Epoch: 700/2000
The training loss is 0.6894517838954926
The validation loss is 0.7068362832069397
The mean relative error (in percentage) for the validation set:
99.17533111572266
Current Epoch: 800/2000
The training loss is 0.6874700784683228
The validation loss is 0.7048247456550598
The mean relative error (in percentage) for the validation set:
99.03411102294922
```

```
Current Epoch: 900/2000
The training loss is 0.6854856908321381
The validation loss is 0.7028103470802307
The mean relative error (in percentage) for the validation set:
98.89250183105469
Current Epoch: 1000/2000
The training loss is 0.6834994852542877
The validation loss is 0.700793981552124
The mean relative error (in percentage) for the validation set:
98.75053405761719
Current Epoch: 1100/2000
The training loss is 0.6815078854560852
The validation loss is 0.6987720131874084
The mean relative error (in percentage) for the validation set:
98.60796356201172
Current Epoch: 1200/2000
The training loss is 0.6795068681240082
The validation loss is 0.6967403292655945
The mean relative error (in percentage) for the validation set:
98.46451568603516
Current Epoch: 1300/2000
The training loss is 0.6774937957525253
The validation loss is 0.6946962475776672
The mean relative error (in percentage) for the validation set: 98.3199691772461
Current Epoch: 1400/2000
The training loss is 0.6754693686962128
The validation loss is 0.6926406025886536
The mean relative error (in percentage) for the validation set:
98.17439270019531
Current Epoch: 1500/2000
The training loss is 0.6734327226877213
The validation loss is 0.6905723214149475
The mean relative error (in percentage) for the validation set: 98.0277099609375
Current Epoch: 1600/2000
The training loss is 0.6713776737451553
The validation loss is 0.688485324382782
The mean relative error (in percentage) for the validation set:
97.87947082519531
Current Epoch: 1700/2000
The training loss is 0.6693015545606613
The validation loss is 0.6863768696784973
The mean relative error (in percentage) for the validation set:
97.72947692871094
Current Epoch: 1800/2000
The training loss is 0.667200356721878
The validation loss is 0.6842427253723145
The mean relative error (in percentage) for the validation set:
97.57743072509766
```

```
Current Epoch: 1900/2000
The training loss is 0.6650702506303787
The validation loss is 0.6820790767669678
The mean relative error (in percentage) for the validation set:
97.42301940917969
Current Epoch: 2000/2000
The training loss is 0.6629088670015335
The validation loss is 0.6798837780952454
The mean relative error (in percentage) for the validation set:
97.26612091064453
```



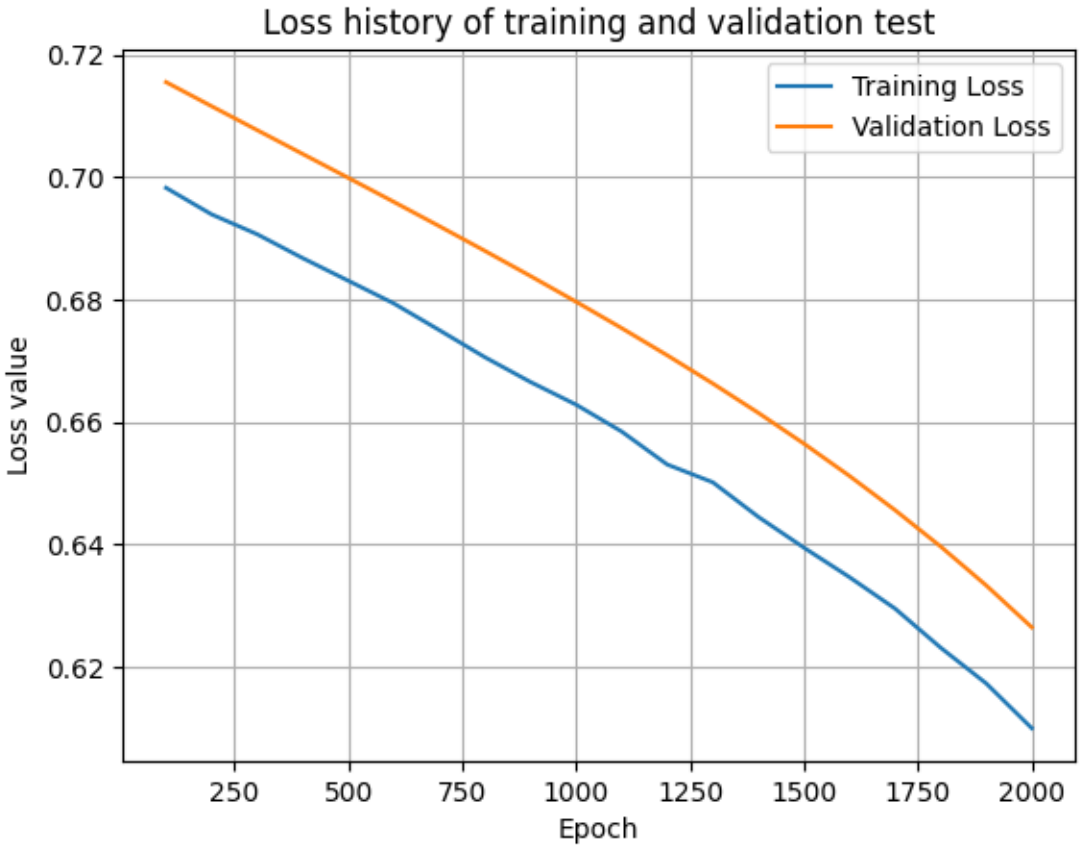

Initial condition (u(x, 0))
Target condition (u(x, T))
DNN prediction (u_pred(x, T))
Error

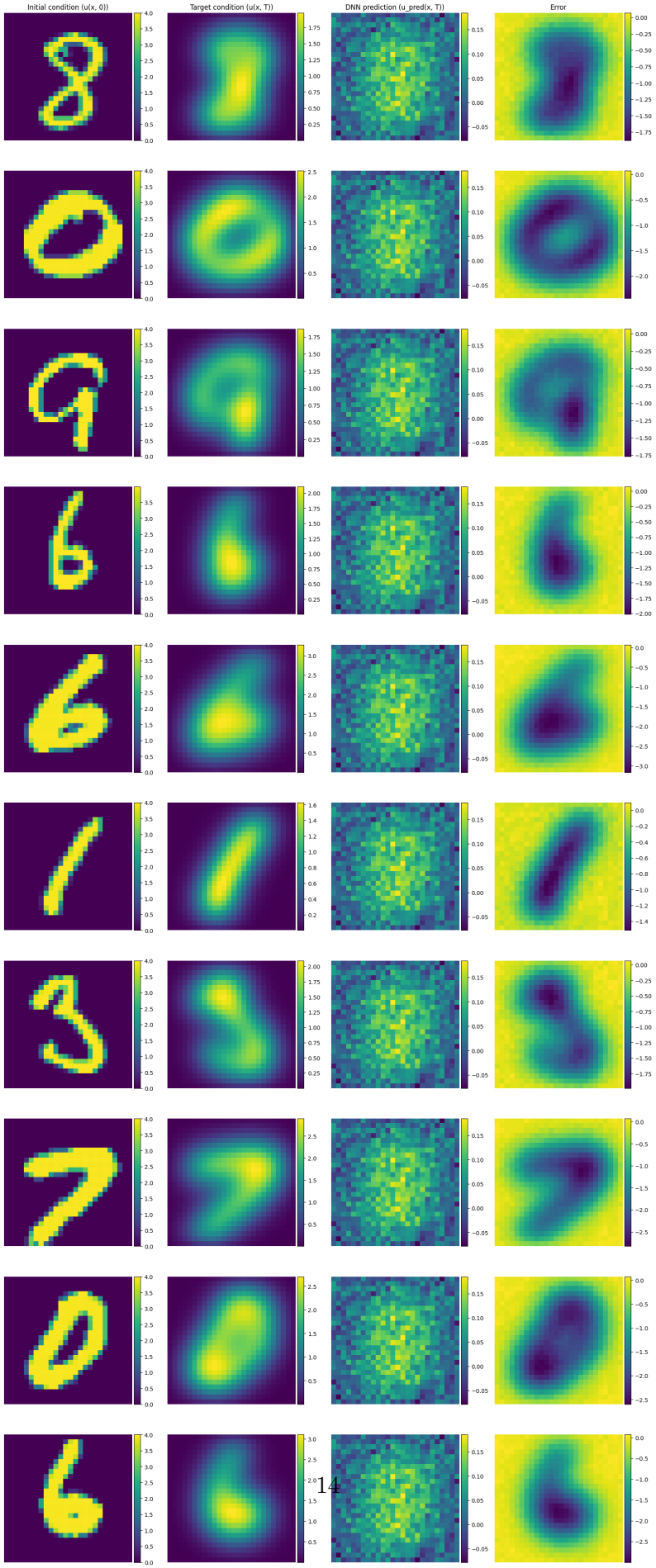
[16]:
```
#minBatch
model = FullyConnectedNetwork(input_dim=676, output_dim=676, n_layers=8,␣
 ↪n_units=700, activation=torch.nn.ReLU())
optimizer_minBatch = torch.optim.SGD(model.parameters(), lr=1e-3)
train_and_plot(model=model, optimizer=optimizer_minBatch, max_epoches=2000,␣
 ↪batch_size=1024)
```

```
--------------------------------------------------------------------------------
-------------------------
Current Optimizer is SGD
Current Epoch: 100/2000
The training loss is 0.6982346251606941
The validation loss is 0.7155052423477173
The mean relative error (in percentage) for the validation set:
99.78164672851562
Current Epoch: 200/2000
The training loss is 0.693898394703865
The validation loss is 0.7115699648857117
The mean relative error (in percentage) for the validation set: 99.5068588256836
Current Epoch: 300/2000
The training loss is 0.6906497851014137
The validation loss is 0.7076614499092102
The mean relative error (in percentage) for the validation set:
99.23320770263672
Current Epoch: 400/2000
The training loss is 0.6867486536502838
The validation loss is 0.7037649750709534
The mean relative error (in percentage) for the validation set:
98.95964050292969
Current Epoch: 500/2000
The training loss is 0.6830547600984573
The validation loss is 0.6998642086982727
The mean relative error (in percentage) for the validation set:
98.68500518798828
Current Epoch: 600/2000
The training loss is 0.6793449968099594
The validation loss is 0.6959326863288879
The mean relative error (in percentage) for the validation set:
98.40742492675781
Current Epoch: 700/2000
The training loss is 0.6749750375747681
The validation loss is 0.6919595003128052
The mean relative error (in percentage) for the validation set: 98.1261215209961
Current Epoch: 800/2000
The training loss is 0.670551523566246
```

```
The validation loss is 0.6879255175590515
The mean relative error (in percentage) for the validation set:
97.83966827392578
Current Epoch: 900/2000
The training loss is 0.6665068939328194
The validation loss is 0.6838164925575256
The mean relative error (in percentage) for the validation set:
97.54702758789062
Current Epoch: 1000/2000
The training loss is 0.6628090068697929
The validation loss is 0.679617702960968
The mean relative error (in percentage) for the validation set:
97.24708557128906
Current Epoch: 1100/2000
The training loss is 0.6584173068404198
The validation loss is 0.6753119230270386
The mean relative error (in percentage) for the validation set:
96.93853759765625
Current Epoch: 1200/2000
The training loss is 0.6530292481184006
The validation loss is 0.6708711385726929
The mean relative error (in percentage) for the validation set:
96.61927795410156
Current Epoch: 1300/2000
The training loss is 0.6501256451010704
The validation loss is 0.666264533996582
The mean relative error (in percentage) for the validation set:
96.28697967529297
Current Epoch: 1400/2000
The training loss is 0.6444696858525276
The validation loss is 0.6614631414413452
The mean relative error (in percentage) for the validation set:
95.93942260742188
Current Epoch: 1500/2000
The training loss is 0.6394447162747383
The validation loss is 0.6564408540725708
The mean relative error (in percentage) for the validation set:
95.57450103759766
Current Epoch: 1600/2000
The training loss is 0.6346257030963898
The validation loss is 0.6511608958244324
The mean relative error (in percentage) for the validation set:
95.18936157226562
Current Epoch: 1700/2000
The training loss is 0.6294907778501511
The validation loss is 0.6455761194229126
The mean relative error (in percentage) for the validation set:
94.78028106689453
```

```
Current Epoch: 1800/2000
The training loss is 0.6231073662638664
The validation loss is 0.6396350264549255
The mean relative error (in percentage) for the validation set:
94.34314727783203
Current Epoch: 1900/2000
The training loss is 0.6172714680433273
The validation loss is 0.6332796812057495
The mean relative error (in percentage) for the validation set:
93.87328338623047
Current Epoch: 2000/2000
The training loss is 0.6099510714411736
The validation loss is 0.6264411211013794
The mean relative error (in percentage) for the validation set:
93.36505889892578
```



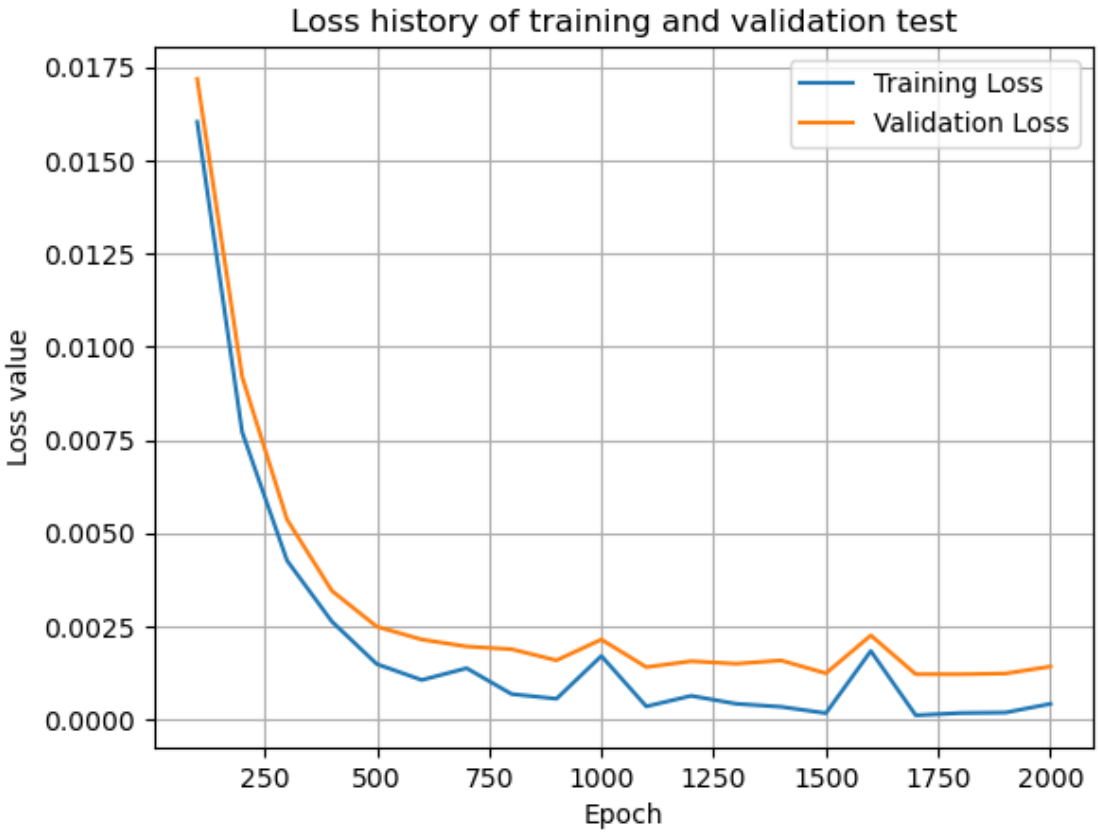

Initial condition (u(x, 0))
Target condition (u(x, T))
DNN prediction (u_pred(x, T))
Error

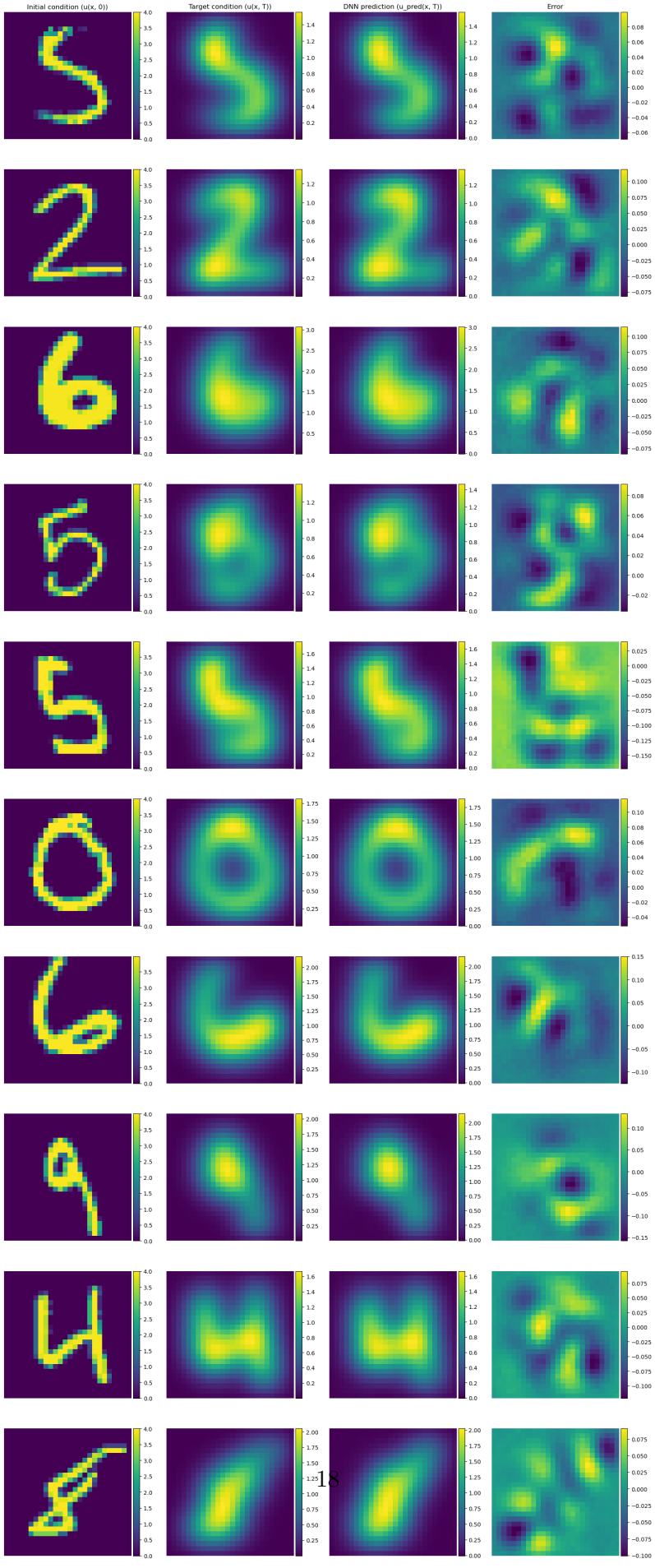
[ ]:
```
#Adam
model = FullyConnectedNetwork(input_dim=676, output_dim=676, n_layers=8,␣
 ↪n_units=700, activation=torch.nn.ReLU())
optimizer_Adam = torch.optim.Adam(model.parameters(), lr=1e-3)
train_and_plot(model=model, optimizer=optimizer_Adam, max_epoches=2000,␣
 ↪batch_size=1024)
```

```
--------------------------------------------------------------------------------
-------------------------
Current Optimizer is Adam
Current Epoch: 100/2000
The training loss is 0.016035154811106622
The validation loss is 0.01718681864440441
The mean relative error (in percentage) for the validation set:
15.46471881866455
Current Epoch: 200/2000
The training loss is 0.007713630213402212
The validation loss is 0.009194821119308472
The mean relative error (in percentage) for the validation set:
11.31138801574707
Current Epoch: 300/2000
The training loss is 0.004265656927600503
The validation loss is 0.005366646684706211
The mean relative error (in percentage) for the validation set:
8.641632080078125
Current Epoch: 400/2000
The training loss is 0.002628024492878467
The validation loss is 0.003452034667134285
The mean relative error (in percentage) for the validation set:
6.930775165557861
Current Epoch: 500/2000
The training loss is 0.0014880167436785996
The validation loss is 0.0024937435518950224
The mean relative error (in percentage) for the validation set:
5.890743732452393
Current Epoch: 600/2000
The training loss is 0.0010618141677696258
The validation loss is 0.00214733206667006
The mean relative error (in percentage) for the validation set:
5.466305255889893
Current Epoch: 700/2000
The training loss is 0.001378260159981437
The validation loss is 0.001960654044523835
The mean relative error (in percentage) for the validation set:
5.223297119140625
```

```
Current Epoch: 800/2000
The training loss is 0.0006845486495876685
The validation loss is 0.0018875467358157039
The mean relative error (in percentage) for the validation set:
5.124990463256836
Current Epoch: 900/2000
The training loss is 0.0005584522659773938
The validation loss is 0.0015891442308202386
The mean relative error (in percentage) for the validation set:
4.702468395233154
Current Epoch: 1000/2000
The training loss is 0.0017064106432371773
The validation loss is 0.002148296218365431
The mean relative error (in percentage) for the validation set:
5.467532157897949
Current Epoch: 1100/2000
The training loss is 0.0003537551747285761
The validation loss is 0.0014070449396967888
The mean relative error (in percentage) for the validation set:
4.424846649169922
Current Epoch: 1200/2000
The training loss is 0.0006373593387252185
The validation loss is 0.001565992017276585
The mean relative error (in percentage) for the validation set:
4.668087959289551
Current Epoch: 1300/2000
The training loss is 0.000423880985181313
The validation loss is 0.0014986606547608972
The mean relative error (in percentage) for the validation set:
4.566631317138672
Current Epoch: 1400/2000
The training loss is 0.0003422456720727496
The validation loss is 0.0015864612068980932
The mean relative error (in percentage) for the validation set:
4.6984968185424805
Current Epoch: 1500/2000
The training loss is 0.00017176096116600093
The validation loss is 0.001241418649442494
The mean relative error (in percentage) for the validation set:
4.156266689300537
Current Epoch: 1600/2000
The training loss is 0.0018383565911790356
The validation loss is 0.002257952932268381
The mean relative error (in percentage) for the validation set:
5.605337142944336
Current Epoch: 1700/2000
The training loss is 0.00011557639572856715
The validation loss is 0.0012179752811789513
```

```
The mean relative error (in percentage) for the validation set:
4.116835594177246
Current Epoch: 1800/2000
The training loss is 0.00017512101476313546
The validation loss is 0.0012161090271547437
The mean relative error (in percentage) for the validation set:
4.113679885864258
Current Epoch: 1900/2000
The training loss is 0.00018862172692024615
The validation loss is 0.001235411036759615
The mean relative error (in percentage) for the validation set:
4.146197319030762
Current Epoch: 2000/2000
The training loss is 0.0004223084542900324
The validation loss is 0.0014230416854843497
The mean relative error (in percentage) for the validation set:
4.4499287605285645
```



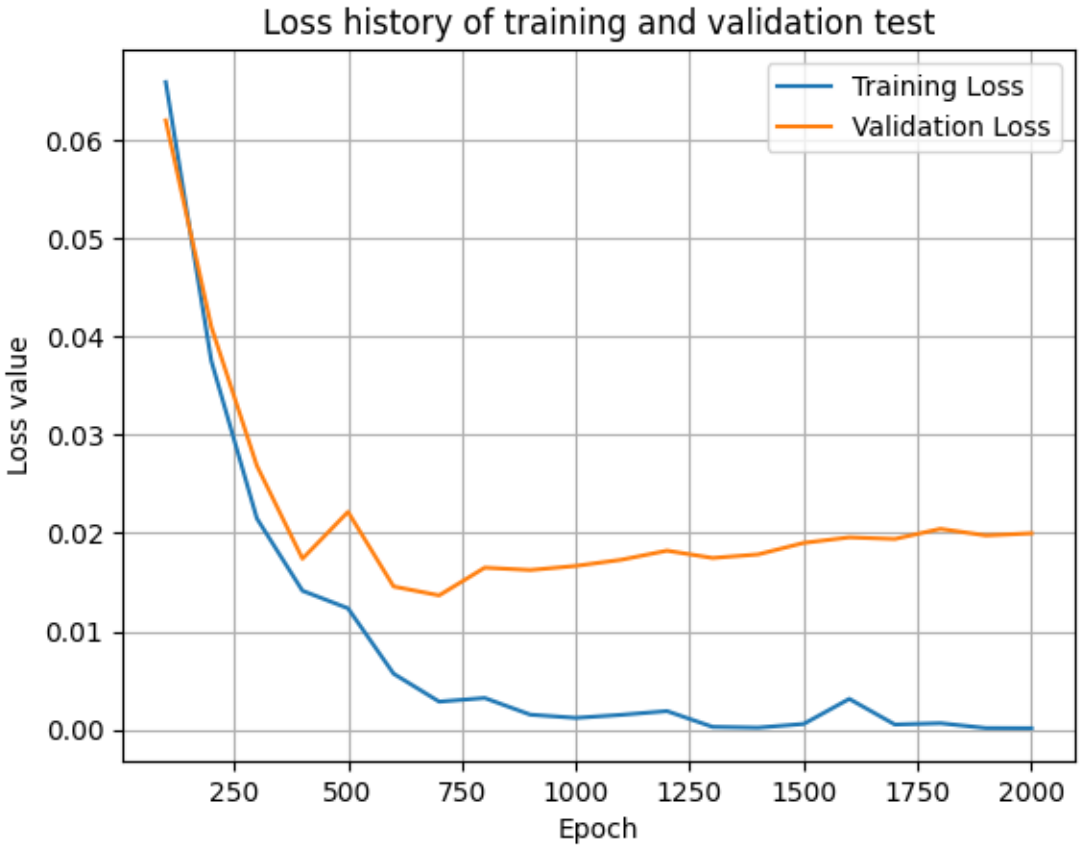

Initial condition (u(x, 0))
Target condition (u(x, T))
DNN prediction (u_pred(x, T))
Error

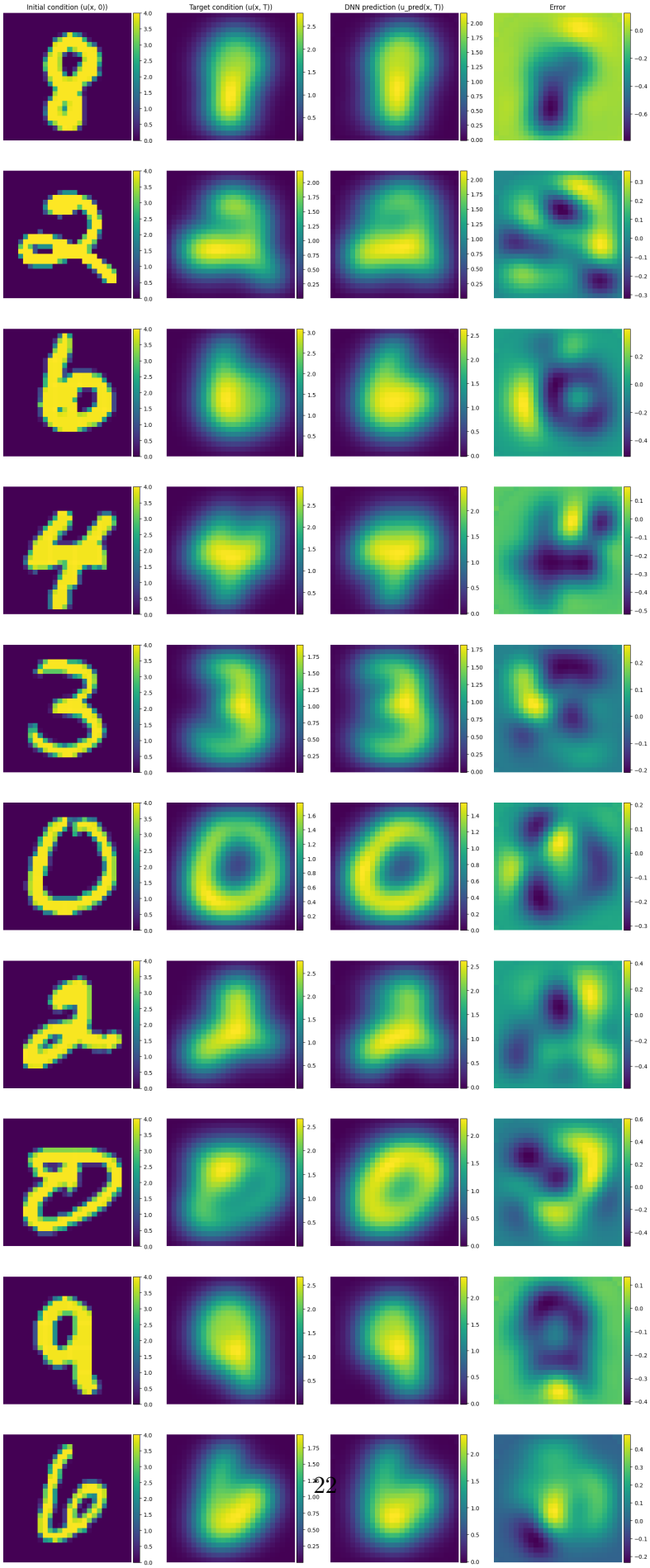
[12]:
```
#RMSprop
optimizer_RMSprop = torch.optim.RMSprop(model.parameters(), lr=1e-3)
train_and_plot(model=model, optimizer=optimizer_RMSprop, max_epoches=2000,␣
 ↪batch_size=1024)
```

```
--------------------------------------------------------------------------------
-------------------------
Current Optimizer is RMSprop
Current Epoch: 100/2000
The training loss is 0.06592527637258172
The validation loss is 0.06202807277441025
The mean relative error (in percentage) for the validation set:
29.379093170166016
Current Epoch: 200/2000
The training loss is 0.03754484001547098
The validation loss is 0.04100668430328369
The mean relative error (in percentage) for the validation set:
23.887548446655273
Current Epoch: 300/2000
The training loss is 0.021497168578207493
The validation loss is 0.026861077174544334
The mean relative error (in percentage) for the validation set:
19.33329963684082
Current Epoch: 400/2000
The training loss is 0.014135089586488903
The validation loss is 0.01739717647433281
The mean relative error (in percentage) for the validation set:
15.55907154083252
Current Epoch: 500/2000
The training loss is 0.012347820913419127
The validation loss is 0.022167064249515533
The mean relative error (in percentage) for the validation set:
17.562984466552734
Current Epoch: 600/2000
The training loss is 0.005688605597242713
The validation loss is 0.014568558894097805
The mean relative error (in percentage) for the validation set:
14.238116264343262
Current Epoch: 700/2000
The training loss is 0.0028448639786802232
The validation loss is 0.013662341982126236
The mean relative error (in percentage) for the validation set:
13.788177490234375
Current Epoch: 800/2000
The training loss is 0.00321504840394482
```

```
The validation loss is 0.01648375764489174
The mean relative error (in percentage) for the validation set:
15.145108222961426
Current Epoch: 900/2000
The training loss is 0.0015179515321506187
The validation loss is 0.01624458283185959
The mean relative error (in percentage) for the validation set:
15.034830093383789
Current Epoch: 1000/2000
The training loss is 0.0011912226618733257
The validation loss is 0.01666153222322464
The mean relative error (in percentage) for the validation set:
15.226557731628418
Current Epoch: 1100/2000
The training loss is 0.0015016532561276108
The validation loss is 0.01729205995798111
The mean relative error (in percentage) for the validation set:
15.511994361877441
Current Epoch: 1200/2000
The training loss is 0.0018826353625627235
The validation loss is 0.01821596547961235
The mean relative error (in percentage) for the validation set:
15.920999526977539
Current Epoch: 1300/2000
The training loss is 0.0002751839092525188
The validation loss is 0.017471706494688988
The mean relative error (in percentage) for the validation set:
15.592363357543945
Current Epoch: 1400/2000
The training loss is 0.00019874886311299633
The validation loss is 0.01783457025885582
The mean relative error (in percentage) for the validation set:
15.753446578979492
Current Epoch: 1500/2000
The training loss is 0.0005736646089644637
The validation loss is 0.019007273018360138
The mean relative error (in percentage) for the validation set:
16.26313018798828
Current Epoch: 1600/2000
The training loss is 0.003124711394775659
The validation loss is 0.019570790231227875
The mean relative error (in percentage) for the validation set:
16.502450942993164
Current Epoch: 1700/2000
The training loss is 0.0005140332359587774
The validation loss is 0.019406024366617203
The mean relative error (in percentage) for the validation set:
16.432838439941406
```

```
Current Epoch: 1800/2000
The training loss is 0.0006470260341302492
The validation loss is 0.020435599610209465
The mean relative error (in percentage) for the validation set:
16.86311912536621
Current Epoch: 1900/2000
The training loss is 0.0001534410966996802
The validation loss is 0.019763678312301636
The mean relative error (in percentage) for the validation set:
16.583574295043945
Current Epoch: 2000/2000
The training loss is 0.00013720081096835202
The validation loss is 0.01998259872198105
The mean relative error (in percentage) for the validation set:
16.675168991088867
```




Initial condition (u(x, 0))
Target condition (u(x, T))
DNN prediction (u_pred(x, T))
Error

Use this text cell to: comment on your observations. Which optimizer performs best? Which criteria are you using to judge the relative performance of different optimizers and reach at your conclusion.

When we look at the four optimizers, full batch and mini batch didn't do well in predicting test results as yielding blurring images. Full batch is even worse than mini batch considering its even higher losses, making them the less favorable options in this problem.

On the other hand side, Adam and RMSProp are better at predicting test outcomes. Both success to generate a result which is close to the ground trut In addition, Adam shows a clear and stable improvement, and its loss is much smaller than that of RMSProp. It it probably because Adam not only maintainis a moving average of the squared gradients, but also keeps an exponentially decaying average of past gradients similar to momentum. Combining these advantages make it the most effective optimizer among the four.

In a word, Adam is the top choice for our model as it gives a significantly better and more stable results compared to the others.

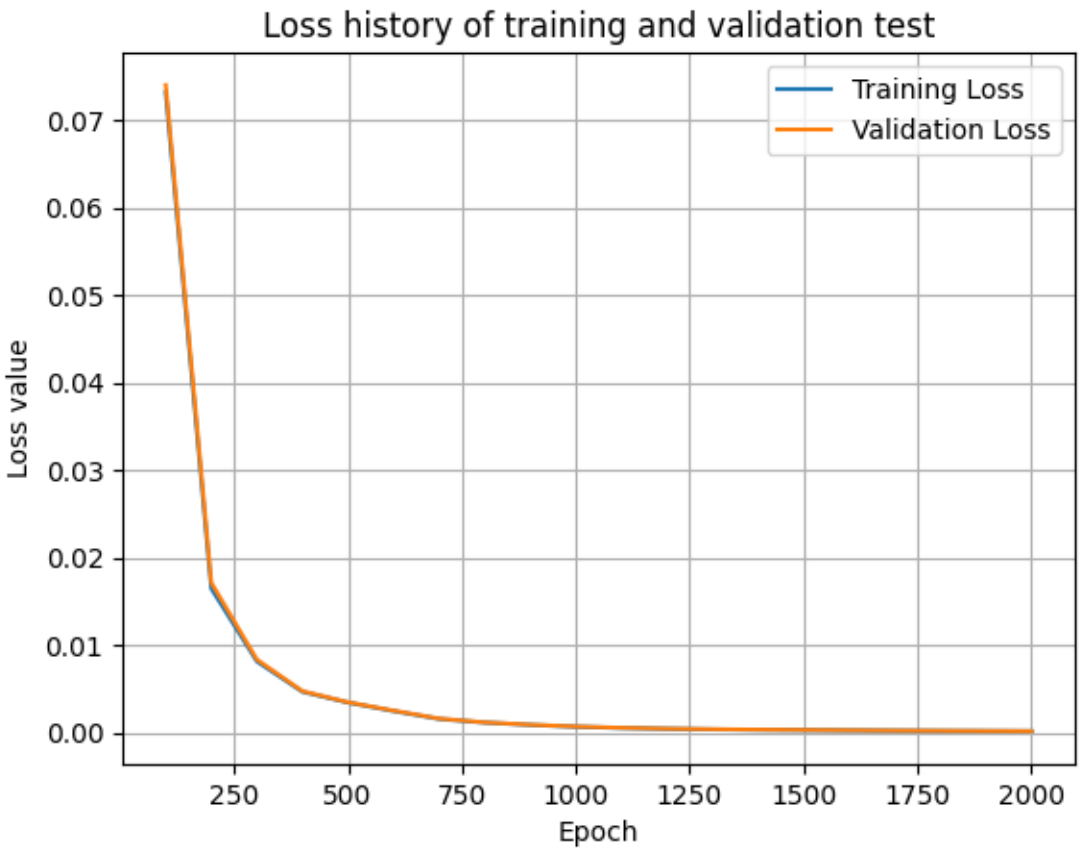
0.6 Q5: Tuning learning rate for the optimal optimizer (learning_rate=[1e-5, 1e-2, 1])

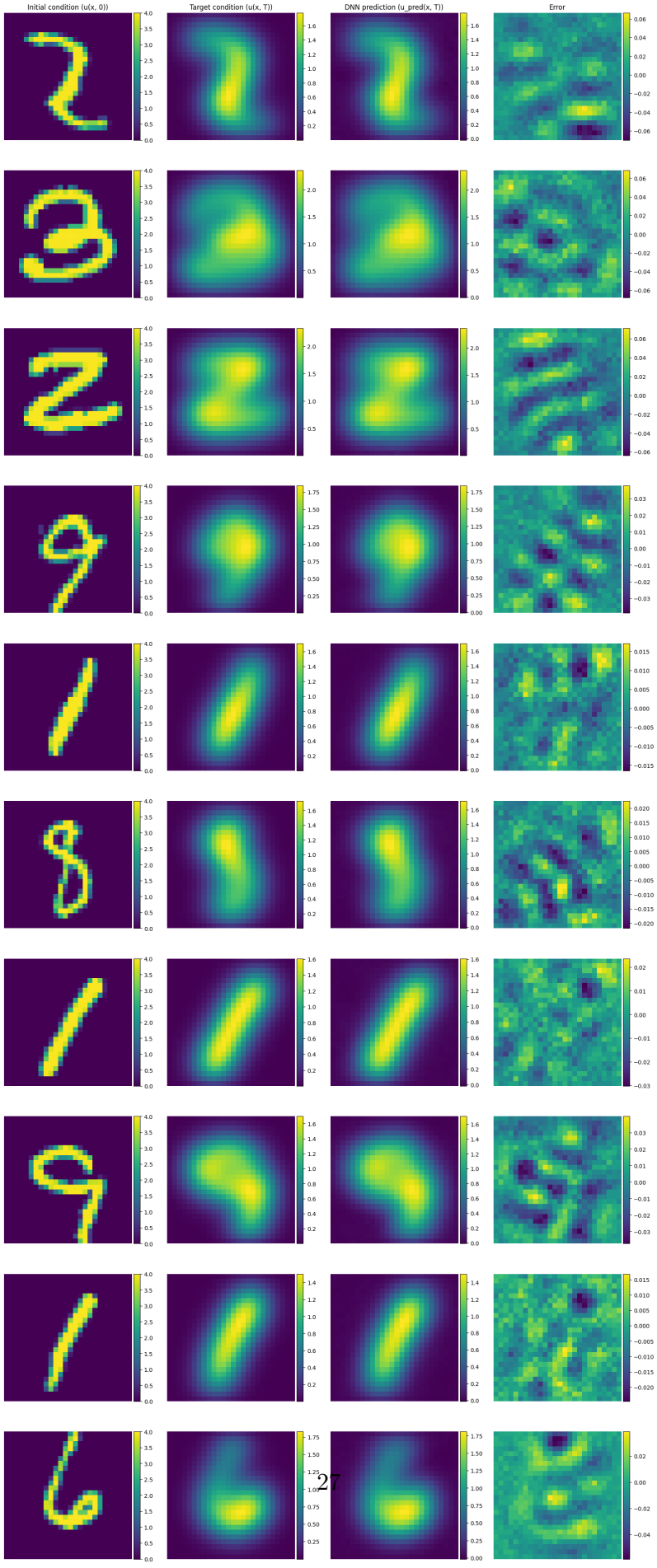
[17]:
```
# write your code here.
#learning rate = 1e-5
model = FullyConnectedNetwork(input_dim=676, output_dim=676, n_layers=8,␣
 ↪n_units=700, activation=torch.nn.ReLU())
optimizer_Adam = torch.optim.Adam(model.parameters(), lr=1e-5)
train_and_plot(model=model, optimizer=optimizer_Adam, max_epoches=2000,␣
 ↪batch_size=1024)
```

```
--------------------------------------------------------------------------------
--------------------------
Current Optimizer is Adam
Current Epoch: 100/2000
The training loss is 0.0732037490233779
The validation loss is 0.07402309030294418
The mean relative error (in percentage) for the validation set:
32.094295501708984
Current Epoch: 200/2000
The training loss is 0.016488875960931182
The validation loss is 0.017165424302220345
The mean relative error (in percentage) for the validation set:
15.455089569091797
Current Epoch: 300/2000
The training loss is 0.008167476975359023
The validation loss is 0.008378922007977962
The mean relative error (in percentage) for the validation set:
```

```
10.797876358032227
Current Epoch: 400/2000
The training loss is 0.004695432202424854
The validation loss is 0.004771742038428783
The mean relative error (in percentage) for the validation set:
8.148594856262207
Current Epoch: 500/2000
The training loss is 0.003462780936388299
The validation loss is 0.0034981602802872658
The mean relative error (in percentage) for the validation set:
6.976926326751709
Current Epoch: 600/2000
The training loss is 0.0024923233431763947
The validation loss is 0.002549166092649102
The mean relative error (in percentage) for the validation set:
5.955843925476074
Current Epoch: 700/2000
The training loss is 0.0015917477576294914
The validation loss is 0.0016314048552885652
The mean relative error (in percentage) for the validation set:
4.764585971832275
Current Epoch: 800/2000
The training loss is 0.0011777512554544955
The validation loss is 0.0011971226194873452
The mean relative error (in percentage) for the validation set:
4.081441402435303
Current Epoch: 900/2000
The training loss is 0.0009105527424253523
The validation loss is 0.0009317250223830342
The mean relative error (in percentage) for the validation set:
3.600709915161133
Current Epoch: 1000/2000
The training loss is 0.0007010106492089108
The validation loss is 0.0007263597217388451
The mean relative error (in percentage) for the validation set:
3.1792168617248535
Current Epoch: 1100/2000
The training loss is 0.0005564140519709326
The validation loss is 0.0005805307300761342
The mean relative error (in percentage) for the validation set:
2.842214584350586
Current Epoch: 1200/2000
The training loss is 0.00046397744517889805
The validation loss is 0.000482636911328882
The mean relative error (in percentage) for the validation set:
2.591519594192505
Current Epoch: 1300/2000
The training loss is 0.00040156741306418553
```

```
The validation loss is 0.0004233557265251875
The mean relative error (in percentage) for the validation set:
2.427152156829834
Current Epoch: 1400/2000
The training loss is 0.0003519557139952667
The validation loss is 0.0003704318078234792
The mean relative error (in percentage) for the validation set:
2.270379066467285
Current Epoch: 1500/2000
The training loss is 0.0003166227907058783
The validation loss is 0.0003316572692710906
The mean relative error (in percentage) for the validation set:
2.148270845413208
Current Epoch: 1600/2000
The training loss is 0.00028115843088016845
The validation loss is 0.00030011171475052834
The mean relative error (in percentage) for the validation set:
2.0435521602630615
Current Epoch: 1700/2000
The training loss is 0.0002507584977138322
The validation loss is 0.00026570053887553513
The mean relative error (in percentage) for the validation set:
1.922828197479248
Current Epoch: 1800/2000
The training loss is 0.00022914254986972082
The validation loss is 0.00024494013632647693
The mean relative error (in percentage) for the validation set:
1.8461809158325195
Current Epoch: 1900/2000
The training loss is 0.00021395164003479294
The validation loss is 0.00022384049952961504
The mean relative error (in percentage) for the validation set:
1.7648736238479614
Current Epoch: 2000/2000
The training loss is 0.0001891307965706801
The validation loss is 0.00020098662935197353
The mean relative error (in percentage) for the validation set: 1.67235267162323
```


Loss history of training and validation test
Training Loss
Validation Loss
Loss value
0.07
0.06
0.05
0.04
0.03
0.02
0.01
0.00
250
500
750
1000
1250
1500
1750
2000
Epoch

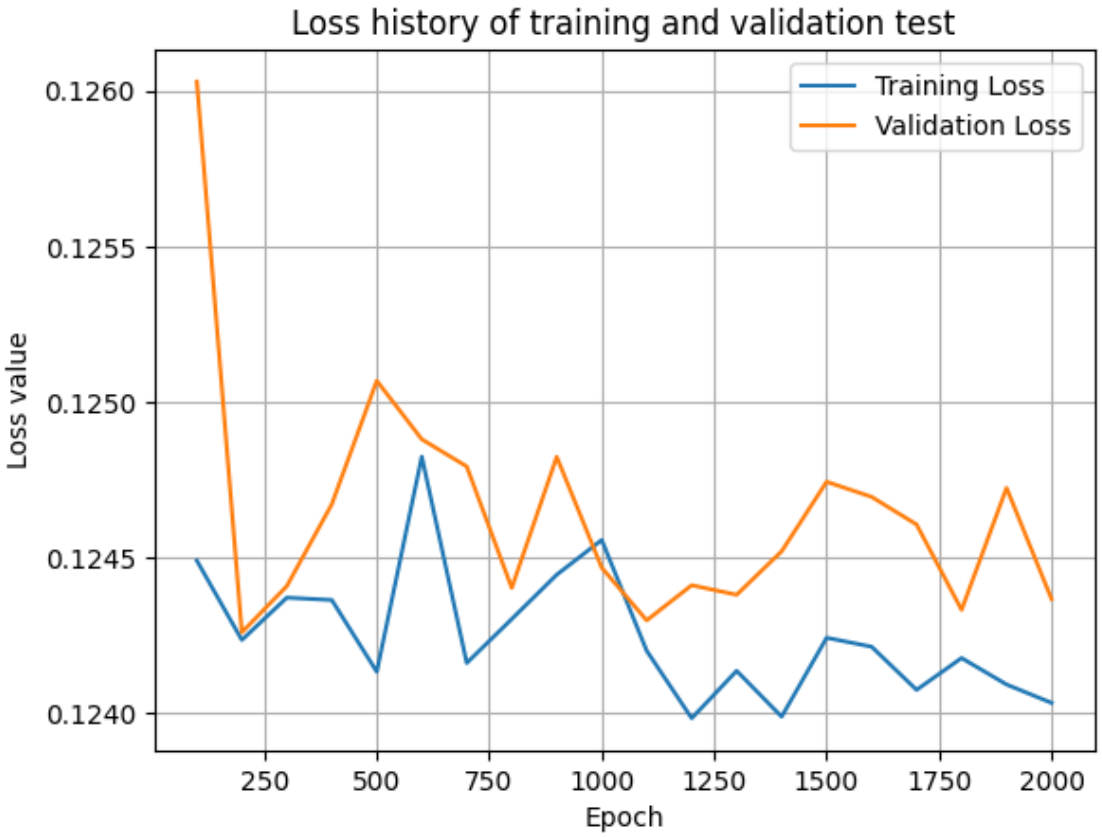

Initial condition (u(x, 0))
Target condition (u(x, T))
DNN prediction (u_pred(x, T))
Error

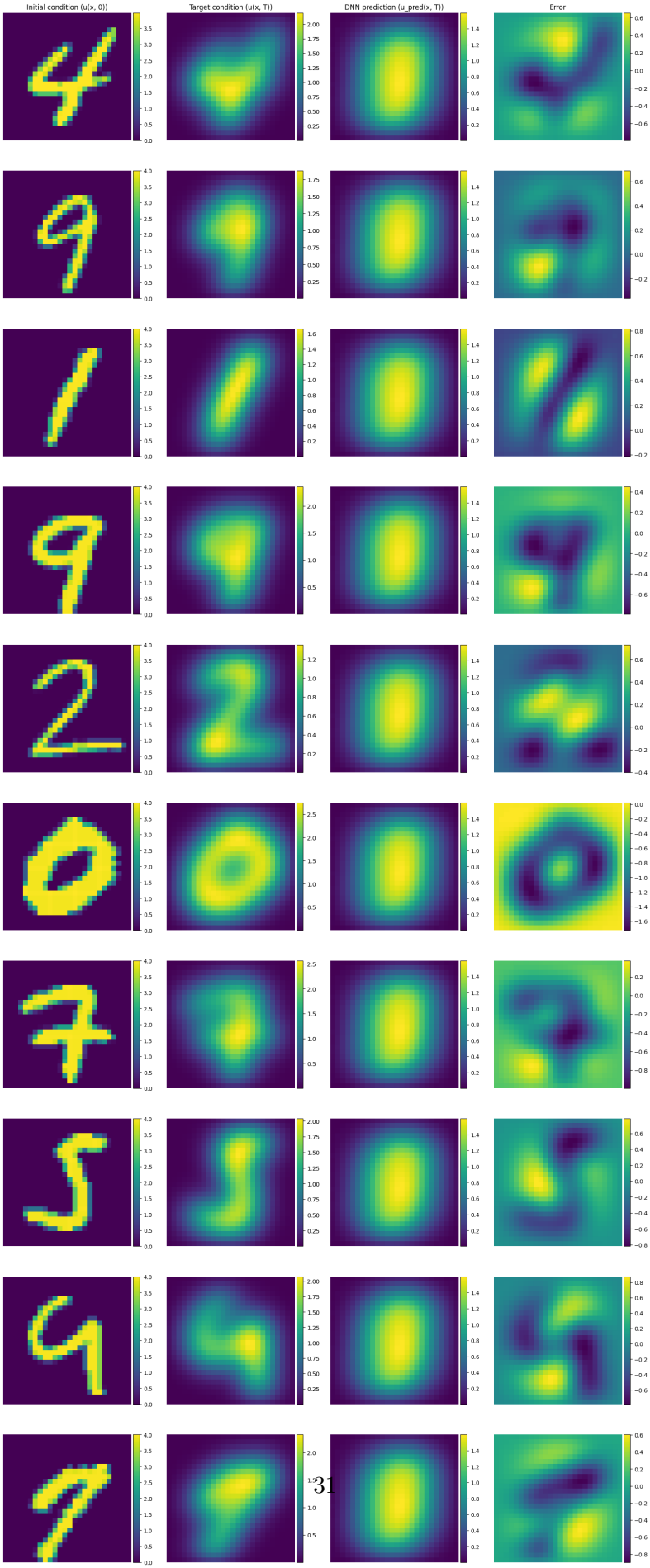
[18]:
```
# write your code here.
#learning rate = 1e-2
model = FullyConnectedNetwork(input_dim=676, output_dim=676, n_layers=8,␣
 ↪n_units=700, activation=torch.nn.ReLU())
optimizer_Adam = torch.optim.Adam(model.parameters(), lr=1e-2)
train_and_plot(model=model, optimizer=optimizer_Adam, max_epoches=2000,␣
 ↪batch_size=1024)
```

```
--------------------------------------------------------------------------------
--------------------------
Current Optimizer is Adam
Current Epoch: 100/2000
The training loss is 0.1244893791154027
The validation loss is 0.12602907419204712
The mean relative error (in percentage) for the validation set:
41.87739181518555
Current Epoch: 200/2000
The training loss is 0.12423437274992466
The validation loss is 0.12425895780324936
The mean relative error (in percentage) for the validation set:
41.58226013183594
Current Epoch: 300/2000
The training loss is 0.12437046505510807
The validation loss is 0.12440760433673859
The mean relative error (in percentage) for the validation set:
41.60712814331055
Current Epoch: 400/2000
The training loss is 0.12436282727867365
The validation loss is 0.1246708557009697
The mean relative error (in percentage) for the validation set:
41.651126861572266
Current Epoch: 500/2000
The training loss is 0.12413218803703785
The validation loss is 0.12506747245788574
The mean relative error (in percentage) for the validation set:
41.717323303222656
Current Epoch: 600/2000
The training loss is 0.12482384499162436
The validation loss is 0.1248796209692955
The mean relative error (in percentage) for the validation set:
41.68598175048828
Current Epoch: 700/2000
The training loss is 0.12416043598204851
The validation loss is 0.12479200214147568
The mean relative error (in percentage) for the validation set:
```

```
41.67135238647461
Current Epoch: 800/2000
The training loss is 0.12430169619619846
The validation loss is 0.12440173327922821
The mean relative error (in percentage) for the validation set:
41.60614776611328
Current Epoch: 900/2000
The training loss is 0.124444087035954
The validation loss is 0.12482304126024246
The mean relative error (in percentage) for the validation set:
41.67654037475586
Current Epoch: 1000/2000
The training loss is 0.1245558699592948
The validation loss is 0.12446676194667816
The mean relative error (in percentage) for the validation set:
41.61701965332031
Current Epoch: 1100/2000
The training loss is 0.1241996344178915
The validation loss is 0.1242973580956459
The mean relative error (in percentage) for the validation set:
41.588687896728516
Current Epoch: 1200/2000
The training loss is 0.1239828085526824
The validation loss is 0.12441037595272064
The mean relative error (in percentage) for the validation set:
41.60758972167969
Current Epoch: 1300/2000
The training loss is 0.12413513846695423
The validation loss is 0.12437976151704788
The mean relative error (in percentage) for the validation set:
41.60247039794922
Current Epoch: 1400/2000
The training loss is 0.12398776970803738
The validation loss is 0.12451941519975662
The mean relative error (in percentage) for the validation set:
41.62582015991211
Current Epoch: 1500/2000
The training loss is 0.12424139026552439
The validation loss is 0.12474294751882553
The mean relative error (in percentage) for the validation set:
41.66316604614258
Current Epoch: 1600/2000
The training loss is 0.12421269807964563
The validation loss is 0.12469441443681717
The mean relative error (in percentage) for the validation set:
41.65505599975586
Current Epoch: 1700/2000
The training loss is 0.12407391890883446
```

```
The validation loss is 0.12460561096668243
The mean relative error (in percentage) for the validation set:
41.640228271484375
Current Epoch: 1800/2000
The training loss is 0.1241763811558485
The validation loss is 0.12433124333620071
The mean relative error (in percentage) for the validation set:
41.59435272216797
Current Epoch: 1900/2000
The training loss is 0.12409126479178667
The validation loss is 0.12472323328256607
The mean relative error (in percentage) for the validation set:
41.659873962402344
Current Epoch: 2000/2000
The training loss is 0.12403220869600773
The validation loss is 0.12436600774526596
The mean relative error (in percentage) for the validation set:
41.60016632080078
```

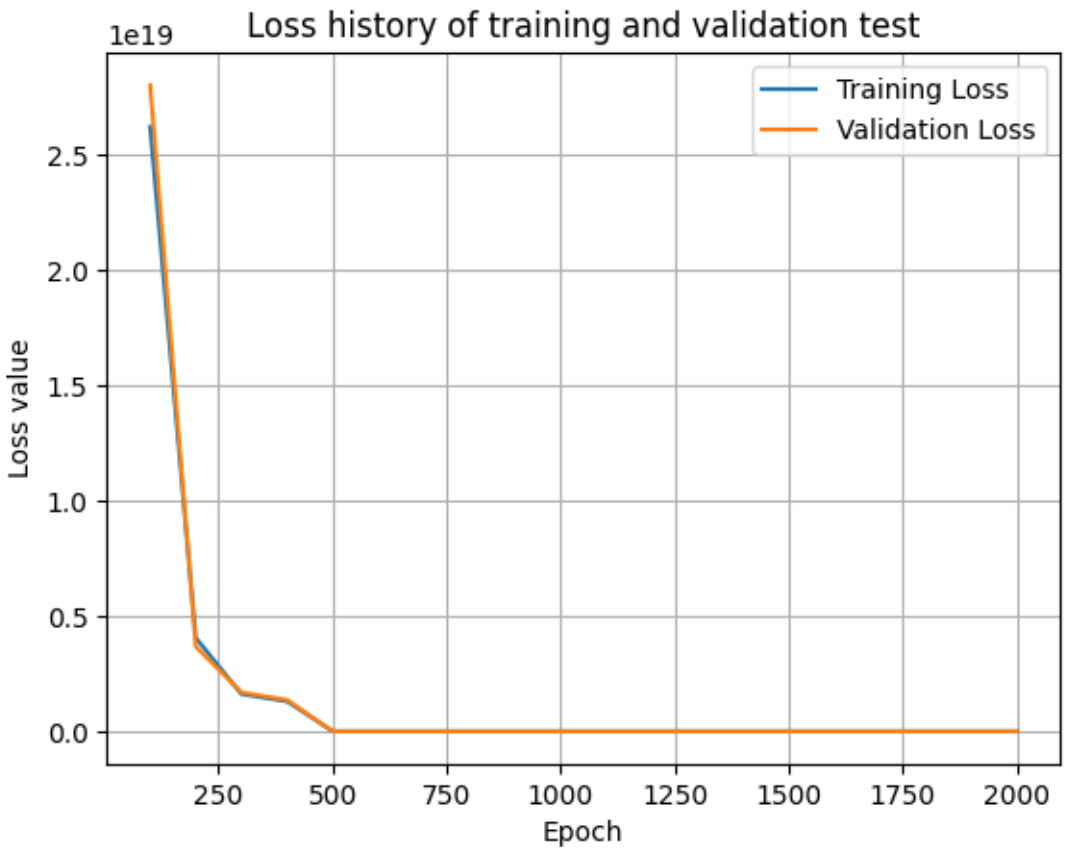


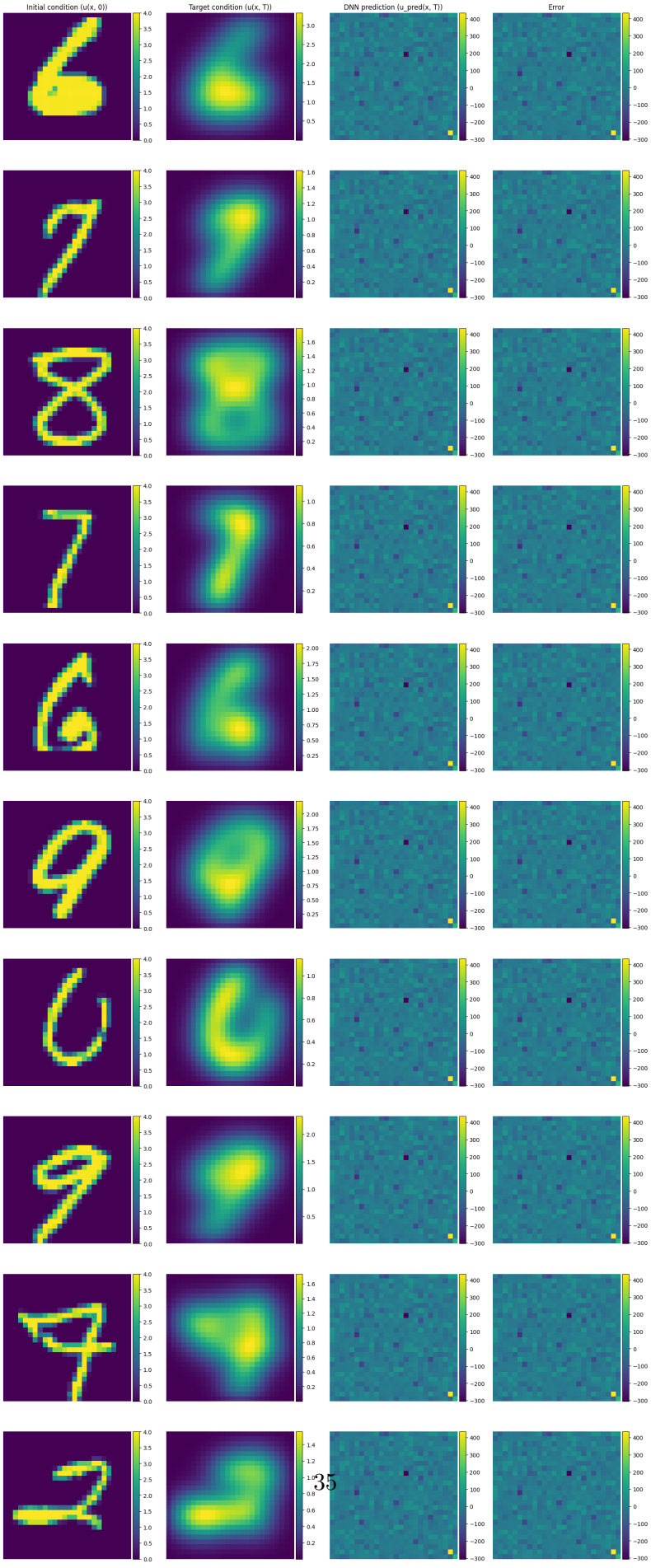

Initial condition (u(x, 0))
Target condition (u(x, T))
DNN prediction (u_pred(x, T))
Error

[19]:
```
# write your code here.
#learning rate = 1e-1
model = FullyConnectedNetwork(input_dim=676, output_dim=676, n_layers=8,␣
 ↪n_units=700, activation=torch.nn.ReLU())
optimizer_Adam = torch.optim.Adam(model.parameters(), lr=1e-1)
train_and_plot(model=model, optimizer=optimizer_Adam, max_epoches=2000,␣
 ↪batch_size=1024)
```

```
--------------------------------------------------------------------------------
--------------------------
Current Optimizer is Adam
Current Epoch: 100/2000
The training loss is 2.6201067420785836e+19
The validation loss is 2.800778052950085e+19
The mean relative error (in percentage) for the validation set: 624285974528.0
Current Epoch: 200/2000
The training loss is 4.0483679691289395e+18
The validation loss is 3.6655293246912594e+18
The mean relative error (in percentage) for the validation set: 225846099968.0
Current Epoch: 300/2000
The training loss is 1.6110097370070385e+18
The validation loss is 1.6825761843547996e+18
The mean relative error (in percentage) for the validation set: 153014157312.0
Current Epoch: 400/2000
The training loss is 1.2950867174007767e+18
The validation loss is 1.3563598804866826e+18
The mean relative error (in percentage) for the validation set: 137382592512.0
Current Epoch: 500/2000
The training loss is 14239713.875
The validation loss is 14191528.0
The mean relative error (in percentage) for the validation set: 444384.4375
Current Epoch: 600/2000
The training loss is 7385130.375
The validation loss is 7355932.0
The mean relative error (in percentage) for the validation set: 319936.0
Current Epoch: 700/2000
The training loss is 3587647.46875
The validation loss is 3573395.75
The mean relative error (in percentage) for the validation set: 222989.703125
Current Epoch: 800/2000
The training loss is 1928045.125
The validation loss is 1922617.875
The mean relative error (in percentage) for the validation set: 163565.15625
Current Epoch: 900/2000
The training loss is 1303476.265625
```

```
The validation loss is 1301199.0
The mean relative error (in percentage) for the validation set: 134560.046875
Current Epoch: 1000/2000
The training loss is 967890.6640625
The validation loss is 966251.5625
The mean relative error (in percentage) for the validation set: 115954.96875
Current Epoch: 1100/2000
The training loss is 705234.4453125
The validation loss is 703938.25
The mean relative error (in percentage) for the validation set: 98971.8125
Current Epoch: 1200/2000
The training loss is 550260.3046875
The validation loss is 549503.5625
The mean relative error (in percentage) for the validation set: 87443.90625
Current Epoch: 1300/2000
The training loss is 423821.54296875
The validation loss is 423166.53125
The mean relative error (in percentage) for the validation set: 76736.1328125
Current Epoch: 1400/2000
The training loss is 317624.8203125
The validation loss is 317085.90625
The mean relative error (in percentage) for the validation set: 66425.1796875
Current Epoch: 1500/2000
The training loss is 229779.185546875
The validation loss is 229324.90625
The mean relative error (in percentage) for the validation set: 56489.77734375
Current Epoch: 1600/2000
The training loss is 153767.43359375
The validation loss is 153365.84375
The mean relative error (in percentage) for the validation set: 46196.44921875
Current Epoch: 1700/2000
The training loss is 87863.375
The validation loss is 87530.46875
The mean relative error (in percentage) for the validation set: 34899.87890625
Current Epoch: 1800/2000
The training loss is 38179.63427734375
The validation loss is 37961.6171875
The mean relative error (in percentage) for the validation set: 22983.5234375
Current Epoch: 1900/2000
The training loss is 10919.94140625
The validation loss is 10827.2744140625
The mean relative error (in percentage) for the validation set: 12274.5
Current Epoch: 2000/2000
The training loss is 1760.8511810302734
The validation loss is 1739.4849853515625
The mean relative error (in percentage) for the validation set: 4919.88134765625
```


Loss history of training and validation test
1e19
Training Loss
Validation Loss
Loss value
2.5
2.0
1.5
1.0
0.5
0.0
250
500
750
1000
1250
1500
1750
2000
Epoch


Initial condition (u(x, 0))
Target condition (u(x, T))
DNN prediction (u_pred(x, T))
Error

Use this text cell to: comment on your observations. Which learning rate performs best? Which criteria are you using to judge the relative performance? Why do you think the specific value of learning rate performed better compared to other two.

When Learning rate = 1e-5: In the graph, both training and validation loss sharply decreasing and then stabilizing at small losses,indicating good convergence. Since both losses decrease in a similar fashion and reach low levels, it suggests that the model is learning effectively without overfitting. Also, The final loss values are very low, suggesting this learning rate is effective for the model.

When Learning rate = 1e-2: In the graph, both the training and validation losses presents fluctuations all the time, indicating the learning is instable. Though the losses show a decreasing trend, such oscillations indicate that this learning rate may be too high, leading to less effective learning compared to first case when Learning rate = 1e-5

When Learning rate = 1e-1: In this case, Though the training and validation losses decrease initially but they remains extremely high, with values in the order of 1e19 initially and 1e5 in the end, This indicates that the learning rate is too high, causing the updates to overshoot the minimum and failing to converge properly.

I'm using the following criteria to judge their performance:

1. Convergence: if the loss smoothly decrease to a low level.
2. Stability: No large and constant fluctuations are observed
3. Fitting: the model is not overfitting, we have a small gap between the training and validation losses.

the 1e-5 learning rate performs best might be because it converges smoothly to a really low-level loss with little fluctuation and overshoot. Also, the validation and training losses are close to each other, meaning we don't have overfit.

[22]:
```
# !pip install nbconvert
# !apt-get install texlive texlive-xetex texlive-latex-extra pandoc
!jupyter nbconvert --to pdf /content/drive/MyDrive/Colab\ Notebooks/HW5
```

```
[NbConvertApp] Converting notebook
/content/drive/MyDrive/ME343/HW5A/HW5_PartA_starter_code.ipynb to pdf
[NbConvertApp] Support files will be in HW5_PartA_starter_code_files/
[NbConvertApp] Making directory ./HW5_PartA_starter_code_files
[NbConvertApp] Making directory ./HW5_PartA_starter_code_files
[NbConvertApp] Making directory ./HW5_PartA_starter_code_files
[NbConvertApp] Making directory ./HW5_PartA_starter_code_files
[NbConvertApp] Making directory ./HW5_PartA_starter_code_files
[NbConvertApp] Making directory ./HW5_PartA_starter_code_files
[NbConvertApp] Making directory ./HW5_PartA_starter_code_files
[NbConvertApp] Making directory ./HW5_PartA_starter_code_files
[NbConvertApp] Writing 71607 bytes to notebook.tex
[NbConvertApp] Building PDF
[NbConvertApp] Running xelatex 3 times: ['xelatex', 'notebook.tex', '-quiet']
```

```
[NbConvertApp] Running bibtex 1 time: ['bibtex', 'notebook']
[NbConvertApp] WARNING | bibtex had problems, most likely because there were no
citations
[NbConvertApp] PDF successfully created
[NbConvertApp] Writing 1411448 bytes to
/content/drive/MyDrive/ME343/HW5A/HW5_PartA_starter_code.pdf
```