# Colab & PyTorch Introduction

## Benefits of using Colab

### 1. Zero configuration & pre-installed environment

Include some commonly used packages in Python:

In [13]:
```
import numpy as np
import pandas
import sklearn
import matplotlib.pyplot as plt
```

Also, the common deep learning framework:

In [2]:
```
import keras
import torch
```

### 2. Free access to GPU

In [3]:
```
gpu_info = !nvidia-smi
gpu_info = '\n'.join(gpu_info)
print(gpu_info)
```

```
Wed Feb 23 16:45:27 2022
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 460.32.03    Driver Version: 460.32.03    CUDA Version: 11.2     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|                               |                      |               MIG M. |
|===============================+======================+======================|
|   0  Tesla P100-PCIE...  Off  | 00000000:00:04.0 Off |                    0 |
| N/A   37C    P0    26W / 250W |      0MiB / 16280MiB |      0%      Default |
|                               |                      |                  N/A |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                                  |
|  GPU   GI   CI        PID   Type   Process name                  GPU Memory |
|        ID   ID                                                   Usage      |
|=============================================================================|
|  No running processes found                                                 |
+-----------------------------------------------------------------------------+
```

In [4]:
```
import torch
print(torch.cuda.is_available())
```

```
True
```

The "cuda" stands for the **Compute Unified Device Architecture**, which is a parallel computing platform and programming model developed by Nvidia that makes using a GPU for general purpose computing.

In Pytorch, if the

```
torch.cuda.is_available()
```

is **True**, then the Pytorch successfully connected to GPU, and we can use GPU to compute and train our models.

### 3. Easy sharing

## Colab Tips:

### 1. Install packages or run commands

Easy to use NLP transformers models: Transformers: https://huggingface.co/transformers/

In [ ]:
```
! pip install transformers
```

```
Collecting transformers
  Downloading transformers-4.10.0-py3-none-any.whl (2.8 MB)
     |████████████████████████████████| 2.8 MB 4.3 MB/s
Collecting sacremoses
  Downloading sacremoses-0.0.45-py3-none-any.whl (895 kB)
     |████████████████████████████████| 895 kB 71.4 MB/s
Requirement already satisfied: tqdm>=4.27 in /usr/local/lib/python3.7/dist-packages (from transf
ormers) (4.62.0)
Requirement already satisfied: packaging in /usr/local/lib/python3.7/dist-packages (from transfo
rmers) (21.0)
Requirement already satisfied: numpy>=1.17 in /usr/local/lib/python3.7/dist-packages (from trans
formers) (1.19.5)
Requirement already satisfied: filelock in /usr/local/lib/python3.7/dist-packages (from transfor
mers) (3.0.12)
Requirement already satisfied: importlib-metadata in /usr/local/lib/python3.7/dist-packages (fro
m transformers) (4.6.4)
Collecting pyyaml>=5.1
  Downloading PyYAML-5.4.1-cp37-cp37m-manylinux1_x86_64.whl (636 kB)
     |████████████████████████████████| 636 kB 81.4 MB/s
Requirement already satisfied: regex!=2019.12.17 in /usr/local/lib/python3.7/dist-packages (from
transformers) (2019.12.20)
Requirement already satisfied: requests in /usr/local/lib/python3.7/dist-packages (from transfor
mers) (2.23.0)
Collecting tokenizers<0.11,>=0.10.1
  Downloading tokenizers-0.10.3-cp37-cp37m-manylinux_2_5_x86_64.manylinux1_x86_64.manylinux_2_12
_x86_64.manylinux2010_x86_64.whl (3.3 MB)
     |████████████████████████████████| 3.3 MB 78.1 MB/s
Collecting huggingface-hub>=0.0.12
  Downloading huggingface_hub-0.0.16-py3-none-any.whl (50 kB)
     |████████████████████████████████| 50 kB 8.5 MB/s
Requirement already satisfied: typing-extensions in /usr/local/lib/python3.7/dist-packages (from
huggingface-hub>=0.0.12->transformers) (3.7.4.3)
Requirement already satisfied: pyparsing>=2.0.2 in /usr/local/lib/python3.7/dist-packages (from
packaging->transformers) (2.4.7)
Requirement already satisfied: zipp>=0.5 in /usr/local/lib/python3.7/dist-packages (from importl
ib-metadata->transformers) (3.5.0)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.7/dist-packages (fro
m requests->transformers) (2021.5.30)
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.7/dist-packages (from requ
ests->transformers) (2.10)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in /usr/local/lib/python
3.7/dist-packages (from requests->transformers) (1.24.3)
Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/python3.7/dist-packages (from
requests->transformers) (3.0.4)
Requirement already satisfied: joblib in /usr/local/lib/python3.7/dist-packages (from sacremoses
->transformers) (1.0.1)
Requirement already satisfied: click in /usr/local/lib/python3.7/dist-packages (from sacremoses-
>transformers) (7.1.2)
Requirement already satisfied: six in /usr/local/lib/python3.7/dist-packages (from sacremoses->t
ransformers) (1.15.0)
Installing collected packages: tokenizers, sacremoses, pyyaml, huggingface-hub, transformers
  Attempting uninstall: pyyaml
    Found existing installation: PyYAML 3.13
    Uninstalling PyYAML-3.13:
      Successfully uninstalled PyYAML-3.13
Successfully installed huggingface-hub-0.0.16 pyyaml-5.4.1 sacremoses-0.0.45 tokenizers-0.10.3 t
ransformers-4.10.0
```

Run commands:

Create a test folder

In [ ]:
```
#! mkdir test_folder
```

## 2. Load drive

Show alll files in your colab:

In [6]:
```
import os
from google.colab import drive

drive.mount('/content/drive') # mount the drive
cwd = os.path.join('drive', 'MyDrive')

# show all files
#print(os.listdir(cwd))
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/cont
ent/drive", force_remount=True).
```

List all files in this folder:

In [7]:
```
cwd = os.path.join('drive','MyDrive', 'BIA667_Lab')
print(os.listdir(cwd))
```

```
['ColabIntroduction.ipynb', 'hw', 'PytorchBasics.ipynb', 'PytorchDataset.ipynb']
```

# Pytorch & Tensor Basics

## Import Packages

> Import the torcch and torch.nn (torch neural networks module)

In [8]:
```
# import
import torch
import torch.nn as nn
```

## Setup device

Set up our training device:

> If the gpu is available, we will use gpu. Otherwise, use cpu instead.

In [9]:
```
device = torch.device("cuda" if torch.cuda.is_available() else 'cpu')
print(device)
```

```
cuda
```

## Tensor Operations

The basic building block in Pytorch is the tensor. Tensors are very similar to the narray in the numpy and it has many pre-defined operations:

In [14]:
```
# define a new tensor is similar to define a numpy array
a_numpy = np.array([7, 7, 7])
a_tensor = torch.Tensor([7, 7, 7])
```

```python
print('Tensor and Numpy:')
print(a_numpy)
print(a_tensor)
print()

# tensor support basic operations
tensor1 = torch.Tensor([1, 2, 3])
tensor2 = torch.Tensor([2, 3, 4])
print('Basic Operations')
print('Add')
print(tensor1 + tensor2)
print('Multiply')
print(tensor1 * tensor2)
print()

# tensor object also has its own method
print('Tensor Oject method')
print(tensor1.add(tensor2))
print()

# function operates on tensor
print('Function operates on tensor')
max_value, max_index = torch.max(tensor1, dim=0)
print(f'Max value: {max_value}, Max Index: {max_index}')
```

```
Tensor and Numpy:
[7 7 7]
tensor([7., 7., 7.])

Basic Operations
Add
tensor([3., 5., 7.])
Multiply
tensor([ 2.,  6., 12.])

Tensor Oject method
tensor([3., 5., 7.])

Function operates on tensor
Max value: 3.0, Max Index: 2
```

Please note these oepartions will return a new tensor as result and the variables participating operations will not be modified. Alternatively, we can make these operations happens in-place. Instead of returing a extra tensor as result, in-place operations will directly change the content of given tensor.

In [15]:

```python
# not in-plcae
tensor1 = torch.Tensor([1, 2, 3])
tensor2 = torch.Tensor([2, 3, 4])
tensor1 = tensor1.add(tensor2)
print(tensor1)  # the content in tensor1 is not modified

# in-place
tensor1.add_(tensor2)
print(tensor1) # the content in tensor1 is modified
```

```
tensor([3., 5., 7.])
tensor([ 5.,  8., 11.])
```

In general, an in-place is the normal operation with extra '_' at the end. For example,

```
tensor.add() # not in-place
tensor.add_() # in-place

tensor.abs() # not in-place
tensor.abs_() # in-place
```

The in-place operation will be useful when you have limitation on memory. For example, if you have a huge tensor representation for a high resolution image, it may be costly to keep an another copy. For the full list of operations, please see here.

## Change the shape of the tensor

To change the shape of a tensor, we can use the

```
Tensor.view()
```

which is similar to numpy.reshape(). Also, we can use -1 as place holder to let the pytorch find the correct shape for us.

In [16]:
```
# initialize a tensor with random numbers
a_tensor = torch.randn((100, 33, 22, 11), dtype=torch.float)
print(a_tensor.size())  # use Tensor.size() to find the shape
print()


# reshape to (33, 22, 11, 100)
reshaped1 = a_tensor.view(33, 22, 11, 100)
print('After reshape')
print(reshaped1.size())
print()

# use -1 as place holder
reshaped2 = a_tensor.view(33, -1, 10, 220)  # let pytorch calculate the last dimension for us
print('After reshape')
print(reshaped2.size())
print()
```

```
torch.Size([100, 33, 22, 11])

After reshape
torch.Size([33, 22, 11, 100])

After reshape
torch.Size([33, 11, 10, 220])
```

## Move Tensor to GPU/CPU

You may have heard deep learning models are training fatser on GPUs. To do this, we will need to move our data, which is represents as tensors, to GPU.

In [17]:
```
# get the device on your machine
cpu = torch.device('cpu')
gpu = torch.device('cuda')
```

```
# often we use the following one liner to help us choose device
device = torch.device("cuda" if torch.cuda.is_available() else "cpu")

# move tensor to gpu
a_tensor = torch.Tensor([1, 2, 3])
a_tensor = a_tensor.to(gpu)  # you might get an error if you did not activate the gpu in your co
print('GPU tensor:')
print(a_tensor)
print()

# move back to cpu
a_tensor = a_tensor.to(cpu)
print('CPU tensor:')
print(a_tensor)
print()
```

```
GPU tensor:
tensor([1., 2., 3.], device='cuda:0')

CPU tensor:
tensor([1., 2., 3.])
```

The

```
tensor(...., device='cuda:0')
```

menas we have successfully moved tensor to GPU:0.

# Datasets

In PyTorch, we define a dataset class to generate mini-batches for model training

## Dependicies

import torch import torch.nn as nn import numpy as np from torch.utils.data import DataLoader, random_split, Dataset

In [18]:
```
import torch
import torch.nn as nn
import numpy as np
from torch.utils.data import DataLoader, random_split, Dataset
```

## Generate fake data

In [19]:
```
# fake data
features = torch.randn(size=(1000, 2)) * 20 + 30
w = torch.Tensor([1.5, 2])
targets = torch.matmul(features, w) + 1 +  torch.randn(size=(1000,))
```

## Check data

In [20]:
```
print(features[:5])
```

```
tensor([[36.6525, 13.8604],
        [37.0056, 55.6256],
        [32.3917, 41.7897],
        [63.2859, 24.9272],
        [15.1525, 20.7489]])
```

In [21]:
```
print(targets[:5])
```

```
tensor([ 82.3267, 167.7586, 134.6422, 144.8727,  65.9563])
```

In [22]:
```
type(features)
```

Out[22]:
```
torch.Tensor
```

In [23]:
```
# varify data
print(f'features shape: {features.size()}')
print(f'targets shape: {targets.size()}')
```

```
features shape: torch.Size([1000, 2])
targets shape: torch.Size([1000])
```

```
torch.Tensor.size()
```

will return the size(shape) of the Tensor.

In our case, the shape is (1000, 2) for features, which means we have 1000 data points and 2 features.

## Define Dataset Class

The dataset class should inherit from the pytorch's Dataset class, and we need to define:

1. `__init__`: Initialize your parent class and preprocess
2. `__getitem__`: Define how to retrieve your data by index, usually we return both data and corresponding label
3. `__len__`: Define how to get the total length of your data(how many observations/data points/rows) in your dataset

In [24]:
```
class MyDataset(Dataset):
    def __init__(self, features, labels):
        super(MyDataset, self).__init__()
        self.features = torch.Tensor(features)
        self.labels = torch.Tensor(labels)

    def __getitem__(self, index):
        return self.features[index], self.labels[index]

    def __len__(self):
        return self.labels.size()[0]
```

Create a dataset object:

In [26]:
```
dataset_example = MyDataset(features=features, labels=targets)

a_feature, a_target = dataset_example[0]
print(f'A feature:\n{a_feature}')
print(f'A target:\n{a_target}')
```

```
A feature:
tensor([36.6525, 13.8604])
A target:
82.32672119140625
```

## Split Dataset

To split our dataset to train, validation, test datasets, we can use the `random_split` function.

In [27]:
```
split_size = (np.array([0.6, 0.2, 0.2]) * len(dataset_example)).astype(np.int)
train_data, valid_data, test_data = random_split(dataset_example, lengths=split_size)

print(f'Train dataset length: {len(train_data)}')
print(f'Validation dataset length: {len(valid_data)}')
print(f'Test dataset length: {len(test_data)}')
```

```
Train dataset length: 600
Validation dataset length: 200
Test dataset length: 200
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1: DeprecationWarning: `np.int` is
a deprecated alias for the builtin `int`. To silence this warning, use `int` by itself. Doing th
is will not modify any behavior and is safe. When replacing `np.int`, you may wish to use e.g. `
np.int64` or `np.int32` to specify the precision. If you wish to review your current use, check
the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/devdocs/release/1.20.
0-notes.html#deprecations
  """Entry point for launching an IPython kernel.
```

## Data Loader

Convert to dataloader so we can use our data in train function:

In [28]:
```
train_loader = DataLoader(train_data, batch_size=32, shuffle=True)
valid_loader = DataLoader(valid_data, batch_size=32, shuffle=True)
test_loader = DataLoader(test_data, batch_size=32, shuffle=True)

# one batch example
one_batch_features, one_batch_labels = next(train_loader.__iter__())
print(one_batch_features.size())  # (batch x num_features)
print(one_batch_labels.size())  # (batch x num_labels)
```

```
torch.Size([32, 2])
torch.Size([32])
```

# Put Everything Together: Simple Linear Regression

## Define Model

$$y = ax + b$$

In [29]:
```
# pytorch simple linear regression model
class SimpleLinearRegression(nn.Module):
    # initilize and set up the layers
    def __init__(self):
        # initialize parent class
        super(SimpleLinearRegression, self).__init__()
        # define linear layer
```

```python
        self.linear = nn.Linear(in_features=2, out_features=1, bias=True)

    # how to pass your data through NN
    def forward(self, x):
        output = self.linear(x)
        return output

# model object
model = SimpleLinearRegression()
```

For the details of Pytorch Linear layers: Linear Layers

Take a look at the initialized parameters

In [30]:
```python
print([i for i in model.parameters()])
```

```
[Parameter containing:
tensor([[-0.1966,  0.6684]], requires_grad=True), Parameter containing:
tensor([0.0591], requires_grad=True)]
```

## Define a loss function and optimizer

In [31]:
```python
loss_func = nn.MSELoss()
optimizer = torch.optim.Adam(model.parameters(), lr=0.001)
```

## Training

In [32]:
```python
# move model to (device: GPU)
model = model.to(device)
```

In [33]:
```python
# traning loop
print('Training Starts:')

# training epochs
num_epochs = 1000

# record validation loss history
val_loss_hist = []

for epoch in range(num_epochs):

    model.train()  # start to train the model, activate training behavior
    train_loss = 0
    val_loss = 0

    for x, y in train_loader:

      # move batch to device
      x = x.to(device)
      y = y.to(device)

      # forward
      y_hat = model(x).squeeze()  # y_hat has shape (32,1), but y has shape (32,). Use squeeze to


      # calculate loss
      loss = loss_func(y_hat, y)
      train_loss += (loss.detach().item())
```

```
        # backpropation: calculate the gradients
        loss.backward()

        # update weight based on gradients
        optimizer.step()

        # delete gradients after parameter update
        optimizer.zero_grad()

    # valid
    model.eval()  # put the model in evaluation model

    with torch.no_grad():  # tell pytorch not to update parameters

        for x, y in valid_loader:

            # move batch to device
            x = x.to(device)
            y = y.to(device)

            # forward
            y_hat = model(x).squeeze()

            # calculate loss
            loss = loss_func(y_hat, y)
            val_loss += (loss.detach().item())

    # print
    print(f"Epoch:{epoch + 1} / {num_epochs}, train loss:{train_loss/len(train_loader):.3f}, val
    val_loss_hist.append(val_loss/len(valid_loader))
```

```
Training Starts:
Epoch:1 / 1000, train loss:10293.358, valid loss:9821.064
Epoch:2 / 1000, train loss:9969.612, valid loss:9911.798
Epoch:3 / 1000, train loss:9791.334, valid loss:9840.330
Epoch:4 / 1000, train loss:9522.004, valid loss:9604.985
Epoch:5 / 1000, train loss:9245.204, valid loss:8942.694
Epoch:6 / 1000, train loss:8992.308, valid loss:8605.342
Epoch:7 / 1000, train loss:8797.828, valid loss:8516.048
Epoch:8 / 1000, train loss:8587.433, valid loss:8073.622
Epoch:9 / 1000, train loss:8348.084, valid loss:8020.488
Epoch:10 / 1000, train loss:8131.169, valid loss:7519.508
Epoch:11 / 1000, train loss:7935.241, valid loss:7555.013
Epoch:12 / 1000, train loss:7740.336, valid loss:7717.530
Epoch:13 / 1000, train loss:7502.515, valid loss:6917.183
Epoch:14 / 1000, train loss:7298.234, valid loss:7048.142
Epoch:15 / 1000, train loss:7096.614, valid loss:6879.930
Epoch:16 / 1000, train loss:6889.314, valid loss:6549.749
Epoch:17 / 1000, train loss:6729.177, valid loss:6629.095
Epoch:18 / 1000, train loss:6536.209, valid loss:6467.546
Epoch:19 / 1000, train loss:6349.944, valid loss:6263.491
Epoch:20 / 1000, train loss:6183.684, valid loss:6104.233
Epoch:21 / 1000, train loss:6023.001, valid loss:6428.740
Epoch:22 / 1000, train loss:5839.276, valid loss:5437.506
Epoch:23 / 1000, train loss:5656.769, valid loss:5524.119
Epoch:24 / 1000, train loss:5508.502, valid loss:5399.432
Epoch:25 / 1000, train loss:5363.393, valid loss:5359.089
Epoch:26 / 1000, train loss:5178.326, valid loss:4791.985
Epoch:27 / 1000, train loss:5027.923, valid loss:5124.201
Epoch:28 / 1000, train loss:4885.981, valid loss:4674.917
Epoch:29 / 1000, train loss:4732.030, valid loss:4789.361
Epoch:30 / 1000, train loss:4603.532, valid loss:4464.841
Epoch:31 / 1000, train loss:4485.269, valid loss:4334.869
Epoch:32 / 1000, train loss:4327.399, valid loss:4174.158
Epoch:33 / 1000, train loss:4175.082, valid loss:4181.865
Epoch:34 / 1000, train loss:4063.484, valid loss:3976.902
Epoch:35 / 1000, train loss:3926.481, valid loss:3993.550
Epoch:36 / 1000, train loss:3813.459, valid loss:3697.308
Epoch:37 / 1000, train loss:3690.968, valid loss:3428.519
Epoch:38 / 1000, train loss:3580.820, valid loss:3578.672
Epoch:39 / 1000, train loss:3476.633, valid loss:3324.972
Epoch:40 / 1000, train loss:3370.167, valid loss:3098.745
Epoch:41 / 1000, train loss:3252.499, valid loss:2964.979
Epoch:42 / 1000, train loss:3123.649, valid loss:3032.172
Epoch:43 / 1000, train loss:3025.476, valid loss:2860.366
Epoch:44 / 1000, train loss:2932.585, valid loss:2987.524
Epoch:45 / 1000, train loss:2823.286, valid loss:2665.502
Epoch:46 / 1000, train loss:2732.156, valid loss:2635.947
Epoch:47 / 1000, train loss:2641.368, valid loss:2560.350
Epoch:48 / 1000, train loss:2552.462, valid loss:2326.824
Epoch:49 / 1000, train loss:2464.901, valid loss:2334.891
Epoch:50 / 1000, train loss:2368.981, valid loss:2344.266
Epoch:51 / 1000, train loss:2293.799, valid loss:2188.349
Epoch:52 / 1000, train loss:2216.044, valid loss:2105.435
Epoch:53 / 1000, train loss:2131.120, valid loss:2177.461
Epoch:54 / 1000, train loss:2056.687, valid loss:1936.776
Epoch:55 / 1000, train loss:1981.737, valid loss:1875.374
Epoch:56 / 1000, train loss:1907.917, valid loss:1819.689
Epoch:57 / 1000, train loss:1840.970, valid loss:1779.622
Epoch:58 / 1000, train loss:1776.591, valid loss:1684.867
Epoch:59 / 1000, train loss:1703.768, valid loss:1637.564
Epoch:60 / 1000, train loss:1644.184, valid loss:1538.045
Epoch:61 / 1000, train loss:1575.209, valid loss:1543.758
```

```
Epoch:62 / 1000, train loss:1521.071, valid loss:1471.767
Epoch:63 / 1000, train loss:1457.394, valid loss:1361.728
Epoch:64 / 1000, train loss:1402.453, valid loss:1294.330
Epoch:65 / 1000, train loss:1349.182, valid loss:1307.155
Epoch:66 / 1000, train loss:1298.498, valid loss:1259.898
Epoch:67 / 1000, train loss:1244.384, valid loss:1171.637
Epoch:68 / 1000, train loss:1196.603, valid loss:1194.547
Epoch:69 / 1000, train loss:1143.968, valid loss:1127.927
Epoch:70 / 1000, train loss:1099.122, valid loss:1032.900
Epoch:71 / 1000, train loss:1052.975, valid loss:1007.348
Epoch:72 / 1000, train loss:1009.927, valid loss:930.656
Epoch:73 / 1000, train loss:965.433, valid loss:956.900
Epoch:74 / 1000, train loss:924.382, valid loss:897.472
Epoch:75 / 1000, train loss:885.406, valid loss:865.783
Epoch:76 / 1000, train loss:847.364, valid loss:797.853
Epoch:77 / 1000, train loss:813.641, valid loss:756.585
Epoch:78 / 1000, train loss:775.632, valid loss:730.784
Epoch:79 / 1000, train loss:741.548, valid loss:692.445
Epoch:80 / 1000, train loss:709.051, valid loss:660.235
Epoch:81 / 1000, train loss:679.180, valid loss:641.214
Epoch:82 / 1000, train loss:646.186, valid loss:608.672
Epoch:83 / 1000, train loss:617.937, valid loss:566.324
Epoch:84 / 1000, train loss:588.015, valid loss:544.902
Epoch:85 / 1000, train loss:558.162, valid loss:520.525
Epoch:86 / 1000, train loss:534.926, valid loss:513.615
Epoch:87 / 1000, train loss:509.702, valid loss:478.999
Epoch:88 / 1000, train loss:485.759, valid loss:448.630
Epoch:89 / 1000, train loss:462.805, valid loss:422.229
Epoch:90 / 1000, train loss:442.533, valid loss:411.611
Epoch:91 / 1000, train loss:420.572, valid loss:386.536
Epoch:92 / 1000, train loss:399.983, valid loss:371.541
Epoch:93 / 1000, train loss:380.455, valid loss:368.574
Epoch:94 / 1000, train loss:361.281, valid loss:353.791
Epoch:95 / 1000, train loss:342.389, valid loss:336.035
Epoch:96 / 1000, train loss:327.131, valid loss:306.849
Epoch:97 / 1000, train loss:308.755, valid loss:296.210
Epoch:98 / 1000, train loss:293.448, valid loss:284.927
Epoch:99 / 1000, train loss:280.163, valid loss:258.851
Epoch:100 / 1000, train loss:265.225, valid loss:260.647
Epoch:101 / 1000, train loss:251.487, valid loss:255.165
Epoch:102 / 1000, train loss:238.950, valid loss:218.110
Epoch:103 / 1000, train loss:225.816, valid loss:208.243
Epoch:104 / 1000, train loss:213.699, valid loss:199.168
Epoch:105 / 1000, train loss:202.091, valid loss:195.653
Epoch:106 / 1000, train loss:191.601, valid loss:174.183
Epoch:107 / 1000, train loss:180.839, valid loss:175.446
Epoch:108 / 1000, train loss:171.435, valid loss:156.222
Epoch:109 / 1000, train loss:162.650, valid loss:151.464
Epoch:110 / 1000, train loss:153.473, valid loss:140.808
Epoch:111 / 1000, train loss:145.837, valid loss:131.264
Epoch:112 / 1000, train loss:137.744, valid loss:134.873
Epoch:113 / 1000, train loss:130.397, valid loss:124.798
Epoch:114 / 1000, train loss:122.831, valid loss:119.766
Epoch:115 / 1000, train loss:116.329, valid loss:100.566
Epoch:116 / 1000, train loss:109.591, valid loss:105.617
Epoch:117 / 1000, train loss:104.375, valid loss:100.562
Epoch:118 / 1000, train loss:97.691, valid loss:89.017
Epoch:119 / 1000, train loss:92.960, valid loss:88.107
Epoch:120 / 1000, train loss:87.400, valid loss:80.779
Epoch:121 / 1000, train loss:82.876, valid loss:77.886
Epoch:122 / 1000, train loss:78.118, valid loss:73.080
Epoch:123 / 1000, train loss:73.879, valid loss:69.550
```

```
Epoch:124 / 1000, train loss:70.003, valid loss:66.827
Epoch:125 / 1000, train loss:66.666, valid loss:66.633
Epoch:126 / 1000, train loss:62.857, valid loss:61.755
Epoch:127 / 1000, train loss:58.973, valid loss:56.150
Epoch:128 / 1000, train loss:55.701, valid loss:53.597
Epoch:129 / 1000, train loss:52.959, valid loss:46.945
Epoch:130 / 1000, train loss:50.026, valid loss:48.374
Epoch:131 / 1000, train loss:47.227, valid loss:45.534
Epoch:132 / 1000, train loss:45.402, valid loss:40.706
Epoch:133 / 1000, train loss:42.669, valid loss:38.709
Epoch:134 / 1000, train loss:40.456, valid loss:40.757
Epoch:135 / 1000, train loss:38.346, valid loss:35.768
Epoch:136 / 1000, train loss:36.431, valid loss:34.271
Epoch:137 / 1000, train loss:34.697, valid loss:34.283
Epoch:138 / 1000, train loss:32.975, valid loss:30.604
Epoch:139 / 1000, train loss:31.570, valid loss:32.905
Epoch:140 / 1000, train loss:30.025, valid loss:27.756
Epoch:141 / 1000, train loss:28.668, valid loss:27.745
Epoch:142 / 1000, train loss:27.216, valid loss:24.856
Epoch:143 / 1000, train loss:26.061, valid loss:24.800
Epoch:144 / 1000, train loss:24.902, valid loss:23.186
Epoch:145 / 1000, train loss:23.770, valid loss:24.272
Epoch:146 / 1000, train loss:22.907, valid loss:21.588
Epoch:147 / 1000, train loss:21.780, valid loss:21.158
Epoch:148 / 1000, train loss:20.966, valid loss:20.134
Epoch:149 / 1000, train loss:20.057, valid loss:19.405
Epoch:150 / 1000, train loss:19.326, valid loss:19.806
Epoch:151 / 1000, train loss:18.571, valid loss:19.563
Epoch:152 / 1000, train loss:17.864, valid loss:18.121
Epoch:153 / 1000, train loss:17.263, valid loss:17.316
Epoch:154 / 1000, train loss:16.607, valid loss:16.607
Epoch:155 / 1000, train loss:16.026, valid loss:16.338
Epoch:156 / 1000, train loss:15.538, valid loss:17.184
Epoch:157 / 1000, train loss:14.964, valid loss:14.327
Epoch:158 / 1000, train loss:14.542, valid loss:15.301
Epoch:159 / 1000, train loss:14.127, valid loss:13.069
Epoch:160 / 1000, train loss:13.678, valid loss:12.555
Epoch:161 / 1000, train loss:13.244, valid loss:14.326
Epoch:162 / 1000, train loss:12.819, valid loss:12.456
Epoch:163 / 1000, train loss:12.482, valid loss:11.840
Epoch:164 / 1000, train loss:12.200, valid loss:12.867
Epoch:165 / 1000, train loss:11.748, valid loss:11.824
Epoch:166 / 1000, train loss:11.425, valid loss:11.949
Epoch:167 / 1000, train loss:11.169, valid loss:11.976
Epoch:168 / 1000, train loss:10.860, valid loss:11.635
Epoch:169 / 1000, train loss:10.612, valid loss:11.850
Epoch:170 / 1000, train loss:10.355, valid loss:11.847
Epoch:171 / 1000, train loss:10.155, valid loss:10.243
Epoch:172 / 1000, train loss:9.803, valid loss:9.599
Epoch:173 / 1000, train loss:9.533, valid loss:9.178
Epoch:174 / 1000, train loss:9.341, valid loss:9.432
Epoch:175 / 1000, train loss:9.133, valid loss:9.069
Epoch:176 / 1000, train loss:8.988, valid loss:9.099
Epoch:177 / 1000, train loss:8.707, valid loss:8.597
Epoch:178 / 1000, train loss:8.487, valid loss:8.829
Epoch:179 / 1000, train loss:8.303, valid loss:8.691
Epoch:180 / 1000, train loss:8.090, valid loss:9.074
Epoch:181 / 1000, train loss:7.883, valid loss:8.196
Epoch:182 / 1000, train loss:7.723, valid loss:9.011
Epoch:183 / 1000, train loss:7.582, valid loss:7.588
Epoch:184 / 1000, train loss:7.382, valid loss:8.084
Epoch:185 / 1000, train loss:7.254, valid loss:8.813
```

```
Epoch:186 / 1000, train loss:7.105, valid loss:7.315
Epoch:187 / 1000, train loss:6.930, valid loss:7.609
Epoch:188 / 1000, train loss:6.738, valid loss:7.671
Epoch:189 / 1000, train loss:6.637, valid loss:6.674
Epoch:190 / 1000, train loss:6.455, valid loss:7.085
Epoch:191 / 1000, train loss:6.301, valid loss:6.203
Epoch:192 / 1000, train loss:6.191, valid loss:6.197
Epoch:193 / 1000, train loss:6.038, valid loss:6.024
Epoch:194 / 1000, train loss:5.938, valid loss:6.395
Epoch:195 / 1000, train loss:5.777, valid loss:5.668
Epoch:196 / 1000, train loss:5.645, valid loss:5.515
Epoch:197 / 1000, train loss:5.564, valid loss:6.125
Epoch:198 / 1000, train loss:5.372, valid loss:5.660
Epoch:199 / 1000, train loss:5.299, valid loss:5.370
Epoch:200 / 1000, train loss:5.163, valid loss:5.411
Epoch:201 / 1000, train loss:5.076, valid loss:4.863
Epoch:202 / 1000, train loss:4.944, valid loss:4.955
Epoch:203 / 1000, train loss:4.820, valid loss:5.416
Epoch:204 / 1000, train loss:4.699, valid loss:5.048
Epoch:205 / 1000, train loss:4.598, valid loss:4.906
Epoch:206 / 1000, train loss:4.468, valid loss:4.553
Epoch:207 / 1000, train loss:4.397, valid loss:4.461
Epoch:208 / 1000, train loss:4.286, valid loss:4.926
Epoch:209 / 1000, train loss:4.157, valid loss:4.046
Epoch:210 / 1000, train loss:4.071, valid loss:4.276
Epoch:211 / 1000, train loss:3.990, valid loss:4.083
Epoch:212 / 1000, train loss:3.898, valid loss:4.102
Epoch:213 / 1000, train loss:3.805, valid loss:3.838
Epoch:214 / 1000, train loss:3.722, valid loss:3.611
Epoch:215 / 1000, train loss:3.637, valid loss:3.745
Epoch:216 / 1000, train loss:3.537, valid loss:3.704
Epoch:217 / 1000, train loss:3.460, valid loss:3.653
Epoch:218 / 1000, train loss:3.380, valid loss:3.525
Epoch:219 / 1000, train loss:3.327, valid loss:3.931
Epoch:220 / 1000, train loss:3.226, valid loss:3.383
Epoch:221 / 1000, train loss:3.152, valid loss:3.193
Epoch:222 / 1000, train loss:3.092, valid loss:3.176
Epoch:223 / 1000, train loss:3.002, valid loss:2.921
Epoch:224 / 1000, train loss:2.957, valid loss:3.305
Epoch:225 / 1000, train loss:2.896, valid loss:2.905
Epoch:226 / 1000, train loss:2.806, valid loss:3.018
Epoch:227 / 1000, train loss:2.745, valid loss:2.872
Epoch:228 / 1000, train loss:2.686, valid loss:2.801
Epoch:229 / 1000, train loss:2.609, valid loss:2.775
Epoch:230 / 1000, train loss:2.556, valid loss:2.486
Epoch:231 / 1000, train loss:2.516, valid loss:2.631
Epoch:232 / 1000, train loss:2.459, valid loss:2.658
Epoch:233 / 1000, train loss:2.410, valid loss:2.389
Epoch:234 / 1000, train loss:2.363, valid loss:2.417
Epoch:235 / 1000, train loss:2.301, valid loss:2.486
Epoch:236 / 1000, train loss:2.259, valid loss:2.263
Epoch:237 / 1000, train loss:2.200, valid loss:2.324
Epoch:238 / 1000, train loss:2.173, valid loss:2.161
Epoch:239 / 1000, train loss:2.119, valid loss:2.107
Epoch:240 / 1000, train loss:2.069, valid loss:1.982
Epoch:241 / 1000, train loss:2.029, valid loss:2.057
Epoch:242 / 1000, train loss:1.992, valid loss:1.878
Epoch:243 / 1000, train loss:1.954, valid loss:2.098
Epoch:244 / 1000, train loss:1.918, valid loss:1.843
Epoch:245 / 1000, train loss:1.876, valid loss:2.198
Epoch:246 / 1000, train loss:1.847, valid loss:1.811
Epoch:247 / 1000, train loss:1.813, valid loss:1.892
```

```
Epoch:248 / 1000, train loss:1.788, valid loss:2.053
Epoch:249 / 1000, train loss:1.754, valid loss:1.681
Epoch:250 / 1000, train loss:1.723, valid loss:1.832
Epoch:251 / 1000, train loss:1.688, valid loss:1.680
Epoch:252 / 1000, train loss:1.651, valid loss:1.625
Epoch:253 / 1000, train loss:1.631, valid loss:1.619
Epoch:254 / 1000, train loss:1.601, valid loss:1.822
Epoch:255 / 1000, train loss:1.574, valid loss:1.757
Epoch:256 / 1000, train loss:1.551, valid loss:1.863
Epoch:257 / 1000, train loss:1.533, valid loss:1.460
Epoch:258 / 1000, train loss:1.513, valid loss:1.601
Epoch:259 / 1000, train loss:1.482, valid loss:1.460
Epoch:260 / 1000, train loss:1.473, valid loss:1.432
Epoch:261 / 1000, train loss:1.449, valid loss:1.628
Epoch:262 / 1000, train loss:1.428, valid loss:1.515
Epoch:263 / 1000, train loss:1.412, valid loss:1.425
Epoch:264 / 1000, train loss:1.394, valid loss:1.383
Epoch:265 / 1000, train loss:1.380, valid loss:1.355
Epoch:266 / 1000, train loss:1.352, valid loss:1.338
Epoch:267 / 1000, train loss:1.342, valid loss:1.306
Epoch:268 / 1000, train loss:1.334, valid loss:1.426
Epoch:269 / 1000, train loss:1.319, valid loss:1.263
Epoch:270 / 1000, train loss:1.306, valid loss:1.304
Epoch:271 / 1000, train loss:1.283, valid loss:1.242
Epoch:272 / 1000, train loss:1.276, valid loss:1.201
Epoch:273 / 1000, train loss:1.269, valid loss:1.277
Epoch:274 / 1000, train loss:1.250, valid loss:1.413
Epoch:275 / 1000, train loss:1.242, valid loss:1.261
Epoch:276 / 1000, train loss:1.237, valid loss:1.147
Epoch:277 / 1000, train loss:1.224, valid loss:1.181
Epoch:278 / 1000, train loss:1.216, valid loss:1.140
Epoch:279 / 1000, train loss:1.202, valid loss:1.333
Epoch:280 / 1000, train loss:1.197, valid loss:1.177
Epoch:281 / 1000, train loss:1.189, valid loss:1.174
Epoch:282 / 1000, train loss:1.181, valid loss:1.272
Epoch:283 / 1000, train loss:1.181, valid loss:1.154
Epoch:284 / 1000, train loss:1.168, valid loss:1.149
Epoch:285 / 1000, train loss:1.166, valid loss:1.167
Epoch:286 / 1000, train loss:1.151, valid loss:1.169
Epoch:287 / 1000, train loss:1.156, valid loss:1.060
Epoch:288 / 1000, train loss:1.144, valid loss:1.176
Epoch:289 / 1000, train loss:1.147, valid loss:1.135
Epoch:290 / 1000, train loss:1.133, valid loss:1.072
Epoch:291 / 1000, train loss:1.138, valid loss:1.082
Epoch:292 / 1000, train loss:1.133, valid loss:1.184
Epoch:293 / 1000, train loss:1.124, valid loss:1.092
Epoch:294 / 1000, train loss:1.126, valid loss:1.091
Epoch:295 / 1000, train loss:1.115, valid loss:1.024
Epoch:296 / 1000, train loss:1.113, valid loss:1.052
Epoch:297 / 1000, train loss:1.120, valid loss:1.103
Epoch:298 / 1000, train loss:1.107, valid loss:1.082
Epoch:299 / 1000, train loss:1.106, valid loss:1.079
Epoch:300 / 1000, train loss:1.102, valid loss:1.055
Epoch:301 / 1000, train loss:1.098, valid loss:1.074
Epoch:302 / 1000, train loss:1.093, valid loss:1.087
Epoch:303 / 1000, train loss:1.106, valid loss:1.038
Epoch:304 / 1000, train loss:1.095, valid loss:1.133
Epoch:305 / 1000, train loss:1.090, valid loss:1.140
Epoch:306 / 1000, train loss:1.091, valid loss:1.025
Epoch:307 / 1000, train loss:1.089, valid loss:1.091
Epoch:308 / 1000, train loss:1.082, valid loss:1.033
Epoch:309 / 1000, train loss:1.086, valid loss:1.014
```

```
Epoch:310 / 1000, train loss:1.085, valid loss:1.030
Epoch:311 / 1000, train loss:1.090, valid loss:1.078
Epoch:312 / 1000, train loss:1.084, valid loss:1.157
Epoch:313 / 1000, train loss:1.078, valid loss:1.119
Epoch:314 / 1000, train loss:1.079, valid loss:1.145
Epoch:315 / 1000, train loss:1.082, valid loss:1.134
Epoch:316 / 1000, train loss:1.078, valid loss:1.018
Epoch:317 / 1000, train loss:1.078, valid loss:1.053
Epoch:318 / 1000, train loss:1.078, valid loss:1.020
Epoch:319 / 1000, train loss:1.073, valid loss:1.042
Epoch:320 / 1000, train loss:1.078, valid loss:1.097
Epoch:321 / 1000, train loss:1.071, valid loss:0.977
Epoch:322 / 1000, train loss:1.075, valid loss:1.027
Epoch:323 / 1000, train loss:1.071, valid loss:1.069
Epoch:324 / 1000, train loss:1.073, valid loss:0.978
Epoch:325 / 1000, train loss:1.070, valid loss:1.005
Epoch:326 / 1000, train loss:1.074, valid loss:1.137
Epoch:327 / 1000, train loss:1.074, valid loss:0.986
Epoch:328 / 1000, train loss:1.074, valid loss:0.994
Epoch:329 / 1000, train loss:1.077, valid loss:0.962
Epoch:330 / 1000, train loss:1.070, valid loss:0.989
Epoch:331 / 1000, train loss:1.071, valid loss:1.021
Epoch:332 / 1000, train loss:1.073, valid loss:1.029
Epoch:333 / 1000, train loss:1.069, valid loss:0.991
Epoch:334 / 1000, train loss:1.071, valid loss:0.992
Epoch:335 / 1000, train loss:1.071, valid loss:1.019
Epoch:336 / 1000, train loss:1.071, valid loss:0.978
Epoch:337 / 1000, train loss:1.072, valid loss:1.028
Epoch:338 / 1000, train loss:1.069, valid loss:1.085
Epoch:339 / 1000, train loss:1.067, valid loss:1.085
Epoch:340 / 1000, train loss:1.068, valid loss:0.990
Epoch:341 / 1000, train loss:1.072, valid loss:0.969
Epoch:342 / 1000, train loss:1.063, valid loss:1.065
Epoch:343 / 1000, train loss:1.066, valid loss:0.951
Epoch:344 / 1000, train loss:1.068, valid loss:1.031
Epoch:345 / 1000, train loss:1.066, valid loss:0.988
Epoch:346 / 1000, train loss:1.068, valid loss:0.968
Epoch:347 / 1000, train loss:1.070, valid loss:1.072
Epoch:348 / 1000, train loss:1.067, valid loss:1.129
Epoch:349 / 1000, train loss:1.067, valid loss:1.037
Epoch:350 / 1000, train loss:1.071, valid loss:1.094
Epoch:351 / 1000, train loss:1.076, valid loss:1.014
Epoch:352 / 1000, train loss:1.074, valid loss:0.988
Epoch:353 / 1000, train loss:1.061, valid loss:0.960
Epoch:354 / 1000, train loss:1.070, valid loss:0.970
Epoch:355 / 1000, train loss:1.063, valid loss:1.080
Epoch:356 / 1000, train loss:1.067, valid loss:0.967
Epoch:357 / 1000, train loss:1.064, valid loss:0.999
Epoch:358 / 1000, train loss:1.067, valid loss:1.072
Epoch:359 / 1000, train loss:1.067, valid loss:1.154
Epoch:360 / 1000, train loss:1.063, valid loss:1.019
Epoch:361 / 1000, train loss:1.064, valid loss:1.049
Epoch:362 / 1000, train loss:1.066, valid loss:0.999
Epoch:363 / 1000, train loss:1.062, valid loss:1.093
Epoch:364 / 1000, train loss:1.064, valid loss:1.028
Epoch:365 / 1000, train loss:1.073, valid loss:1.091
Epoch:366 / 1000, train loss:1.061, valid loss:1.047
Epoch:367 / 1000, train loss:1.066, valid loss:1.024
Epoch:368 / 1000, train loss:1.063, valid loss:1.095
Epoch:369 / 1000, train loss:1.066, valid loss:1.001
Epoch:370 / 1000, train loss:1.065, valid loss:0.978
Epoch:371 / 1000, train loss:1.060, valid loss:0.986
```

```
Epoch:372 / 1000, train loss:1.060, valid loss:0.974
Epoch:373 / 1000, train loss:1.066, valid loss:1.006
Epoch:374 / 1000, train loss:1.063, valid loss:1.004
Epoch:375 / 1000, train loss:1.066, valid loss:1.079
Epoch:376 / 1000, train loss:1.060, valid loss:0.949
Epoch:377 / 1000, train loss:1.066, valid loss:0.984
Epoch:378 / 1000, train loss:1.072, valid loss:1.101
Epoch:379 / 1000, train loss:1.065, valid loss:0.978
Epoch:380 / 1000, train loss:1.064, valid loss:1.064
Epoch:381 / 1000, train loss:1.065, valid loss:0.942
Epoch:382 / 1000, train loss:1.060, valid loss:1.037
Epoch:383 / 1000, train loss:1.062, valid loss:1.164
Epoch:384 / 1000, train loss:1.067, valid loss:1.106
Epoch:385 / 1000, train loss:1.061, valid loss:0.998
Epoch:386 / 1000, train loss:1.071, valid loss:1.024
Epoch:387 / 1000, train loss:1.061, valid loss:1.010
Epoch:388 / 1000, train loss:1.060, valid loss:0.986
Epoch:389 / 1000, train loss:1.065, valid loss:1.045
Epoch:390 / 1000, train loss:1.069, valid loss:1.016
Epoch:391 / 1000, train loss:1.057, valid loss:1.077
Epoch:392 / 1000, train loss:1.065, valid loss:1.023
Epoch:393 / 1000, train loss:1.059, valid loss:0.958
Epoch:394 / 1000, train loss:1.055, valid loss:0.967
Epoch:395 / 1000, train loss:1.062, valid loss:0.967
Epoch:396 / 1000, train loss:1.064, valid loss:1.013
Epoch:397 / 1000, train loss:1.064, valid loss:0.982
Epoch:398 / 1000, train loss:1.066, valid loss:1.002
Epoch:399 / 1000, train loss:1.060, valid loss:1.065
Epoch:400 / 1000, train loss:1.056, valid loss:0.955
Epoch:401 / 1000, train loss:1.059, valid loss:1.015
Epoch:402 / 1000, train loss:1.063, valid loss:1.114
Epoch:403 / 1000, train loss:1.057, valid loss:1.048
Epoch:404 / 1000, train loss:1.058, valid loss:0.954
Epoch:405 / 1000, train loss:1.059, valid loss:1.056
Epoch:406 / 1000, train loss:1.059, valid loss:0.970
Epoch:407 / 1000, train loss:1.060, valid loss:0.961
Epoch:408 / 1000, train loss:1.064, valid loss:1.013
Epoch:409 / 1000, train loss:1.062, valid loss:1.094
Epoch:410 / 1000, train loss:1.059, valid loss:1.055
Epoch:411 / 1000, train loss:1.060, valid loss:0.988
Epoch:412 / 1000, train loss:1.060, valid loss:1.035
Epoch:413 / 1000, train loss:1.057, valid loss:1.117
Epoch:414 / 1000, train loss:1.053, valid loss:0.939
Epoch:415 / 1000, train loss:1.061, valid loss:1.040
Epoch:416 / 1000, train loss:1.057, valid loss:1.051
Epoch:417 / 1000, train loss:1.057, valid loss:1.097
Epoch:418 / 1000, train loss:1.058, valid loss:1.117
Epoch:419 / 1000, train loss:1.058, valid loss:1.177
Epoch:420 / 1000, train loss:1.051, valid loss:0.998
Epoch:421 / 1000, train loss:1.056, valid loss:1.056
Epoch:422 / 1000, train loss:1.057, valid loss:0.981
Epoch:423 / 1000, train loss:1.058, valid loss:1.065
Epoch:424 / 1000, train loss:1.057, valid loss:1.077
Epoch:425 / 1000, train loss:1.055, valid loss:1.009
Epoch:426 / 1000, train loss:1.050, valid loss:1.000
Epoch:427 / 1000, train loss:1.056, valid loss:0.928
Epoch:428 / 1000, train loss:1.052, valid loss:0.985
Epoch:429 / 1000, train loss:1.056, valid loss:1.039
Epoch:430 / 1000, train loss:1.060, valid loss:1.071
Epoch:431 / 1000, train loss:1.052, valid loss:1.041
Epoch:432 / 1000, train loss:1.060, valid loss:0.980
Epoch:433 / 1000, train loss:1.048, valid loss:0.970
```

```
Epoch:434 / 1000, train loss:1.055, valid loss:1.036
Epoch:435 / 1000, train loss:1.052, valid loss:0.966
Epoch:436 / 1000, train loss:1.047, valid loss:1.037
Epoch:437 / 1000, train loss:1.052, valid loss:0.967
Epoch:438 / 1000, train loss:1.053, valid loss:1.025
Epoch:439 / 1000, train loss:1.055, valid loss:1.119
Epoch:440 / 1000, train loss:1.056, valid loss:1.247
Epoch:441 / 1000, train loss:1.052, valid loss:0.980
Epoch:442 / 1000, train loss:1.053, valid loss:0.987
Epoch:443 / 1000, train loss:1.046, valid loss:1.020
Epoch:444 / 1000, train loss:1.051, valid loss:1.007
Epoch:445 / 1000, train loss:1.052, valid loss:1.088
Epoch:446 / 1000, train loss:1.055, valid loss:1.018
Epoch:447 / 1000, train loss:1.050, valid loss:1.192
Epoch:448 / 1000, train loss:1.049, valid loss:1.116
Epoch:449 / 1000, train loss:1.047, valid loss:1.041
Epoch:450 / 1000, train loss:1.051, valid loss:0.995
Epoch:451 / 1000, train loss:1.043, valid loss:0.965
Epoch:452 / 1000, train loss:1.050, valid loss:1.120
Epoch:453 / 1000, train loss:1.053, valid loss:1.062
Epoch:454 / 1000, train loss:1.047, valid loss:0.953
Epoch:455 / 1000, train loss:1.045, valid loss:1.024
Epoch:456 / 1000, train loss:1.050, valid loss:1.071
Epoch:457 / 1000, train loss:1.046, valid loss:1.045
Epoch:458 / 1000, train loss:1.045, valid loss:1.030
Epoch:459 / 1000, train loss:1.054, valid loss:1.007
Epoch:460 / 1000, train loss:1.045, valid loss:0.963
Epoch:461 / 1000, train loss:1.049, valid loss:0.962
Epoch:462 / 1000, train loss:1.044, valid loss:1.098
Epoch:463 / 1000, train loss:1.049, valid loss:0.975
Epoch:464 / 1000, train loss:1.049, valid loss:1.027
Epoch:465 / 1000, train loss:1.043, valid loss:1.085
Epoch:466 / 1000, train loss:1.046, valid loss:0.937
Epoch:467 / 1000, train loss:1.043, valid loss:1.158
Epoch:468 / 1000, train loss:1.046, valid loss:0.951
Epoch:469 / 1000, train loss:1.049, valid loss:0.947
Epoch:470 / 1000, train loss:1.046, valid loss:1.206
Epoch:471 / 1000, train loss:1.045, valid loss:1.030
Epoch:472 / 1000, train loss:1.047, valid loss:1.003
Epoch:473 / 1000, train loss:1.049, valid loss:0.944
Epoch:474 / 1000, train loss:1.046, valid loss:0.942
Epoch:475 / 1000, train loss:1.046, valid loss:1.135
Epoch:476 / 1000, train loss:1.043, valid loss:1.003
Epoch:477 / 1000, train loss:1.047, valid loss:1.035
Epoch:478 / 1000, train loss:1.043, valid loss:1.050
Epoch:479 / 1000, train loss:1.046, valid loss:0.989
Epoch:480 / 1000, train loss:1.048, valid loss:1.049
Epoch:481 / 1000, train loss:1.046, valid loss:0.951
Epoch:482 / 1000, train loss:1.048, valid loss:1.091
Epoch:483 / 1000, train loss:1.039, valid loss:1.122
Epoch:484 / 1000, train loss:1.041, valid loss:1.070
Epoch:485 / 1000, train loss:1.043, valid loss:1.005
Epoch:486 / 1000, train loss:1.047, valid loss:1.023
Epoch:487 / 1000, train loss:1.041, valid loss:1.025
Epoch:488 / 1000, train loss:1.057, valid loss:0.948
Epoch:489 / 1000, train loss:1.045, valid loss:1.014
Epoch:490 / 1000, train loss:1.043, valid loss:0.986
Epoch:491 / 1000, train loss:1.041, valid loss:1.108
Epoch:492 / 1000, train loss:1.047, valid loss:1.014
Epoch:493 / 1000, train loss:1.037, valid loss:0.963
Epoch:494 / 1000, train loss:1.039, valid loss:1.128
Epoch:495 / 1000, train loss:1.044, valid loss:1.001
```

```
Epoch:496 / 1000, train loss:1.038, valid loss:1.002
Epoch:497 / 1000, train loss:1.041, valid loss:1.073
Epoch:498 / 1000, train loss:1.042, valid loss:0.990
Epoch:499 / 1000, train loss:1.038, valid loss:1.141
Epoch:500 / 1000, train loss:1.043, valid loss:0.955
Epoch:501 / 1000, train loss:1.046, valid loss:1.057
Epoch:502 / 1000, train loss:1.035, valid loss:0.974
Epoch:503 / 1000, train loss:1.038, valid loss:1.016
Epoch:504 / 1000, train loss:1.038, valid loss:1.057
Epoch:505 / 1000, train loss:1.034, valid loss:0.967
Epoch:506 / 1000, train loss:1.038, valid loss:0.957
Epoch:507 / 1000, train loss:1.041, valid loss:0.947
Epoch:508 / 1000, train loss:1.036, valid loss:0.985
Epoch:509 / 1000, train loss:1.045, valid loss:1.011
Epoch:510 / 1000, train loss:1.035, valid loss:0.999
Epoch:511 / 1000, train loss:1.038, valid loss:1.018
Epoch:512 / 1000, train loss:1.036, valid loss:1.043
Epoch:513 / 1000, train loss:1.038, valid loss:1.036
Epoch:514 / 1000, train loss:1.033, valid loss:0.995
Epoch:515 / 1000, train loss:1.044, valid loss:0.953
Epoch:516 / 1000, train loss:1.038, valid loss:0.921
Epoch:517 / 1000, train loss:1.031, valid loss:0.934
Epoch:518 / 1000, train loss:1.038, valid loss:0.963
Epoch:519 / 1000, train loss:1.033, valid loss:1.044
Epoch:520 / 1000, train loss:1.026, valid loss:1.057
Epoch:521 / 1000, train loss:1.033, valid loss:1.068
Epoch:522 / 1000, train loss:1.038, valid loss:1.058
Epoch:523 / 1000, train loss:1.029, valid loss:1.008
Epoch:524 / 1000, train loss:1.030, valid loss:1.020
Epoch:525 / 1000, train loss:1.029, valid loss:1.061
Epoch:526 / 1000, train loss:1.032, valid loss:1.083
Epoch:527 / 1000, train loss:1.039, valid loss:1.156
Epoch:528 / 1000, train loss:1.033, valid loss:1.096
Epoch:529 / 1000, train loss:1.036, valid loss:1.164
Epoch:530 / 1000, train loss:1.042, valid loss:0.938
Epoch:531 / 1000, train loss:1.030, valid loss:1.194
Epoch:532 / 1000, train loss:1.032, valid loss:1.010
Epoch:533 / 1000, train loss:1.029, valid loss:0.989
Epoch:534 / 1000, train loss:1.031, valid loss:1.119
Epoch:535 / 1000, train loss:1.031, valid loss:1.138
Epoch:536 / 1000, train loss:1.026, valid loss:1.084
Epoch:537 / 1000, train loss:1.027, valid loss:0.954
Epoch:538 / 1000, train loss:1.029, valid loss:0.994
Epoch:539 / 1000, train loss:1.028, valid loss:1.059
Epoch:540 / 1000, train loss:1.028, valid loss:1.028
Epoch:541 / 1000, train loss:1.028, valid loss:1.042
Epoch:542 / 1000, train loss:1.031, valid loss:0.976
Epoch:543 / 1000, train loss:1.035, valid loss:1.053
Epoch:544 / 1000, train loss:1.035, valid loss:0.960
Epoch:545 / 1000, train loss:1.022, valid loss:1.028
Epoch:546 / 1000, train loss:1.029, valid loss:0.987
Epoch:547 / 1000, train loss:1.029, valid loss:0.995
Epoch:548 / 1000, train loss:1.027, valid loss:0.993
Epoch:549 / 1000, train loss:1.023, valid loss:0.981
Epoch:550 / 1000, train loss:1.028, valid loss:0.980
Epoch:551 / 1000, train loss:1.026, valid loss:1.033
Epoch:552 / 1000, train loss:1.028, valid loss:0.964
Epoch:553 / 1000, train loss:1.023, valid loss:1.041
Epoch:554 / 1000, train loss:1.024, valid loss:0.992
Epoch:555 / 1000, train loss:1.028, valid loss:1.038
Epoch:556 / 1000, train loss:1.029, valid loss:0.957
Epoch:557 / 1000, train loss:1.027, valid loss:1.044
```

```
Epoch:558 / 1000, train loss:1.023, valid loss:1.047
Epoch:559 / 1000, train loss:1.030, valid loss:1.013
Epoch:560 / 1000, train loss:1.025, valid loss:0.991
Epoch:561 / 1000, train loss:1.024, valid loss:1.077
Epoch:562 / 1000, train loss:1.020, valid loss:1.023
Epoch:563 / 1000, train loss:1.023, valid loss:1.030
Epoch:564 / 1000, train loss:1.026, valid loss:0.959
Epoch:565 / 1000, train loss:1.024, valid loss:1.092
Epoch:566 / 1000, train loss:1.019, valid loss:0.940
Epoch:567 / 1000, train loss:1.018, valid loss:1.018
Epoch:568 / 1000, train loss:1.023, valid loss:0.915
Epoch:569 / 1000, train loss:1.025, valid loss:1.003
Epoch:570 / 1000, train loss:1.026, valid loss:1.006
Epoch:571 / 1000, train loss:1.023, valid loss:1.118
Epoch:572 / 1000, train loss:1.023, valid loss:1.092
Epoch:573 / 1000, train loss:1.021, valid loss:1.054
Epoch:574 / 1000, train loss:1.019, valid loss:0.973
Epoch:575 / 1000, train loss:1.014, valid loss:0.969
Epoch:576 / 1000, train loss:1.025, valid loss:1.184
Epoch:577 / 1000, train loss:1.020, valid loss:1.046
Epoch:578 / 1000, train loss:1.014, valid loss:0.991
Epoch:579 / 1000, train loss:1.024, valid loss:1.114
Epoch:580 / 1000, train loss:1.025, valid loss:0.947
Epoch:581 / 1000, train loss:1.014, valid loss:0.991
Epoch:582 / 1000, train loss:1.022, valid loss:0.973
Epoch:583 / 1000, train loss:1.020, valid loss:1.098
Epoch:584 / 1000, train loss:1.021, valid loss:0.995
Epoch:585 / 1000, train loss:1.015, valid loss:1.047
Epoch:586 / 1000, train loss:1.020, valid loss:1.008
Epoch:587 / 1000, train loss:1.016, valid loss:0.950
Epoch:588 / 1000, train loss:1.016, valid loss:1.035
Epoch:589 / 1000, train loss:1.021, valid loss:1.112
Epoch:590 / 1000, train loss:1.013, valid loss:1.049
Epoch:591 / 1000, train loss:1.012, valid loss:1.016
Epoch:592 / 1000, train loss:1.016, valid loss:1.015
Epoch:593 / 1000, train loss:1.017, valid loss:0.977
Epoch:594 / 1000, train loss:1.017, valid loss:1.118
Epoch:595 / 1000, train loss:1.021, valid loss:1.034
Epoch:596 / 1000, train loss:1.017, valid loss:0.965
Epoch:597 / 1000, train loss:1.020, valid loss:0.951
Epoch:598 / 1000, train loss:1.012, valid loss:1.004
Epoch:599 / 1000, train loss:1.021, valid loss:1.003
Epoch:600 / 1000, train loss:1.013, valid loss:1.009
Epoch:601 / 1000, train loss:1.014, valid loss:1.015
Epoch:602 / 1000, train loss:1.018, valid loss:1.018
Epoch:603 / 1000, train loss:1.010, valid loss:0.989
Epoch:604 / 1000, train loss:1.014, valid loss:1.071
Epoch:605 / 1000, train loss:1.018, valid loss:0.994
Epoch:606 / 1000, train loss:1.027, valid loss:1.002
Epoch:607 / 1000, train loss:1.019, valid loss:1.171
Epoch:608 / 1000, train loss:1.012, valid loss:1.076
Epoch:609 / 1000, train loss:1.010, valid loss:1.052
Epoch:610 / 1000, train loss:1.019, valid loss:1.044
Epoch:611 / 1000, train loss:1.013, valid loss:0.990
Epoch:612 / 1000, train loss:1.017, valid loss:1.054
Epoch:613 / 1000, train loss:1.009, valid loss:0.966
Epoch:614 / 1000, train loss:1.008, valid loss:0.971
Epoch:615 / 1000, train loss:1.015, valid loss:1.066
Epoch:616 / 1000, train loss:1.013, valid loss:0.950
Epoch:617 / 1000, train loss:1.013, valid loss:1.036
Epoch:618 / 1000, train loss:1.006, valid loss:1.005
Epoch:619 / 1000, train loss:1.015, valid loss:1.031
```

```
Epoch:620 / 1000, train loss:1.014, valid loss:0.953
Epoch:621 / 1000, train loss:1.015, valid loss:0.976
Epoch:622 / 1000, train loss:1.011, valid loss:0.984
Epoch:623 / 1000, train loss:1.012, valid loss:0.958
Epoch:624 / 1000, train loss:1.014, valid loss:1.050
Epoch:625 / 1000, train loss:1.008, valid loss:0.969
Epoch:626 / 1000, train loss:1.012, valid loss:1.122
Epoch:627 / 1000, train loss:1.013, valid loss:1.042
Epoch:628 / 1000, train loss:1.009, valid loss:1.032
Epoch:629 / 1000, train loss:1.013, valid loss:0.948
Epoch:630 / 1000, train loss:1.007, valid loss:0.992
Epoch:631 / 1000, train loss:1.010, valid loss:0.982
Epoch:632 / 1000, train loss:1.013, valid loss:0.944
Epoch:633 / 1000, train loss:1.005, valid loss:1.050
Epoch:634 / 1000, train loss:1.016, valid loss:0.967
Epoch:635 / 1000, train loss:1.012, valid loss:0.997
Epoch:636 / 1000, train loss:1.012, valid loss:0.950
Epoch:637 / 1000, train loss:1.011, valid loss:1.035
Epoch:638 / 1000, train loss:1.019, valid loss:0.994
Epoch:639 / 1000, train loss:1.020, valid loss:1.000
Epoch:640 / 1000, train loss:1.019, valid loss:1.009
Epoch:641 / 1000, train loss:1.019, valid loss:1.146
Epoch:642 / 1000, train loss:1.011, valid loss:0.999
Epoch:643 / 1000, train loss:1.014, valid loss:1.005
Epoch:644 / 1000, train loss:1.010, valid loss:1.049
Epoch:645 / 1000, train loss:1.016, valid loss:1.009
Epoch:646 / 1000, train loss:1.013, valid loss:1.121
Epoch:647 / 1000, train loss:1.003, valid loss:0.980
Epoch:648 / 1000, train loss:1.012, valid loss:1.014
Epoch:649 / 1000, train loss:1.011, valid loss:1.086
Epoch:650 / 1000, train loss:1.012, valid loss:1.002
Epoch:651 / 1000, train loss:1.015, valid loss:1.017
Epoch:652 / 1000, train loss:1.012, valid loss:0.973
Epoch:653 / 1000, train loss:1.003, valid loss:1.044
Epoch:654 / 1000, train loss:1.011, valid loss:1.035
Epoch:655 / 1000, train loss:1.011, valid loss:0.948
Epoch:656 / 1000, train loss:1.009, valid loss:0.967
Epoch:657 / 1000, train loss:1.010, valid loss:0.973
Epoch:658 / 1000, train loss:1.015, valid loss:0.976
Epoch:659 / 1000, train loss:1.013, valid loss:1.047
Epoch:660 / 1000, train loss:1.009, valid loss:0.966
Epoch:661 / 1000, train loss:1.002, valid loss:0.989
Epoch:662 / 1000, train loss:1.011, valid loss:0.987
Epoch:663 / 1000, train loss:1.010, valid loss:1.119
Epoch:664 / 1000, train loss:1.010, valid loss:0.980
Epoch:665 / 1000, train loss:1.020, valid loss:1.002
Epoch:666 / 1000, train loss:1.012, valid loss:1.058
Epoch:667 / 1000, train loss:1.009, valid loss:1.056
Epoch:668 / 1000, train loss:1.012, valid loss:1.053
Epoch:669 / 1000, train loss:1.013, valid loss:0.987
Epoch:670 / 1000, train loss:1.025, valid loss:1.134
Epoch:671 / 1000, train loss:1.012, valid loss:1.002
Epoch:672 / 1000, train loss:1.008, valid loss:1.038
Epoch:673 / 1000, train loss:1.009, valid loss:1.047
Epoch:674 / 1000, train loss:1.020, valid loss:1.021
Epoch:675 / 1000, train loss:1.021, valid loss:1.068
Epoch:676 / 1000, train loss:1.005, valid loss:1.133
Epoch:677 / 1000, train loss:1.008, valid loss:0.986
Epoch:678 / 1000, train loss:1.007, valid loss:0.965
Epoch:679 / 1000, train loss:1.008, valid loss:1.025
Epoch:680 / 1000, train loss:1.012, valid loss:1.051
Epoch:681 / 1000, train loss:1.016, valid loss:1.026
```

```
Epoch:682 / 1000, train loss:1.019, valid loss:1.009
Epoch:683 / 1000, train loss:1.019, valid loss:0.980
Epoch:684 / 1000, train loss:1.025, valid loss:1.061
Epoch:685 / 1000, train loss:1.022, valid loss:1.005
Epoch:686 / 1000, train loss:1.015, valid loss:0.941
Epoch:687 / 1000, train loss:1.014, valid loss:1.118
Epoch:688 / 1000, train loss:1.016, valid loss:0.981
Epoch:689 / 1000, train loss:1.024, valid loss:1.045
Epoch:690 / 1000, train loss:1.014, valid loss:1.054
Epoch:691 / 1000, train loss:1.007, valid loss:1.106
Epoch:692 / 1000, train loss:1.004, valid loss:1.025
Epoch:693 / 1000, train loss:1.006, valid loss:1.032
Epoch:694 / 1000, train loss:1.010, valid loss:0.986
Epoch:695 / 1000, train loss:1.002, valid loss:1.048
Epoch:696 / 1000, train loss:1.007, valid loss:0.963
Epoch:697 / 1000, train loss:1.009, valid loss:1.028
Epoch:698 / 1000, train loss:1.012, valid loss:1.096
Epoch:699 / 1000, train loss:1.015, valid loss:1.013
Epoch:700 / 1000, train loss:1.017, valid loss:1.008
Epoch:701 / 1000, train loss:1.014, valid loss:1.038
Epoch:702 / 1000, train loss:1.009, valid loss:1.038
Epoch:703 / 1000, train loss:1.005, valid loss:1.045
Epoch:704 / 1000, train loss:1.015, valid loss:0.972
Epoch:705 / 1000, train loss:1.004, valid loss:0.936
Epoch:706 / 1000, train loss:1.007, valid loss:1.002
Epoch:707 / 1000, train loss:1.007, valid loss:0.982
Epoch:708 / 1000, train loss:1.013, valid loss:1.229
Epoch:709 / 1000, train loss:1.003, valid loss:0.947
Epoch:710 / 1000, train loss:1.014, valid loss:1.003
Epoch:711 / 1000, train loss:1.006, valid loss:1.002
Epoch:712 / 1000, train loss:1.006, valid loss:1.013
Epoch:713 / 1000, train loss:1.007, valid loss:0.985
Epoch:714 / 1000, train loss:1.006, valid loss:1.062
Epoch:715 / 1000, train loss:1.002, valid loss:1.005
Epoch:716 / 1000, train loss:1.009, valid loss:0.997
Epoch:717 / 1000, train loss:1.013, valid loss:0.968
Epoch:718 / 1000, train loss:1.020, valid loss:0.986
Epoch:719 / 1000, train loss:1.013, valid loss:0.973
Epoch:720 / 1000, train loss:1.012, valid loss:1.064
Epoch:721 / 1000, train loss:1.004, valid loss:1.106
Epoch:722 / 1000, train loss:1.011, valid loss:0.951
Epoch:723 / 1000, train loss:1.015, valid loss:1.081
Epoch:724 / 1000, train loss:1.005, valid loss:0.975
Epoch:725 / 1000, train loss:1.007, valid loss:1.035
Epoch:726 / 1000, train loss:1.012, valid loss:1.010
Epoch:727 / 1000, train loss:1.005, valid loss:1.043
Epoch:728 / 1000, train loss:1.003, valid loss:1.087
Epoch:729 / 1000, train loss:1.006, valid loss:1.071
Epoch:730 / 1000, train loss:1.013, valid loss:0.977
Epoch:731 / 1000, train loss:1.005, valid loss:0.946
Epoch:732 / 1000, train loss:1.010, valid loss:0.962
Epoch:733 / 1000, train loss:1.028, valid loss:0.997
Epoch:734 / 1000, train loss:1.016, valid loss:1.143
Epoch:735 / 1000, train loss:1.005, valid loss:1.093
Epoch:736 / 1000, train loss:1.007, valid loss:1.107
Epoch:737 / 1000, train loss:1.010, valid loss:1.062
Epoch:738 / 1000, train loss:1.016, valid loss:0.970
Epoch:739 / 1000, train loss:1.007, valid loss:0.956
Epoch:740 / 1000, train loss:1.006, valid loss:1.119
Epoch:741 / 1000, train loss:1.010, valid loss:1.014
Epoch:742 / 1000, train loss:1.014, valid loss:0.933
Epoch:743 / 1000, train loss:1.015, valid loss:1.065
```

```
Epoch:744 / 1000, train loss:1.006, valid loss:1.109
Epoch:745 / 1000, train loss:1.002, valid loss:1.044
Epoch:746 / 1000, train loss:1.008, valid loss:1.067
Epoch:747 / 1000, train loss:1.007, valid loss:1.088
Epoch:748 / 1000, train loss:1.009, valid loss:1.015
Epoch:749 / 1000, train loss:1.005, valid loss:1.016
Epoch:750 / 1000, train loss:1.005, valid loss:0.955
Epoch:751 / 1000, train loss:1.004, valid loss:1.007
Epoch:752 / 1000, train loss:1.015, valid loss:1.087
Epoch:753 / 1000, train loss:1.005, valid loss:0.952
Epoch:754 / 1000, train loss:1.006, valid loss:0.989
Epoch:755 / 1000, train loss:1.007, valid loss:1.005
Epoch:756 / 1000, train loss:1.006, valid loss:1.016
Epoch:757 / 1000, train loss:1.011, valid loss:1.028
Epoch:758 / 1000, train loss:1.011, valid loss:1.058
Epoch:759 / 1000, train loss:1.016, valid loss:1.058
Epoch:760 / 1000, train loss:1.010, valid loss:1.012
Epoch:761 / 1000, train loss:1.025, valid loss:0.964
Epoch:762 / 1000, train loss:1.010, valid loss:1.080
Epoch:763 / 1000, train loss:1.006, valid loss:1.031
Epoch:764 / 1000, train loss:1.005, valid loss:0.974
Epoch:765 / 1000, train loss:1.004, valid loss:0.978
Epoch:766 / 1000, train loss:1.007, valid loss:1.245
Epoch:767 / 1000, train loss:1.002, valid loss:1.094
Epoch:768 / 1000, train loss:1.016, valid loss:1.184
Epoch:769 / 1000, train loss:1.016, valid loss:1.089
Epoch:770 / 1000, train loss:1.006, valid loss:1.079
Epoch:771 / 1000, train loss:1.002, valid loss:1.043
Epoch:772 / 1000, train loss:1.016, valid loss:1.131
Epoch:773 / 1000, train loss:1.006, valid loss:0.960
Epoch:774 / 1000, train loss:1.008, valid loss:0.990
Epoch:775 / 1000, train loss:1.009, valid loss:0.977
Epoch:776 / 1000, train loss:1.011, valid loss:0.951
Epoch:777 / 1000, train loss:1.010, valid loss:0.956
Epoch:778 / 1000, train loss:1.007, valid loss:1.005
Epoch:779 / 1000, train loss:1.006, valid loss:1.007
Epoch:780 / 1000, train loss:1.006, valid loss:1.007
Epoch:781 / 1000, train loss:1.012, valid loss:0.964
Epoch:782 / 1000, train loss:1.012, valid loss:1.064
Epoch:783 / 1000, train loss:1.003, valid loss:1.087
Epoch:784 / 1000, train loss:1.008, valid loss:1.117
Epoch:785 / 1000, train loss:1.007, valid loss:0.955
Epoch:786 / 1000, train loss:1.006, valid loss:0.925
Epoch:787 / 1000, train loss:1.010, valid loss:0.940
Epoch:788 / 1000, train loss:1.009, valid loss:1.076
Epoch:789 / 1000, train loss:1.002, valid loss:1.162
Epoch:790 / 1000, train loss:1.008, valid loss:1.031
Epoch:791 / 1000, train loss:1.013, valid loss:1.051
Epoch:792 / 1000, train loss:1.008, valid loss:0.967
Epoch:793 / 1000, train loss:1.012, valid loss:0.999
Epoch:794 / 1000, train loss:1.011, valid loss:1.024
Epoch:795 / 1000, train loss:1.002, valid loss:0.969
Epoch:796 / 1000, train loss:1.008, valid loss:0.967
Epoch:797 / 1000, train loss:1.008, valid loss:0.998
Epoch:798 / 1000, train loss:1.010, valid loss:1.095
Epoch:799 / 1000, train loss:1.013, valid loss:1.005
Epoch:800 / 1000, train loss:1.006, valid loss:1.044
Epoch:801 / 1000, train loss:1.006, valid loss:1.017
Epoch:802 / 1000, train loss:1.003, valid loss:0.995
Epoch:803 / 1000, train loss:1.003, valid loss:1.088
Epoch:804 / 1000, train loss:1.012, valid loss:0.956
Epoch:805 / 1000, train loss:1.019, valid loss:0.934
```

```
Epoch:806 / 1000, train loss:1.013, valid loss:0.986
Epoch:807 / 1000, train loss:1.010, valid loss:0.942
Epoch:808 / 1000, train loss:1.004, valid loss:1.122
Epoch:809 / 1000, train loss:1.013, valid loss:0.936
Epoch:810 / 1000, train loss:1.008, valid loss:0.967
Epoch:811 / 1000, train loss:1.010, valid loss:1.075
Epoch:812 / 1000, train loss:1.009, valid loss:1.109
Epoch:813 / 1000, train loss:1.016, valid loss:0.976
Epoch:814 / 1000, train loss:1.009, valid loss:1.139
Epoch:815 / 1000, train loss:1.002, valid loss:1.070
Epoch:816 / 1000, train loss:1.005, valid loss:0.956
Epoch:817 / 1000, train loss:1.024, valid loss:1.096
Epoch:818 / 1000, train loss:1.024, valid loss:1.037
Epoch:819 / 1000, train loss:1.008, valid loss:1.015
Epoch:820 / 1000, train loss:1.011, valid loss:0.984
Epoch:821 / 1000, train loss:1.005, valid loss:1.113
Epoch:822 / 1000, train loss:1.021, valid loss:0.971
Epoch:823 / 1000, train loss:1.011, valid loss:1.077
Epoch:824 / 1000, train loss:1.004, valid loss:0.949
Epoch:825 / 1000, train loss:1.006, valid loss:0.954
Epoch:826 / 1000, train loss:1.011, valid loss:1.209
Epoch:827 / 1000, train loss:1.006, valid loss:1.046
Epoch:828 / 1000, train loss:1.007, valid loss:1.027
Epoch:829 / 1000, train loss:1.011, valid loss:1.060
Epoch:830 / 1000, train loss:1.015, valid loss:1.027
Epoch:831 / 1000, train loss:1.013, valid loss:0.948
Epoch:832 / 1000, train loss:1.011, valid loss:1.036
Epoch:833 / 1000, train loss:1.018, valid loss:1.017
Epoch:834 / 1000, train loss:1.014, valid loss:1.131
Epoch:835 / 1000, train loss:1.012, valid loss:1.100
Epoch:836 / 1000, train loss:1.024, valid loss:0.988
Epoch:837 / 1000, train loss:1.016, valid loss:1.000
Epoch:838 / 1000, train loss:1.016, valid loss:0.992
Epoch:839 / 1000, train loss:1.003, valid loss:1.059
Epoch:840 / 1000, train loss:1.012, valid loss:1.013
Epoch:841 / 1000, train loss:1.011, valid loss:1.018
Epoch:842 / 1000, train loss:1.003, valid loss:1.015
Epoch:843 / 1000, train loss:1.011, valid loss:0.973
Epoch:844 / 1000, train loss:1.012, valid loss:0.998
Epoch:845 / 1000, train loss:1.005, valid loss:1.017
Epoch:846 / 1000, train loss:1.007, valid loss:1.066
Epoch:847 / 1000, train loss:1.022, valid loss:1.000
Epoch:848 / 1000, train loss:1.006, valid loss:1.033
Epoch:849 / 1000, train loss:1.016, valid loss:0.980
Epoch:850 / 1000, train loss:1.014, valid loss:1.041
Epoch:851 / 1000, train loss:1.014, valid loss:1.050
Epoch:852 / 1000, train loss:1.007, valid loss:1.110
Epoch:853 / 1000, train loss:1.009, valid loss:1.107
Epoch:854 / 1000, train loss:1.022, valid loss:0.981
Epoch:855 / 1000, train loss:1.005, valid loss:0.995
Epoch:856 / 1000, train loss:1.008, valid loss:0.940
Epoch:857 / 1000, train loss:1.009, valid loss:1.068
Epoch:858 / 1000, train loss:1.007, valid loss:1.071
Epoch:859 / 1000, train loss:1.007, valid loss:0.962
Epoch:860 / 1000, train loss:1.013, valid loss:1.012
Epoch:861 / 1000, train loss:1.013, valid loss:1.021
Epoch:862 / 1000, train loss:1.013, valid loss:1.040
Epoch:863 / 1000, train loss:1.002, valid loss:1.006
Epoch:864 / 1000, train loss:1.018, valid loss:1.006
Epoch:865 / 1000, train loss:1.025, valid loss:1.079
Epoch:866 / 1000, train loss:1.003, valid loss:0.974
Epoch:867 / 1000, train loss:1.014, valid loss:0.968
```

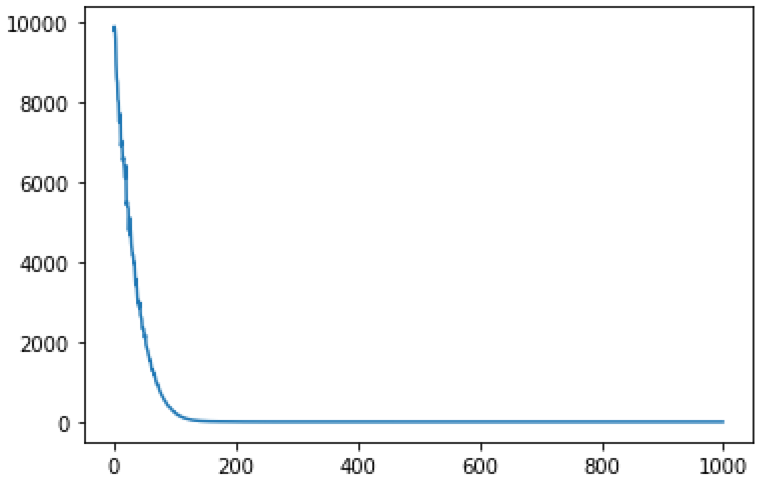
```
Epoch:868 / 1000, train loss:1.021, valid loss:0.967
Epoch:869 / 1000, train loss:1.009, valid loss:0.998
Epoch:870 / 1000, train loss:1.009, valid loss:1.027
Epoch:871 / 1000, train loss:1.010, valid loss:1.027
Epoch:872 / 1000, train loss:1.002, valid loss:0.958
Epoch:873 / 1000, train loss:1.000, valid loss:1.002
Epoch:874 / 1000, train loss:1.004, valid loss:0.995
Epoch:875 / 1000, train loss:1.004, valid loss:0.999
Epoch:876 / 1000, train loss:1.008, valid loss:1.022
Epoch:877 / 1000, train loss:1.007, valid loss:1.119
Epoch:878 / 1000, train loss:1.008, valid loss:1.047
Epoch:879 / 1000, train loss:1.007, valid loss:1.085
Epoch:880 / 1000, train loss:1.006, valid loss:0.984
Epoch:881 / 1000, train loss:1.008, valid loss:0.993
Epoch:882 / 1000, train loss:1.001, valid loss:1.056
Epoch:883 / 1000, train loss:1.019, valid loss:0.960
Epoch:884 / 1000, train loss:1.019, valid loss:1.040
Epoch:885 / 1000, train loss:1.014, valid loss:1.068
Epoch:886 / 1000, train loss:1.017, valid loss:1.060
Epoch:887 / 1000, train loss:1.006, valid loss:1.033
Epoch:888 / 1000, train loss:1.006, valid loss:0.939
Epoch:889 / 1000, train loss:1.016, valid loss:0.991
Epoch:890 / 1000, train loss:1.005, valid loss:1.046
Epoch:891 / 1000, train loss:1.011, valid loss:1.000
Epoch:892 / 1000, train loss:1.010, valid loss:0.988
Epoch:893 / 1000, train loss:1.002, valid loss:1.067
Epoch:894 / 1000, train loss:1.006, valid loss:1.002
Epoch:895 / 1000, train loss:1.006, valid loss:1.067
Epoch:896 / 1000, train loss:1.005, valid loss:0.943
Epoch:897 / 1000, train loss:1.016, valid loss:1.128
Epoch:898 / 1000, train loss:1.018, valid loss:0.968
Epoch:899 / 1000, train loss:1.005, valid loss:0.953
Epoch:900 / 1000, train loss:1.006, valid loss:0.970
Epoch:901 / 1000, train loss:1.006, valid loss:1.023
Epoch:902 / 1000, train loss:1.012, valid loss:1.044
Epoch:903 / 1000, train loss:1.011, valid loss:1.060
Epoch:904 / 1000, train loss:1.009, valid loss:1.002
Epoch:905 / 1000, train loss:1.004, valid loss:1.155
Epoch:906 / 1000, train loss:1.014, valid loss:1.018
Epoch:907 / 1000, train loss:1.016, valid loss:0.951
Epoch:908 / 1000, train loss:1.018, valid loss:1.171
Epoch:909 / 1000, train loss:1.014, valid loss:1.132
Epoch:910 / 1000, train loss:1.011, valid loss:0.964
Epoch:911 / 1000, train loss:1.008, valid loss:1.072
Epoch:912 / 1000, train loss:1.009, valid loss:1.079
Epoch:913 / 1000, train loss:1.013, valid loss:1.058
Epoch:914 / 1000, train loss:1.003, valid loss:0.973
Epoch:915 / 1000, train loss:1.010, valid loss:0.995
Epoch:916 / 1000, train loss:1.002, valid loss:0.992
Epoch:917 / 1000, train loss:1.013, valid loss:1.046
Epoch:918 / 1000, train loss:1.006, valid loss:1.053
Epoch:919 / 1000, train loss:1.011, valid loss:0.994
Epoch:920 / 1000, train loss:1.006, valid loss:0.989
Epoch:921 / 1000, train loss:1.007, valid loss:1.067
Epoch:922 / 1000, train loss:1.015, valid loss:1.015
Epoch:923 / 1000, train loss:1.005, valid loss:0.957
Epoch:924 / 1000, train loss:1.002, valid loss:0.990
Epoch:925 / 1000, train loss:1.006, valid loss:1.031
Epoch:926 / 1000, train loss:1.012, valid loss:1.144
Epoch:927 / 1000, train loss:1.010, valid loss:0.998
Epoch:928 / 1000, train loss:1.011, valid loss:1.049
Epoch:929 / 1000, train loss:1.010, valid loss:1.037
```

```
Epoch:930 / 1000, train loss:1.010, valid loss:1.055
Epoch:931 / 1000, train loss:1.011, valid loss:1.046
Epoch:932 / 1000, train loss:1.009, valid loss:0.983
Epoch:933 / 1000, train loss:1.013, valid loss:0.986
Epoch:934 / 1000, train loss:1.018, valid loss:1.010
Epoch:935 / 1000, train loss:1.011, valid loss:1.119
Epoch:936 / 1000, train loss:1.012, valid loss:1.007
Epoch:937 / 1000, train loss:1.006, valid loss:0.974
Epoch:938 / 1000, train loss:1.005, valid loss:0.959
Epoch:939 / 1000, train loss:1.002, valid loss:1.038
Epoch:940 / 1000, train loss:1.007, valid loss:0.973
Epoch:941 / 1000, train loss:1.004, valid loss:0.931
Epoch:942 / 1000, train loss:1.014, valid loss:1.033
Epoch:943 / 1000, train loss:1.014, valid loss:1.002
Epoch:944 / 1000, train loss:1.011, valid loss:0.956
Epoch:945 / 1000, train loss:1.005, valid loss:1.135
Epoch:946 / 1000, train loss:1.005, valid loss:0.982
Epoch:947 / 1000, train loss:1.007, valid loss:1.020
Epoch:948 / 1000, train loss:1.010, valid loss:1.015
Epoch:949 / 1000, train loss:1.005, valid loss:1.034
Epoch:950 / 1000, train loss:1.009, valid loss:1.024
Epoch:951 / 1000, train loss:1.004, valid loss:0.938
Epoch:952 / 1000, train loss:1.011, valid loss:0.974
Epoch:953 / 1000, train loss:1.013, valid loss:0.997
Epoch:954 / 1000, train loss:1.013, valid loss:1.031
Epoch:955 / 1000, train loss:1.011, valid loss:1.000
Epoch:956 / 1000, train loss:1.014, valid loss:0.960
Epoch:957 / 1000, train loss:1.005, valid loss:1.082
Epoch:958 / 1000, train loss:1.015, valid loss:0.975
Epoch:959 / 1000, train loss:1.014, valid loss:1.029
Epoch:960 / 1000, train loss:1.012, valid loss:0.959
Epoch:961 / 1000, train loss:1.020, valid loss:1.108
Epoch:962 / 1000, train loss:1.012, valid loss:0.958
Epoch:963 / 1000, train loss:1.014, valid loss:1.132
Epoch:964 / 1000, train loss:1.009, valid loss:1.054
Epoch:965 / 1000, train loss:1.006, valid loss:0.974
Epoch:966 / 1000, train loss:1.009, valid loss:1.026
Epoch:967 / 1000, train loss:1.005, valid loss:0.944
Epoch:968 / 1000, train loss:1.011, valid loss:1.096
Epoch:969 / 1000, train loss:1.016, valid loss:1.015
Epoch:970 / 1000, train loss:1.021, valid loss:1.043
Epoch:971 / 1000, train loss:1.007, valid loss:0.992
Epoch:972 / 1000, train loss:1.005, valid loss:1.105
Epoch:973 / 1000, train loss:1.007, valid loss:1.014
Epoch:974 / 1000, train loss:1.015, valid loss:1.098
Epoch:975 / 1000, train loss:1.023, valid loss:1.009
Epoch:976 / 1000, train loss:1.019, valid loss:1.003
Epoch:977 / 1000, train loss:1.007, valid loss:0.998
Epoch:978 / 1000, train loss:1.009, valid loss:0.968
Epoch:979 / 1000, train loss:1.011, valid loss:1.063
Epoch:980 / 1000, train loss:1.007, valid loss:1.141
Epoch:981 / 1000, train loss:1.003, valid loss:1.080
Epoch:982 / 1000, train loss:1.018, valid loss:1.113
Epoch:983 / 1000, train loss:1.003, valid loss:0.988
Epoch:984 / 1000, train loss:1.012, valid loss:0.983
Epoch:985 / 1000, train loss:1.014, valid loss:1.016
Epoch:986 / 1000, train loss:1.010, valid loss:0.975
Epoch:987 / 1000, train loss:1.012, valid loss:1.058
Epoch:988 / 1000, train loss:1.005, valid loss:0.983
Epoch:989 / 1000, train loss:1.004, valid loss:0.991
Epoch:990 / 1000, train loss:1.012, valid loss:1.122
Epoch:991 / 1000, train loss:1.017, valid loss:0.996
```

```
Epoch:992 / 1000, train loss:1.005, valid loss:1.090
Epoch:993 / 1000, train loss:1.011, valid loss:1.070
Epoch:994 / 1000, train loss:1.010, valid loss:1.056
Epoch:995 / 1000, train loss:1.005, valid loss:1.008
Epoch:996 / 1000, train loss:1.017, valid loss:1.022
Epoch:997 / 1000, train loss:1.008, valid loss:0.963
Epoch:998 / 1000, train loss:1.018, valid loss:0.978
Epoch:999 / 1000, train loss:1.005, valid loss:1.005
Epoch:1000 / 1000, train loss:1.004, valid loss:1.010
```

In [34]:
```
plt.plot(val_loss_hist)
```

Out[34]:
```
[<matplotlib.lines.Line2D at 0x7fbcfc6d8410>]
```




Check the model parameters

Notice that the parameters are very close to the formula used in generating fake data.

In [35]:
```
print([i for i in model.parameters()])
```

```
[Parameter containing:
tensor([[1.5001, 2.0019]], device='cuda:0', requires_grad=True), Parameter containing:
tensor([0.9342], device='cuda:0', requires_grad=True)]
```

## Test result

In [36]:
```
ys = []
yhats = []
model.eval()

for x, y in test_loader:

  # move batch to device
  x = x.to(device)

  # forward
  y_hat = model(x).squeeze().detach().cpu().numpy() # remove data from gpu to cpu, convert to nu

  yhats.append(y_hat)      # y_hat has shape (batch,)
  ys.append(y.cpu().numpy())
```

In [37]:
```
# Concatenate the list of arrays to a single array
yhats = np.concatenate(yhats)
ys = np.concatenate(ys)
```

In [38]:

```
plt.figure(figsize=(15,5))
plt.plot(range(len(yhats)), yhats, 'r-', label='Predict')
plt.plot(range(len(ys)), ys, 'b-', label='True')
plt.legend()
plt.show()
```