generate

March 27, 2022

[1]:

```
%load_ext autoreload
%autoreload 2
```

[2]:

```
import numpy as np
import torch
import torch.nn as nn
import torch.nn.functional as F
from torch.utils.data import TensorDataset, DataLoader
import matplotlib.pyplot as plt

import generate as generate
from single_layer import *
```

0.1 Experiment 1

- Data Generation: R1 -> R1, with only one activation unit
- model
  - hidden_dim: 2
  - lr: 0.01

[3]:

```
# Constants
d = 1
N = 3#int(np.exp(d))
M = d
num = 1
T = 2000

lr = 0.01
hidden_dim = 2
```

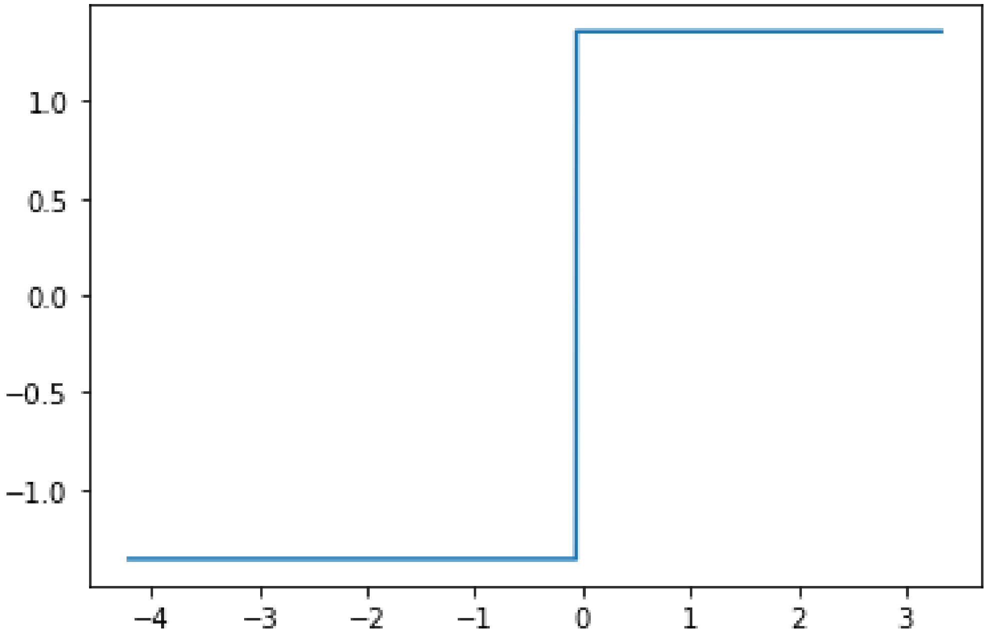
[10]:

```
(an, bn) = generate.generate_activations(d, N)
(In, thetan) = generate.generate_single_layer(N, M, d, num, an, bn)
(X, Y) = generate.generate_single_data(T, an, bn, In, thetan)
print(X.shape)
print(Y.shape)
```

```
(1, 2000, 1)
(1, 2000)
```

[11]:

```
plt.plot(*zip(*sorted(zip(X[0], Y[0]))))
plt.show()
```

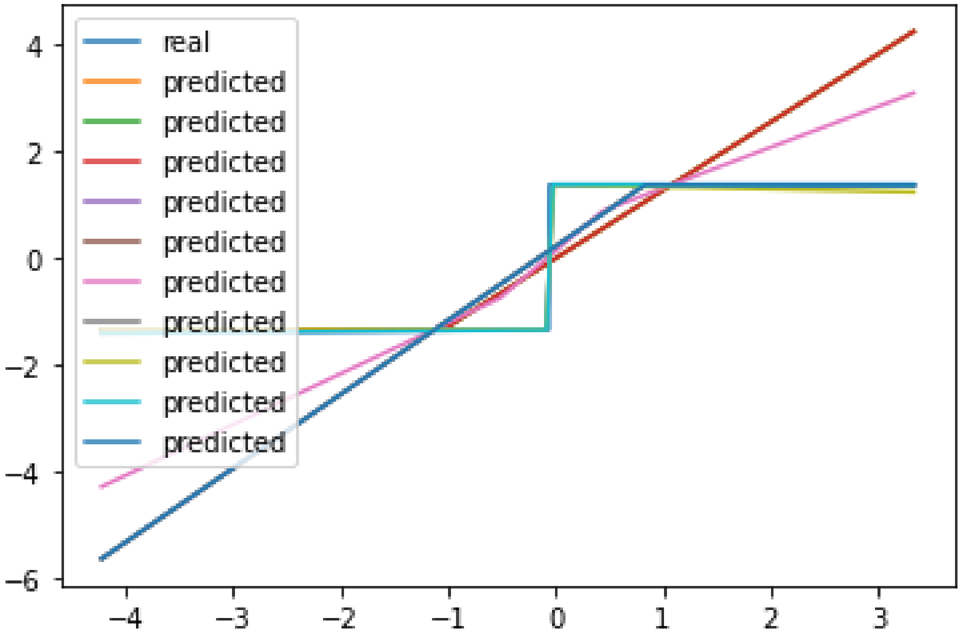


[12]:

```
num_experiments = 10
input = X[0]
plt.plot(*zip(*sorted(zip(X[0], Y[0]))))
for i in range(num_experiments):
    (model, epoch_number, best_vloss) = train_one_model(
        hidden_dim, X[0], Y[0],
        val_ratio=0.2,
        lr=lr,
        patience=100,
        epochs=1000,
    )
    print(f"epochs: {epoch_number}, validation loss: {best_vloss}")
    predicted = model(torch.Tensor(X[0])).detach().numpy()
    plt.plot(*zip(*sorted(zip(X[0], predicted))))
plt.legend(["real"] + ["predicted"] * num_experiments)
plt.show()
```

```
epochs: 366, validation loss: 0.6367917060852051
epochs: 400, validation loss: 0.6367545127868652
epochs: 392, validation loss: 0.6367880702018738
epochs: 1000, validation loss: 0.0249573215842247
epochs: 541, validation loss: 0.6444958448410034
epochs: 309, validation loss: 0.531389057636261
```

```
epochs: 414, validation loss: 0.6451221704483032
epochs: 1000, validation loss: 0.030945193022489548
epochs: 1000, validation loss: 0.024709703400731087
epochs: 310, validation loss: 0.6443562507629395
```



0.2 Experiment 2

- Data Generation: R1 -> R1, with only one activation unit
- model
  - hidden_dim: 3
  - lr: 0.01

[13]:

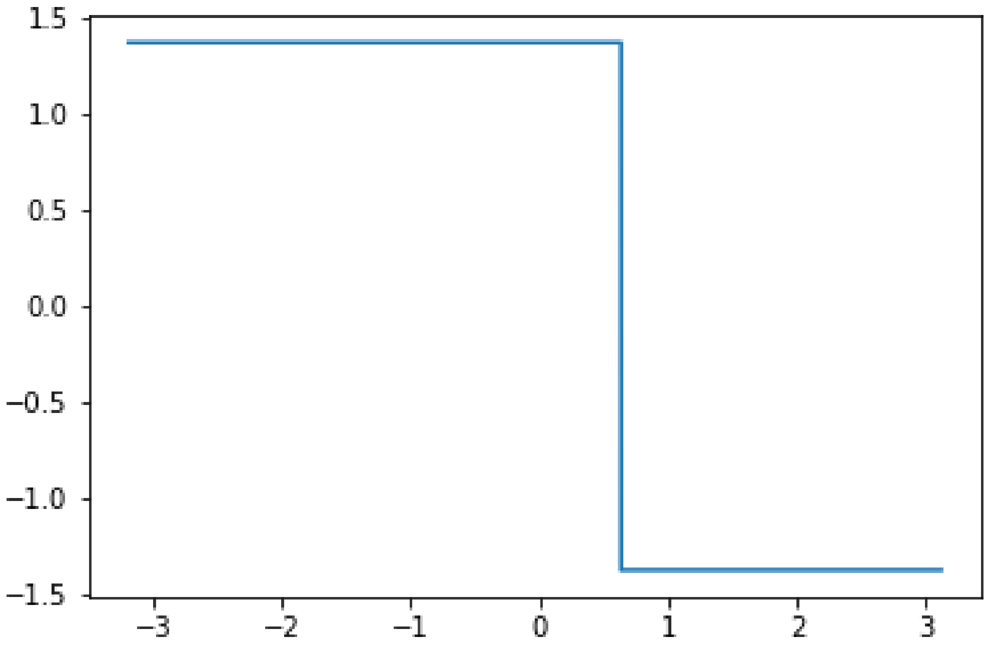
```
# Constants
d = 1
N = 3#int(np.exp(d))
M = d
num = 1
T = 2000

lr = 0.01
hidden_dim = 3
```

[14]:

```
(an, bn) = generate.generate_activations(d, N)
(In, thetan) = generate.generate_single_layer(N, M, d, num, an, bn)
(X, Y) = generate.generate_single_data(T, an, bn, In, thetan)
print(X.shape)
print(Y.shape)
```

```
(1, 2000, 1)
(1, 2000)
```

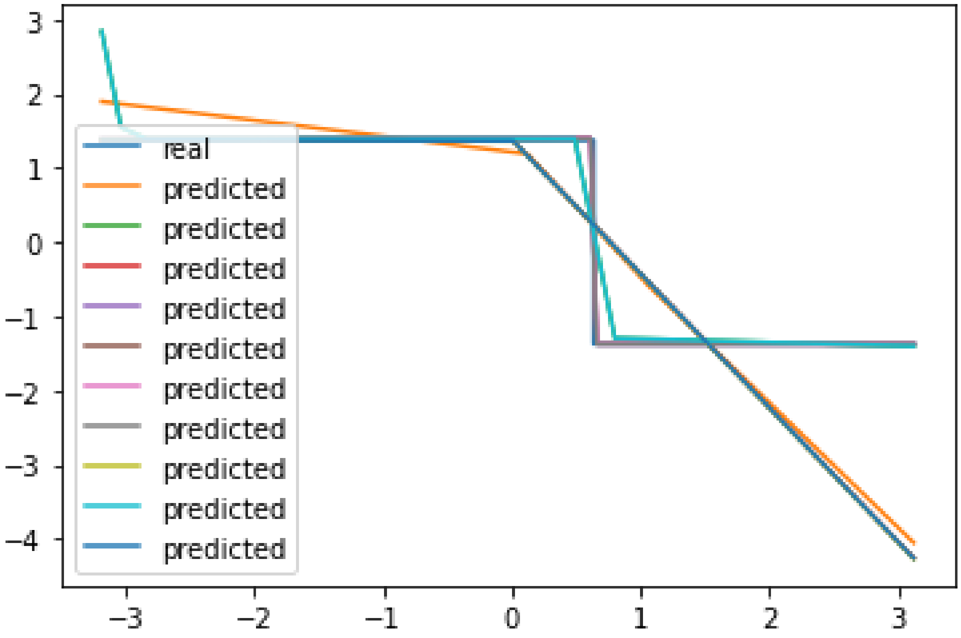
[15]:
```
plt.plot(*zip(*sorted(zip(X[0], Y[0]))))
plt.show()
```



[16]:
```
num_experiments = 10
input = X[0]
plt.plot(*zip(*sorted(zip(X[0], Y[0]))))
for i in range(num_experiments):
    (model, epoch_number, best_vloss) = train_one_model(
        hidden_dim, X[0], Y[0],
        val_ratio=0.2,
        lr=lr,
        patience=100,
        epochs=4000,
    )
    print(f"epochs: {epoch_number}, validation loss: {best_vloss}")
    predicted = model(torch.Tensor(X[0])).detach().numpy()
    plt.plot(*zip(*sorted(zip(X[0], predicted))))
plt.legend(["real"] + ["predicted"] * num_experiments)
plt.show()
```

```
epochs: 189, validation loss: 0.33814457058906555
```

```
epochs: 4000, validation loss: 0.04905490577220917
epochs: 120, validation loss: 0.3504645824432373
epochs: 4000, validation loss: 0.005515805445611477
epochs: 4000, validation loss: 0.005026805214583874
epochs: 3107, validation loss: 0.006231104023754597
epochs: 4000, validation loss: 0.0037900456227362156
epochs: 137, validation loss: 0.35054540634155273
epochs: 4000, validation loss: 0.04905082285404205
epochs: 118, validation loss: 0.3504469394683838
```



0.3 Experiment 3

- Data Generation: R1 -> R1, with only one activation unit
- model
  - hidden_dim: 4
  - lr: 0.01

[17]:
```
# Constants
d = 1
N = 3#int(np.exp(d))
M = d
num = 1
T = 2000

lr = 0.01
hidden_dim = 4
```

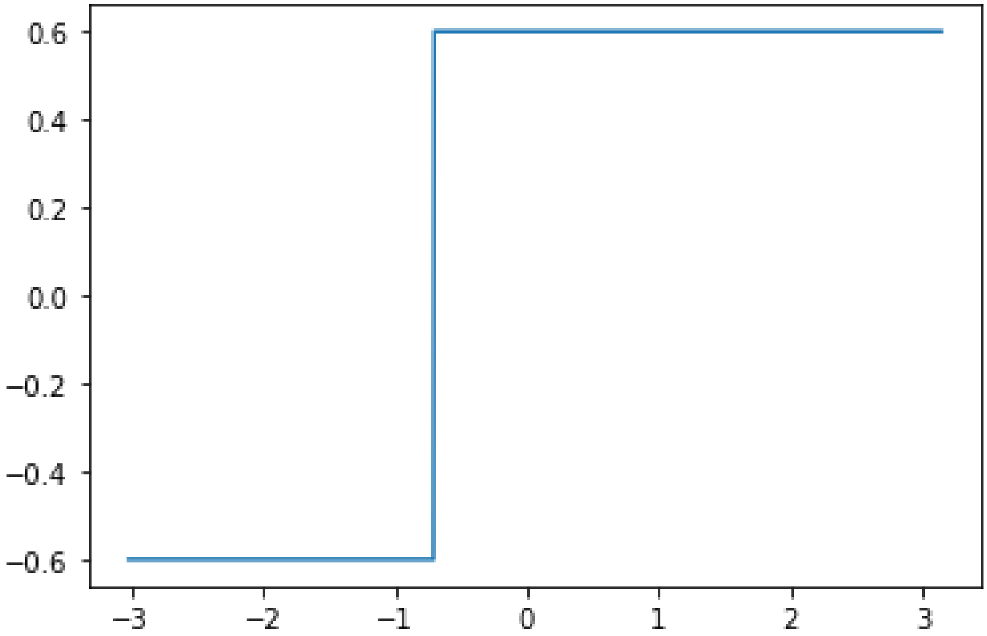
[18]:

```
(an, bn) = generate.generate_activations(d, N)
(In, thetan) = generate.generate_single_layer(N, M, d, num, an, bn)
(X, Y) = generate.generate_single_data(T, an, bn, In, thetan)
print(X.shape)
print(Y.shape)
```

```
(1, 2000, 1)
(1, 2000)
```

[19]:

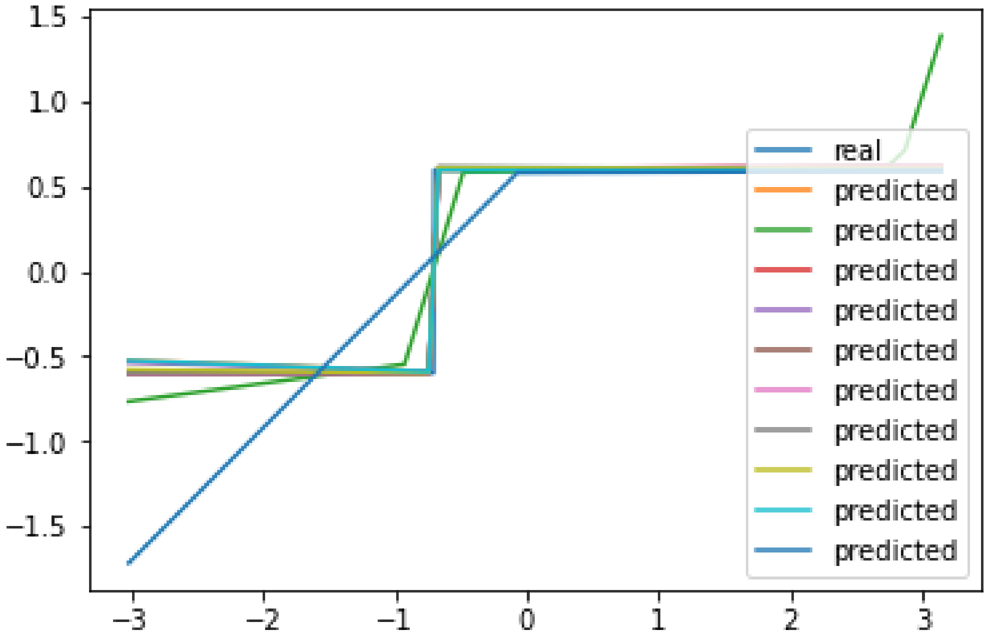
```
plt.plot(*zip(*sorted(zip(X[0], Y[0]))))
plt.show()
```



[20]:

```
num_experiments = 10
input = X[0]
plt.plot(*zip(*sorted(zip(X[0], Y[0]))))
for i in range(num_experiments):
    (model, epoch_number, best_vloss) = train_one_model(
        hidden_dim, X[0], Y[0],
        val_ratio=0.2,
        lr=lr,
        patience=100,
        epochs=1000,
    )
    print(f"epochs: {epoch_number}, validation loss: {best_vloss}")
    predicted = model(torch.Tensor(X[0])).detach().numpy()
    plt.plot(*zip(*sorted(zip(X[0], predicted))))
plt.legend(["real"] + ["predicted"] * num_experiments)
plt.show()
```

```
epochs: 1000, validation loss: 0.0015389877371490002
epochs: 1000, validation loss: 0.015289440751075745
epochs: 1000, validation loss: 0.0015104215126484632
epochs: 1000, validation loss: 0.0012099889572709799
epochs: 1000, validation loss: 0.001457865466363728
epochs: 1000, validation loss: 0.001496882294304669
epochs: 1000, validation loss: 0.0015868310583755374
epochs: 1000, validation loss: 0.0013930064160376787
epochs: 1000, validation loss: 0.0015373507048934698
epochs: 153, validation loss: 0.05847661942243576
```



0.4 Experiment 4

- Data Generation: R1 -> R1, with two activation units
- model
  - hidden_dim: 4
  - lr: 0.01

[21]:

```
# Constants
d = 1
N = 3#int(np.exp(d))
M = 2
num = 1
```

```
T = 2000

lr = 0.01
hidden_dim = 4
```

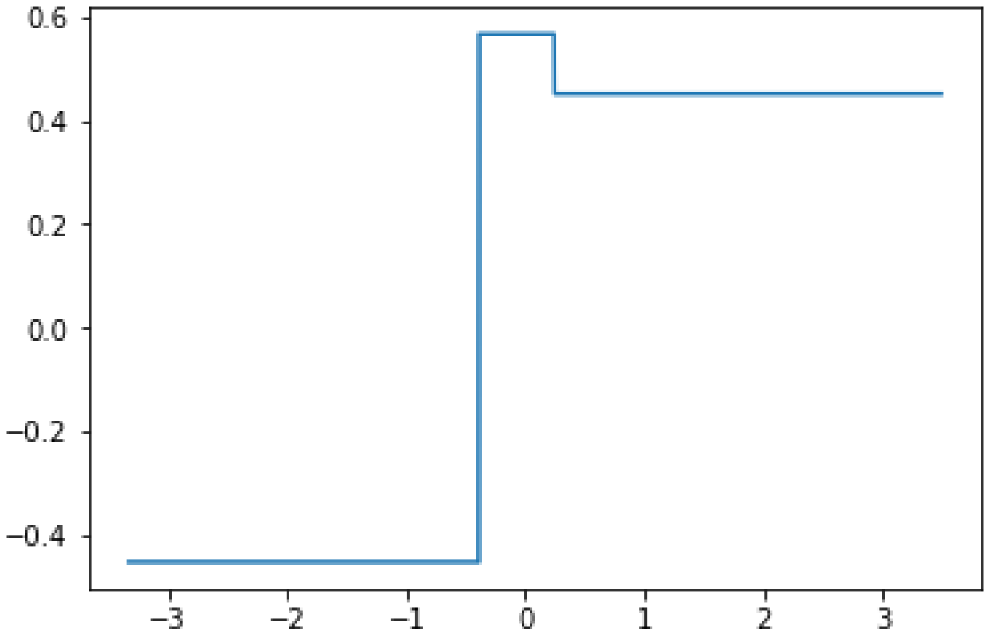
[27]:

```
(an, bn) = generate.generate_activations(d, N)
(In, thetan) = generate.generate_single_layer(N, M, d, num, an, bn)
(X, Y) = generate.generate_single_data(T, an, bn, In, thetan)
print(X.shape)
print(Y.shape)
```

```
(1, 2000, 1)
(1, 2000)
```

[28]:

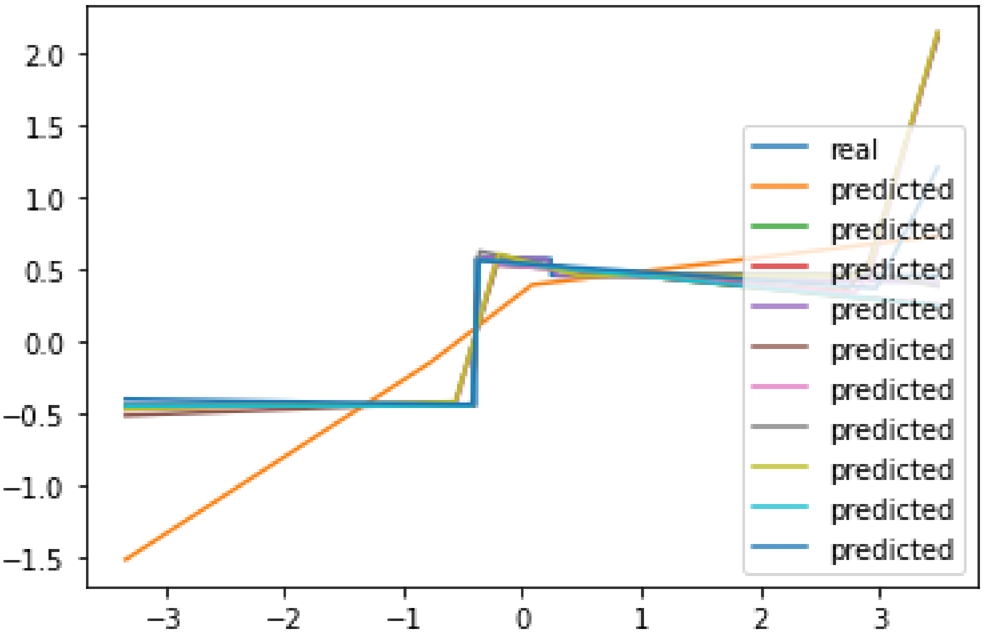
```
plt.plot(*zip(*sorted(zip(X[0], Y[0]))))
plt.show()
```



[29]:

```
num_experiments = 10
input = X[0]
plt.plot(*zip(*sorted(zip(X[0], Y[0]))))
for i in range(num_experiments):
    (model, epoch_number, best_vloss) = train_one_model(
        hidden_dim, X[0], Y[0],
        val_ratio=0.2,
        lr=lr,
        patience=100,
        epochs=2000,
    )
    print(f"epochs: {epoch_number}, validation loss: {best_vloss}")
    predicted = model(torch.Tensor(X[0])).detach().numpy()
    plt.plot(*zip(*sorted(zip(X[0], predicted))))
plt.legend(["real"] + ["predicted"] * num_experiments)
plt.show()
```

```
epochs: 108, validation loss: 0.061039477586746216
epochs: 2000, validation loss: 0.0026688140351325274
epochs: 2000, validation loss: 0.002557485830038786
epochs: 2000, validation loss: 0.001267323736101389
epochs: 1226, validation loss: 0.025260601192712784
epochs: 1423, validation loss: 0.02478194423019886
epochs: 2000, validation loss: 0.0017017138889059424
epochs: 1293, validation loss: 0.024264130741357803
epochs: 2000, validation loss: 0.002609127899631858
epochs: 2000, validation loss: 0.0044997879303991795
```



0.5 Experiment 5

- Data Generation: R1 -> R1, with two activation units
- model
  - hidden_dim: 8
  - lr: 0.01

[30]:

```
# Constants
d = 1
N = 3#int(np.exp(d))
M = 2
num = 1
T = 2000

lr = 0.01
hidden_dim = 8
```

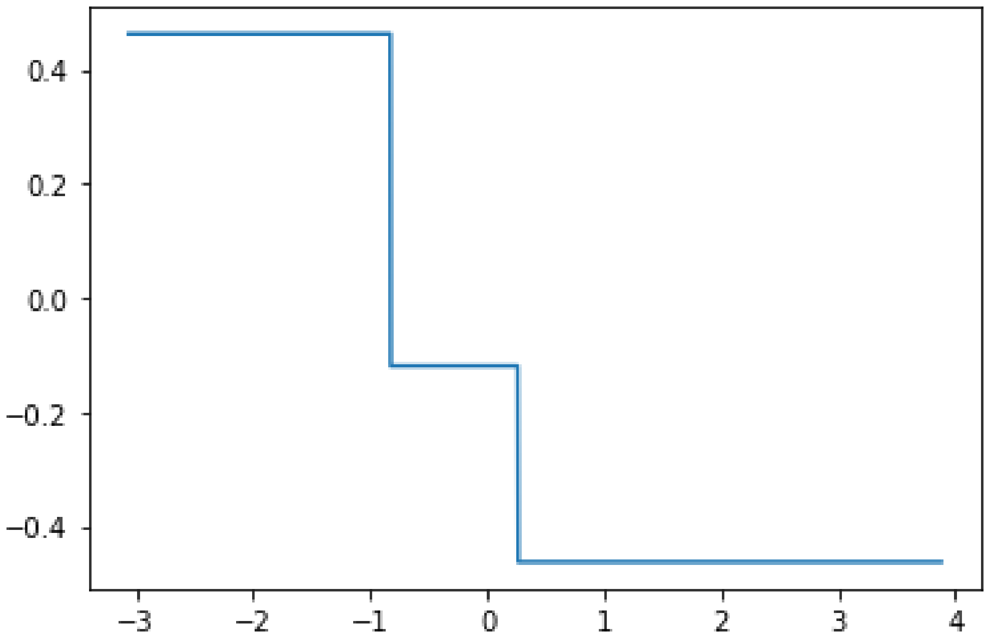
[33]:

```
(an, bn) = generate.generate_activations(d, N)
(In, thetan) = generate.generate_single_layer(N, M, d, num, an, bn)
(X, Y) = generate.generate_single_data(T, an, bn, In, thetan)
print(X.shape)
print(Y.shape)
```

```
(1, 2000, 1)
(1, 2000)
```

[34]:

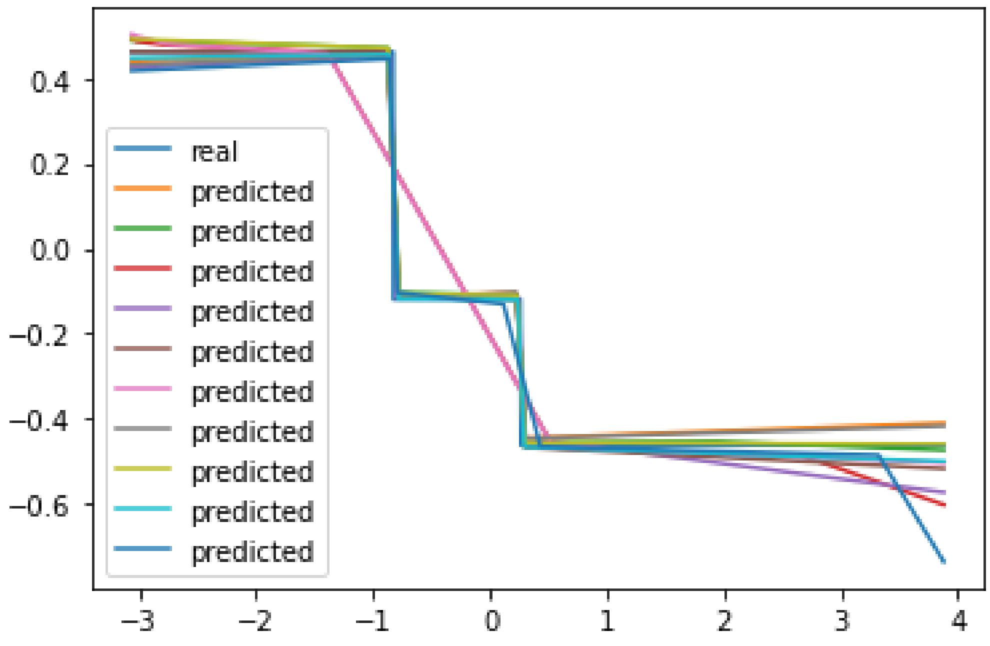
```
plt.plot(*zip(*sorted(zip(X[0], Y[0]))))
plt.show()
```



[35]:

```
num_experiments = 10
input = X[0]
plt.plot(*zip(*sorted(zip(X[0], Y[0]))))
for i in range(num_experiments):
    (model, epoch_number, best_vloss) = train_one_model(
        hidden_dim, X[0], Y[0],
        val_ratio=0.2,
        lr=lr,
        patience=100,
        epochs=2000,
    )
    print(f"epochs: {epoch_number}, validation loss: {best_vloss}")
    predicted = model(torch.Tensor(X[0])).detach().numpy()
    plt.plot(*zip(*sorted(zip(X[0], predicted))))
plt.legend(["real"] + ["predicted"] * num_experiments)
plt.show()
```

```
epochs: 1662, validation loss: 0.0006806793389841914
epochs: 1607, validation loss: 0.0006721923127770424
epochs: 259, validation loss: 0.012223339639604092
epochs: 2000, validation loss: 0.0005429112352430 82
epochs: 1738, validation loss: 0.0006583314971067011
epochs: 248, validation loss: 0.012197546660900116
epochs: 994, validation loss: 0.0008134039817377925
epochs: 1927, validation loss: 0.0006438980344682932
epochs: 1981, validation loss: 0.0005194177501834929
epochs: 2000, validation loss: 0.0017564221052452922
```



[ ]: