APM 598 Homework 2 - 3rd March 2023

Siddharth Jain 1226137070

Ketan Choudhary 1226082301

Pranav Chougule 1225934595

Q 1A

In [ ]:
```
import tensorflow.keras as keras
import numpy as np
import gc
print("Q 1A")
#to clear previous TF cached graph
gc.collect()

x = np.array([[1,0], [-1,0], [0,1], [0,-1]]) #X1, X2, X3, X4
y = np.array([[1,0], [1,0], [0,1], [0,1]])   #S1, S1, S2, S2

#model
model = keras.Sequential([
    keras.layers.Dense(2, input_shape=(2,), activation='relu', kernel_initializer='
    keras.layers.Dense(2, activation=None, kernel_initializer='glorot_uniform')
])

#training the model
optimizer = keras.optimizers.Adam(learning_rate=0.001)
model.compile(loss='mean_absolute_error', optimizer=optimizer, metrics=None)
model.fit(x, y, epochs=10000, verbose=0)

#Get Weights and biases of layer 1
weights_1, biases_1 = model.layers[0].get_weights()

#Get Weights and biases of layer 2
weights_2, biases_2 = model.layers[1].get_weights()

print("\nWeights for layer1:\n",weights_1)
print("Biases  for layer1:\n",biases_1,"\n")
print("Weights for layer2:\n",weights_2)
print("Biases  for layer2:\n",biases_2,"\n")

print("Score[s1, s2] for X1 [ 1,0]: ",model.predict(np.array([x[0]])),"\n")
print("Score[s1, s2] for X2 [-1,0]: ",model.predict(np.array([x[1]])),"\n")
print("Score[s1, s2] for X3 [ 0,1]: ",model.predict(np.array([x[2]])),"\n")
print("Score[s1, s2] for X4 [ 0,-1]: ",model.predict(np.array([x[3]])),"\n")

del model
gc.collect()
```

```
Q 1A
training...

Weights for layer1:
 [[ 1.1349535  -0.2115434 ]
 [ 0.33902076  0.9033544 ]]
Biases  for layer1:
 [-0.31839803 -0.21323949]

Weights for layer2:
 [[ 0.58303607 -0.67433393]
 [-0.77666426  0.6701857 ]]
Biases  for layer2:
 [0.52405316 0.55094624]

1/1 [==============================] - 0s 93ms/step
Score[s1, s2] for X1 [ 1,0]:  [[1.0001345e+00 3.1512976e-04]]

1/1 [==============================] - 0s 25ms/step
Score[s1, s2] for X2 [-1,0]:  [[0.52405316 0.55094624]]

1/1 [==============================] - 0s 24ms/step
Score[s1, s2] for X3 [ 0,1]:  [[8.9347363e-05 9.9954474e-01]]

1/1 [==============================] - 0s 23ms/step
Score[s1, s2] for X4 [ 0,-1]:  [[0.52405316 0.55094624]]
```

Out[ ]: 4640

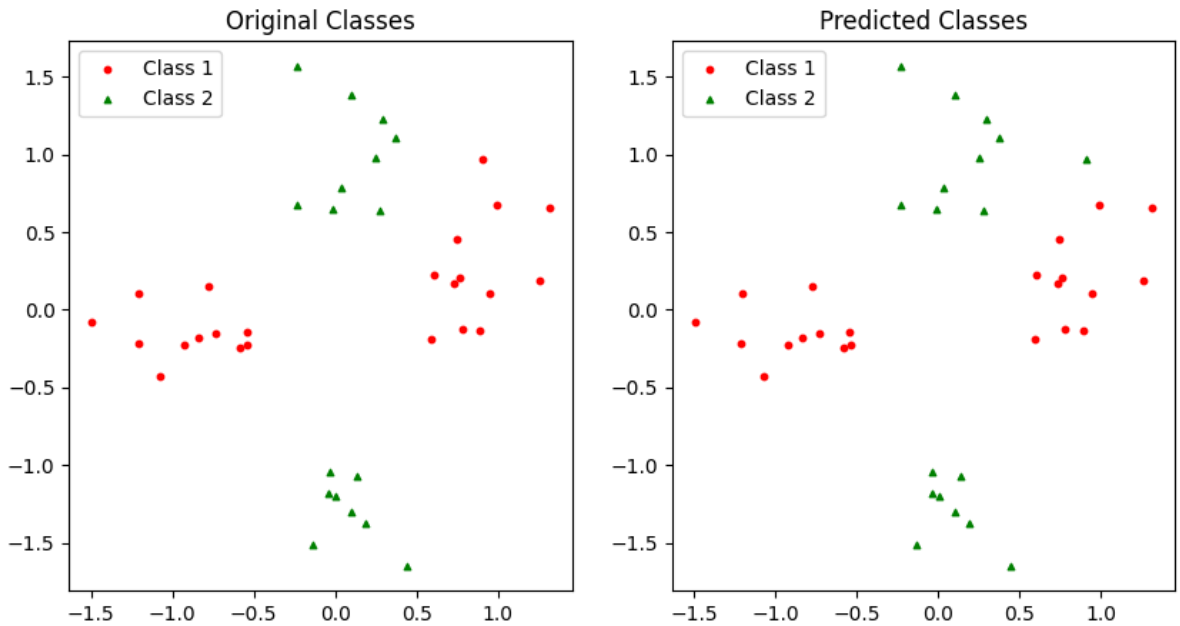
Q 1B

In [ ]:

```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.utils import to_categorical
import gc
print("Q 1B")
# Load and shuffle dataset
df = pd.read_csv('C:\\Users\\siddh\\Documents\\Siddharth\\Classes\\Spring 2023\\APM

# Split into features and classes
x = np.column_stack((df['x1'].values,df['x2'].values))
y = df['class'].values

# Split into training and testing sets
split_ratio = 0.8
split_index = int(split_ratio * len(x))
x_train, x_test = x[:split_index], x[split_index:]
y_train, y_test = y[:split_index], y[split_index:]

# Define the model architecture
model = Sequential([
```

```
    Dense(4, activation='relu', input_shape=(2,)),
    Dense(3, activation='softmax')
])

# Compile the model
model.compile(optimizer=Adam(learning_rate=0.05),
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])
model.summary
# Train the model
history = model.fit(x_train, y_train, epochs=1000, verbose=0)

# Evaluate the model on the test set
score, accuracy = model.evaluate(x_test, y_test)

# Predict classes for the test set
y_pred = np.argmax(model.predict(x_test), axis=1)

# Create a subplot to display the original classes and the predicted classes
fig, ax = plt.subplots(1, 2, figsize=(10, 5))

# Plot the original classes
ax[0].scatter(x_test[y_test == 1][:, 0], x_test[y_test == 1][:, 1], s=10, c='r', ma
ax[0].scatter(x_test[y_test == 2][:, 0], x_test[y_test == 2][:, 1], s=10, c='g', ma
ax[0].set_title("Original Classes")
ax[0].legend(loc='upper left')

# Plot the predicted classes
ax[1].scatter(x_test[y_pred == 1][:, 0], x_test[y_pred == 1][:, 1], s=10, c='r', ma
ax[1].scatter(x_test[y_pred == 2][:, 0], x_test[y_pred == 2][:, 1], s=10, c='g', ma
ax[1].set_title("Predicted Classes")
ax[1].legend(loc='upper left')

plt.show()

# Perform garbage collection to free up memory
gc.collect()
```

```
2/2 [==============================] - 0s 3ms/step - loss: 0.0424 - accuracy: 0.97
50
2/2 [==============================] - 0s 1ms/step
```



Out[ ]: 1567

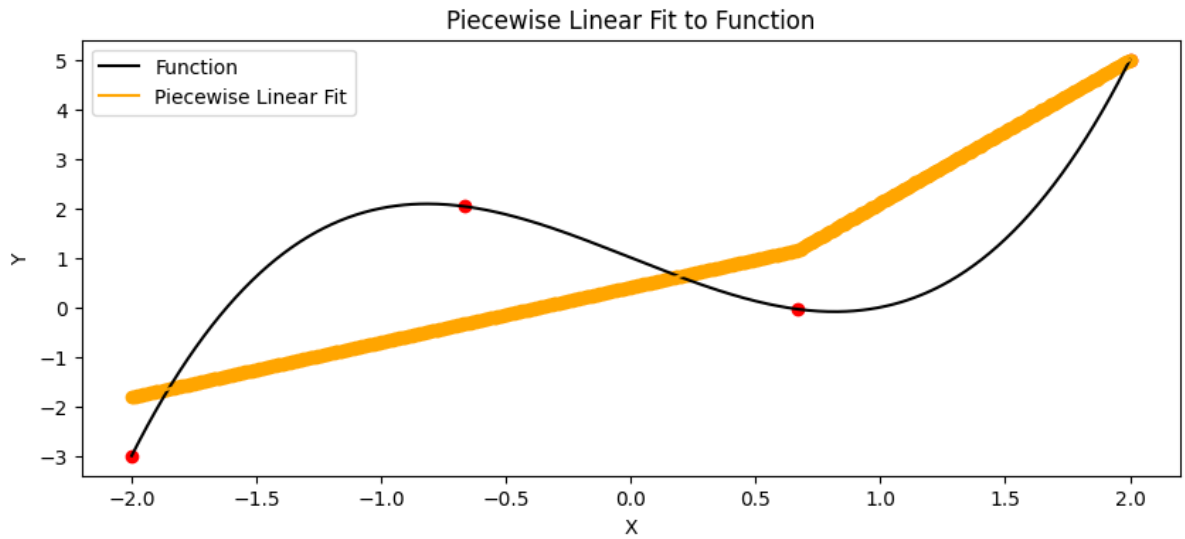
Q 2A and 2B

In [ ]:
```
import torch
from torch.utils.data import DataLoader, TensorDataset
import torch.nn as nn
import torch.optim as optim

import matplotlib.pyplot as plt
import numpy as np
print("Q2A and Q2B")
'''Define any function to be used as input'''
def f(x):
    return x**3 - 2*x + 1

def get_data_points(f, start, end, num_points):
    '''Generate data points'''
    x = torch.linspace(start, end, num_points).reshape(-1, 1)
    y = f(x)
    return x, y

def get_plotting_points(f, start, end, num_points):
    '''Generate points for plotting the function'''
    x = torch.linspace(start, end, num_points).reshape(-1, 1)
    y = f(x)
    return x, y

class Net(nn.Module):
    def __init__(self, n_feature, n_hidden, n_output):
        super(Net, self).__init__()
        self.hidden = nn.Linear(n_feature, n_hidden)
        self.predict = nn.Linear(n_hidden, n_output)

    def forward(self, x):
        x = nn.functional.relu(self.hidden(x)) # ReLU activation
```

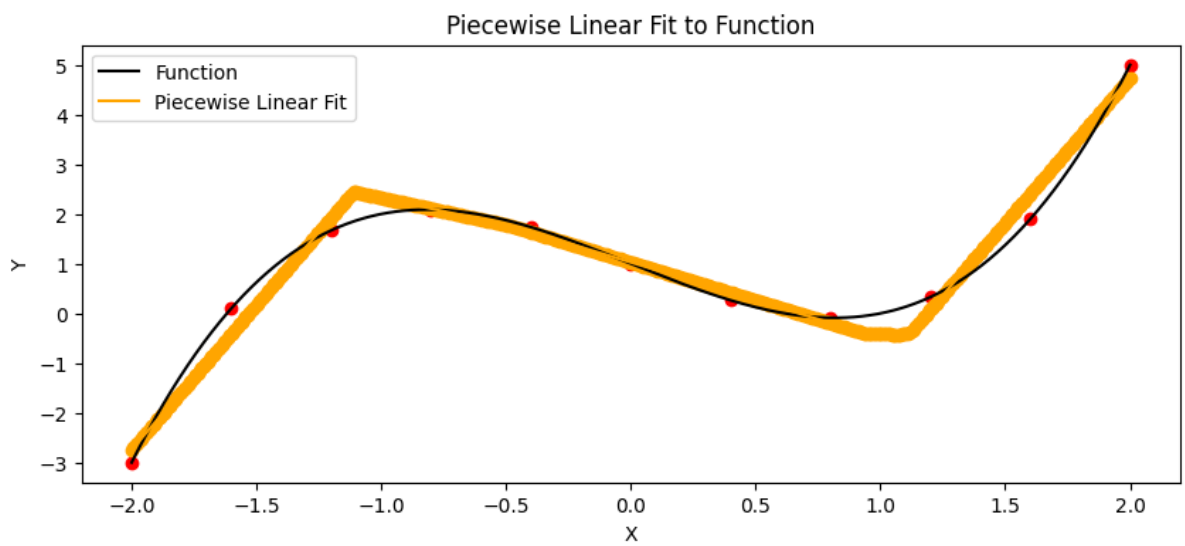
```
        x = self.predict(x)
        return x

def train_model(model, optimizer, criterion, dataloader, num_epochs):
    '''Train the neural network model'''
    losses = []
    for epoch in range(num_epochs):
        epoch_loss = 0
        for x, y in dataloader:
            optimizer.zero_grad()
            y_pred = model(x)
            loss = criterion(y_pred, y)
            loss.backward()
            optimizer.step()
            epoch_loss += loss.item()
        losses.append(epoch_loss)
    return model, losses

def piecewise_linear(f, start, end, num_points):
    '''Fit a piecewise linear function to the input function'''
    x, y = get_data_points(f, start, end, num_points+1)
    dataset = TensorDataset(x, y)
    dataloader = DataLoader(dataset, batch_size=32, shuffle=True)
    model = Net(n_feature=1, n_hidden=num_points+1, n_output=1)
    #print(model)
    optimizer = optim.Adam(model.parameters(), lr=0.01)
    criterion = nn.MSELoss()
    trained_model, losses = train_model(model, optimizer, criterion, dataloader, nu
    p_x, p_y = get_plotting_points(f, start, end, 500)
    y_pred = trained_model(p_x)
    #print("\nThe predicted values for the given X: ", y_pred.data.numpy())
    plt.figure(figsize=(10,4))
    plt.plot(p_x.data.numpy(), p_y.data.numpy(), color="black", label='Function')
    plt.scatter(x.data.numpy(), y.data.numpy(), color="red")
    plt.scatter(p_x.data.numpy(), y_pred.data.numpy(), color="orange")
    plt.plot(p_x.data.numpy(), y_pred.data.numpy(), color="orange", label='Piecewis
    plt.title('Piecewise Linear Fit to Function')
    plt.xlabel('X')
    plt.ylabel('Y')
    plt.legend()
    plt.show()
    print("Trained Network:\nNumber of nodes in hidden layer: {} (num_points+1)".fo
    for name, param in trained_model.named_parameters():
        if param.requires_grad:
            print(name, param.data)
    return trained_model, losses

def main():
    print()
    print("Number of points is 3")
    piecewise_linear(f, start=-2, end=2, num_points=3)
    print()
    print("Number of points is 10")
    piecewise_linear(f, start=-2, end=2, num_points=10)
    print()
    print("Number of points is 25")
```

```
    piecewise_linear(f, start=-2, end=2, num_points=25)

if __name__ == '__main__':
    main()
```

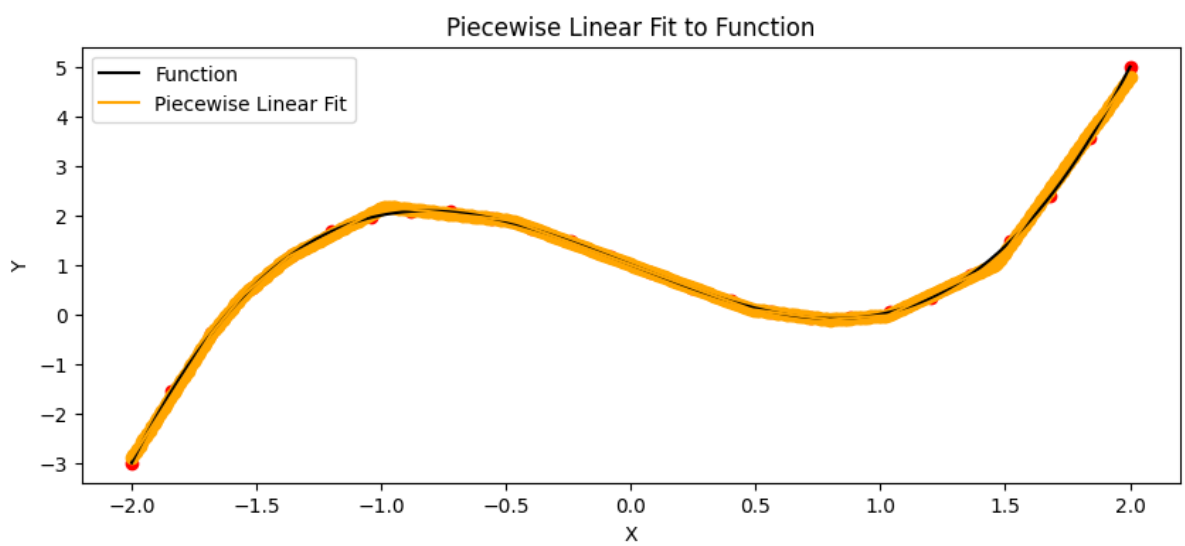
```
Q2A and Q2B

Number of points is 3
```




```
Trained Network:
Number of nodes in hidden layer: 4 (num_points+1)
hidden.weight tensor([[-0.7296],
        [-0.4928],
        [ 0.5730],
        [ 1.4541]])
hidden.bias tensor([ 2.2860, -0.9859,  1.4960, -0.9709])
predict.weight tensor([[-0.7274,  0.4172,  1.0129,  1.2235]])
predict.bias tensor([0.5552])

Number of points is 10
```



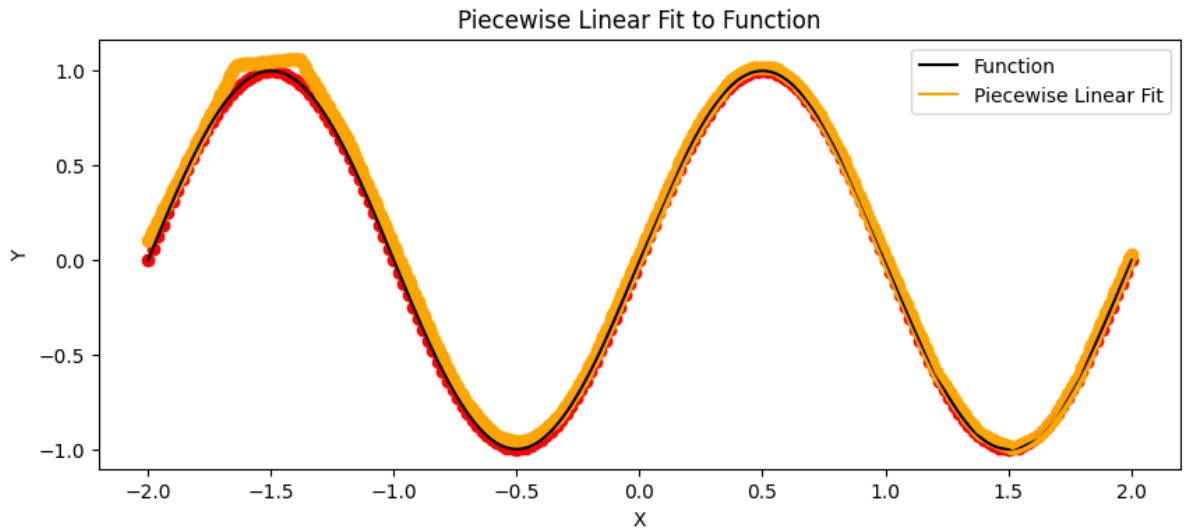
```
Trained Network:
Number of nodes in hidden layer: 11 (num_points+1)
hidden.weight tensor([[-2.9252e-01],
        [ 1.4850e+00],
        [ 2.5294e+00],
        [-2.4116e+00],
        [-3.2056e-01],
        [-6.7032e-02],
        [ 1.0758e+00],
        [-1.7669e-01],
        [-2.1668e-03],
        [-1.0428e+00],
        [-5.0253e-01]])
hidden.bias tensor([-0.8242, -1.4057, -2.8359, -2.6792, -0.8386,  1.3245,  0.5137,
-0.6384,
         0.8700,  1.1278,  0.5541])
predict.weight tensor([[-0.2468,  0.9945,  1.9004, -2.8725,  0.1702,  0.1193, -0.4
080, -0.0659,
         -0.1635,  0.8689,  0.3476]])
predict.bias tensor([0.0497])

Number of points is 25
```



```
Trained Network:
Number of nodes in hidden layer: 26 (num_points+1)
hidden.weight tensor([[ 0.1231],
        [-1.3295],
        [-0.3945],
        [ 0.1702],
        [-0.0128],
        [-0.5845],
        [ 0.1512],
        [ 0.4325],
        [-1.3151],
        [-1.4069],
        [-0.7412],
        [ 2.0664],
        [-1.3666],
        [-1.1234],
        [-1.3033],
        [-0.1646],
        [ 0.5669],
        [ 0.3191],
        [ 0.4806],
        [ 0.4675],
        [ 0.3582],
        [-1.5539],
        [-1.3051],
        [ 1.5045],
        [-2.0918],
        [-0.4430]])
hidden.bias tensor([ 0.2466, -2.0664, -0.8820,  0.3408,  0.9250,  1.1730,  0.3026,
-1.0735,
        -1.2935, -0.7046,  1.4904, -3.0273, -0.6212, -1.8874,  1.0394, -0.5029,
         1.1341,  0.6383,  0.8087,  0.9352,  0.7165, -1.5368, -1.7902, -0.7267,
         2.1519,  0.8933])
predict.weight tensor([[ 0.1552, -1.3944, -0.0565,  0.1437, -0.0930, -0.4087,  0.0
743,  0.0217,
         -1.1284, -0.4970, -0.7811,  2.2568, -0.4514, -1.4664,  0.7799,  0.0298,
          0.1510,  0.1603, -0.2381,  0.1052,  0.1210, -1.0932, -1.2495,  0.8080,
          0.9719, -0.4346]])
predict.bias tensor([-0.1611])
```

Piecewise Interpolation

In [ ]:
```
gx = lambda x: np.sin(np.pi * x)
piecewise_linear(gx, start=-2, end=2, num_points=200)

gx = lambda x: np.cosh(np.pi * x)
piecewise_linear(gx, start=-2, end=2, num_points=200)
```


Piecewise Linear Fit to Function
Function
Piecewise Linear Fit
1.0
0.5
0.0
−0.5
−1.0
Y
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0
X

```
Trained Network:
Number of nodes in hidden layer: 201 (num_points+1)
hidden.weight tensor([[ 1.1428e+00],
        [-4.0097e-01],
        [ 6.7460e-01],
        [-1.1757e+00],
        [ 8.2221e-01],
        [ 2.3316e-01],
        [ 4.9950e-01],
        [ 7.7658e-01],
        [ 1.2396e-02],
        [ 2.5632e-01],
        [ 3.9102e-02],
        [-5.2259e-01],
        [ 2.1545e-01],
        [ 5.1589e-01],
        [-4.3426e-01],
        [-4.2778e-01],
        [ 3.5306e-01],
        [ 2.2099e-01],
        [-7.7168e-01],
        [-4.9583e-01],
        [ 1.0412e-01],
        [ 7.2305e-01],
        [ 7.5443e-01],
        [-1.8237e-01],
        [ 6.3562e-01],
        [ 8.3488e-01],
        [ 7.8570e-01],
        [ 5.2400e-01],
        [-2.2192e-01],
        [-3.6841e-01],
        [-6.1150e-01],
        [-7.6322e-01],
        [ 9.6353e-01],
        [ 3.3976e-01],
        [ 7.6125e-02],
        [-3.5581e-01],
        [-1.0646e+00],
        [ 8.2080e-01],
        [ 7.4222e-02],
        [ 1.6860e-01],
        [-3.7939e-01],
        [-1.8762e-01],
        [-5.1402e-02],
        [-1.8199e-01],
        [ 6.7047e-01],
        [-4.0811e-02],
        [-6.7726e-01],
        [-1.0353e+00],
        [ 6.0465e-03],
        [ 7.7263e-01],
        [-4.7207e-01],
        [ 4.7230e-01],
        [-2.2276e-01],
        [-2.9446e-03],
```

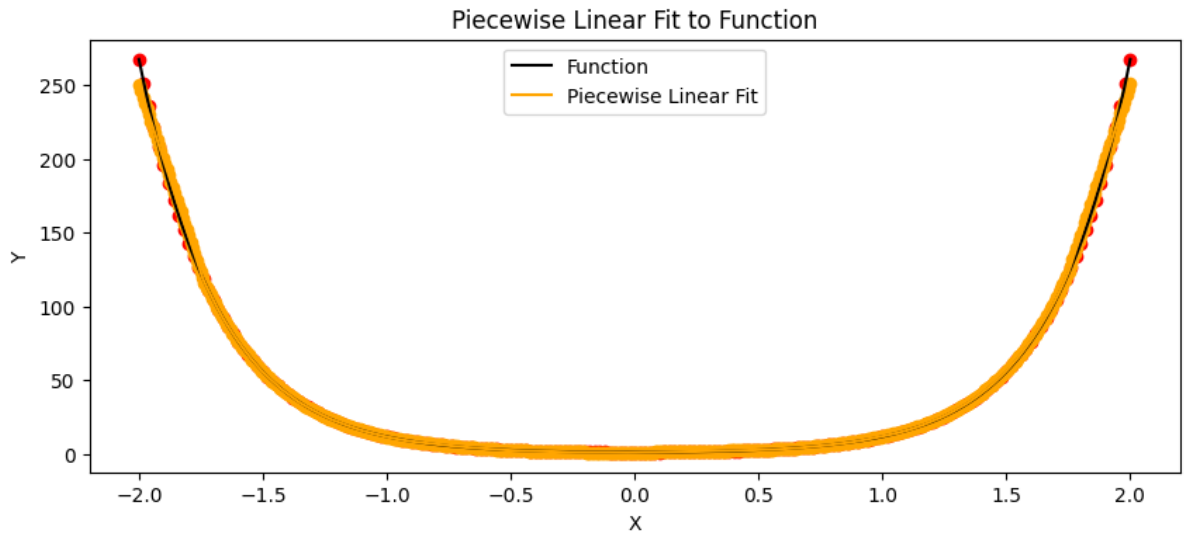
```
[ 6.2826e-01],
[-2.0091e-01],
[ 3.0190e-01],
[-6.3680e-01],
[ 4.9396e-02],
[-5.6465e-02],
[-9.3641e-01],
[-1.1051e-01],
[ 9.0438e-01],
[-1.0196e-01],
[ 6.5596e-01],
[ 8.3284e-01],
[-4.5256e-01],
[ 4.3657e-01],
[ 1.5623e-03],
[-2.0004e-01],
[ 6.9031e-01],
[ 2.4658e-02],
[ 2.1600e-01],
[-4.5282e-01],
[ 5.1173e-01],
[-4.8091e-01],
[-6.7666e-01],
[-1.7990e-01],
[-1.0049e+00],
[ 8.2359e-01],
[-1.9232e-01],
[-1.1909e-01],
[-1.1777e-01],
[-2.8323e-01],
[-7.7666e-01],
[ 2.8742e-01],
[ 1.0881e+00],
[-1.1462e+00],
[ 3.9003e-01],
[-4.2814e-02],
[-8.4198e-01],
[-6.7707e-01],
[ 3.0062e-01],
[-7.8843e-01],
[-3.2250e-01],
[-8.4527e-02],
[ 1.4274e-02],
[-1.2578e+00],
[ 7.6225e-01],
[ 4.4621e-01],
[-2.9437e-01],
[ 6.3414e-01],
[ 3.9945e-01],
[-1.0031e+00],
[-8.5494e-07],
[ 8.0699e-01],
[ 6.3712e-01],
[ 8.6027e-01],
[-2.0818e-01],
[ 8.6861e-01],
```

```
[-2.3745e-01],
[-3.7715e-01],
[ 3.7565e-01],
[ 4.1063e-01],
[-1.0356e-01],
[ 1.5555e-02],
[ 1.2193e-01],
[ 8.9637e-01],
[-3.3029e-01],
[ 9.9377e-01],
[-6.0214e-01],
[-6.1211e-01],
[-9.0091e-02],
[ 3.6266e-01],
[ 2.4291e-02],
[-2.1113e-01],
[ 1.2876e-02],
[ 6.8733e-01],
[-7.3573e-01],
[ 8.0978e-01],
[-8.4776e-01],
[ 5.1269e-01],
[ 9.3435e-01],
[-7.6067e-01],
[-5.8556e-01],
[ 4.9405e-01],
[-9.3594e-01],
[ 5.8041e-01],
[ 6.1043e-01],
[ 1.1148e-01],
[-7.0203e-01],
[-4.3720e-01],
[-3.9807e-01],
[ 4.3780e-01],
[ 8.1264e-04],
[ 9.6366e-01],
[ 7.9211e-04],
[ 6.8935e-01],
[ 1.8652e-05],
[-6.1394e-01],
[-7.8667e-01],
[ 8.7644e-02],
[ 7.7324e-01],
[-2.0008e-01],
[-8.3842e-01],
[ 7.7794e-01],
[-6.2806e-01],
[-7.0751e-01],
[-4.4605e-01],
[-2.5529e-01],
[ 1.3896e-01],
[-4.2235e-02],
[-1.9177e-01],
[ 5.6033e-02],
[ 2.7336e-01],
[-3.6672e-01],
```

```
        [-4.0707e-01],
        [-1.0221e-02],
        [ 5.1991e-01],
        [ 8.3618e-01],
        [ 7.0531e-01],
        [-4.4052e-02],
        [-3.1491e-01],
        [ 8.1840e-03],
        [ 1.1361e+00],
        [ 1.4472e-03],
        [-5.2538e-01],
        [-1.0593e+00],
        [ 9.2209e-01],
        [ 9.7429e-01],
        [ 2.6266e-01],
        [-9.7014e-01],
        [ 9.1810e-01],
        [-5.4437e-01],
        [-1.1049e+00],
        [ 4.8534e-01],
        [ 3.7063e-01],
        [-1.3812e-01],
        [ 3.9486e-01],
        [-1.7504e-01],
        [ 2.6157e-02],
        [ 1.0403e-02],
        [-7.8663e-01],
        [-1.0116e+00],
        [ 7.6121e-01],
        [ 4.4077e-01],
        [ 5.3982e-01],
        [ 4.3924e-01],
        [-2.0973e-01],
        [-5.2809e-01],
        [ 8.0819e-01]])
hidden.bias tensor([-8.2030e-01, -8.9605e-01,  4.3682e-01, -6.6762e-01,  1.7351e-0
1,
         5.6907e-01,  5.9031e-01,  1.9012e-01,  3.4798e-02,  6.3320e-01,
         1.5250e-01, -3.9184e-01, -8.2752e-01, -4.3776e-01,  6.4386e-02,
         6.3188e-02, -2.9778e-01,  2.6068e-01, -3.2677e-01, -4.1748e-01,
        -5.7515e-01,  4.5646e-01,  9.0703e-01,  2.8824e-01,  1.0582e-01,
        -3.8726e-01, -6.2046e-01,  6.1921e-01, -4.7959e-01,  4.0686e-01,
        -4.2995e-01,  1.4739e-01, -5.0269e-01, -8.2384e-01,  2.2061e-01,
         3.9094e-01, -4.5403e-01,  9.8692e-01,  5.5067e-01,  6.5889e-01,
        -7.7836e-01, -5.7225e-01, -2.7632e-02, -4.8038e-01, -2.5726e-01,
         4.5545e-01,  1.8326e-01,  4.6829e-01, -1.3261e-02, -1.1741e+00,
         6.8669e-01,  7.3329e-02,  5.5141e-01, -4.3707e-03, -5.1506e-01,
        -1.6856e-01,  7.0710e-01,  9.1862e-01, -1.9240e-01,  6.2045e-02,
        -6.3694e-01, -5.7244e-01, -5.0375e-01, -2.6548e-01,  2.2908e-01,
        -3.0974e-01, -3.8215e-01, -9.9417e-01, -4.0235e-03, -4.2700e-01,
        -3.2040e-01,  2.4806e-01, -4.8137e-01, -3.5257e-01,  6.0470e-01,
         1.1944e-01, -5.5139e-01, -2.6564e-01, -5.8644e-01,  1.9562e-01,
        -4.6773e-01, -3.1452e-01, -9.6412e-01, -7.7139e-01, -6.0746e-01,
         4.0072e-02, -7.2063e-01, -1.5855e+00,  4.5721e-01, -1.8492e-01,
        -5.9331e-01, -5.4010e-01,  6.5027e-01, -5.8130e-01,  3.5020e-02,
        -9.8018e-02, -5.4724e-01,  5.2507e-01, -4.7219e-01,  7.9701e-02,
```

```
        -7.1571e-01, -1.0269e+00, -9.0961e-01,  2.9111e-01, -4.2189e-06,
         4.0087e-01, -1.0741e+00,  2.7547e-01, -7.1018e-01,  5.3069e-01,
         3.4727e-02, -9.4194e-01, -9.3098e-01, -1.3223e-01, -4.2245e-01,
        -3.6219e-02,  5.0499e-01,  2.7915e-01, -2.8390e-01, -1.2028e+00,
         1.0420e-01,  7.8267e-02, -1.6658e-01, -9.2561e-01,  1.0498e-01,
         7.5615e-01,  4.8207e-02,  1.7142e-01, -5.9247e-01, -3.2824e-01,
        -1.3931e+00,  6.0502e-01, -5.1848e-01,  1.4749e-01,  7.5856e-01,
        -2.8728e-01, -4.5017e-01,  6.9153e-01, -1.0877e+00, -3.4485e-01,
         1.6003e-01,  5.6608e-01,  5.3713e-02, -3.6494e-01,  2.9910e-03,
        -7.8356e-01,  3.1353e-03,  1.2952e-01, -9.1751e-05,  1.1829e-01,
         1.6524e-01, -9.5019e-01, -5.7271e-01,  3.1509e-01,  2.1044e-01,
        -4.8409e-01,  1.0220e-01, -3.4773e-01,  9.1991e-02, -3.5295e-03,
        -3.6102e-01, -8.6626e-02, -9.4403e-01,  3.9925e-01,  3.8081e-02,
        -9.0768e-01,  2.0225e-01,  3.3480e-01,  1.3740e-01,  2.9694e-01,
         2.1895e-01,  4.8425e-02, -6.2987e-01, -7.0378e-01,  7.0348e-01,
        -4.7171e-01,  1.1160e-01, -4.7264e-01, -3.2740e-01,  3.3290e-01,
         4.3548e-02,  2.7241e-01, -1.2632e+00, -3.1605e-01,  3.2711e-01,
         5.7349e-01,  4.3451e-01,  1.5165e-01,  8.3868e-01, -7.8035e-01,
         1.0797e-01,  1.2278e-02,  4.6312e-01, -5.4206e-01,  1.1818e-01,
        -9.1982e-01,  1.0887e-01, -9.4870e-01,  3.3074e-01,  7.6878e-01,
         2.5004e-01])
predict.weight tensor([[-3.2766e-01, -1.1766e-02,  1.3151e-01,  3.2753e-01,  1.164
6e-01,
         -6.8852e-03, -1.9624e-01,  1.3655e-01,  2.0425e-04,  5.9713e-02,
          1.1326e-03,  2.8487e-01, -6.1381e-02, -9.3999e-02, -4.4286e-02,
         -7.4519e-02, -1.2649e-01, -6.4512e-02,  3.2242e-01,  4.9339e-02,
         -2.0601e-02,  7.4262e-02, -1.5443e-01,  2.3585e-02,  8.9585e-02,
         -3.5099e-01, -2.2057e-01, -1.6602e-01,  5.8982e-03,  7.1582e-02,
          2.4707e-01, -6.4962e-02, -2.8901e-01,  3.6119e-02,  1.7931e-02,
          9.7429e-02,  2.4103e-01, -1.7790e-01, -2.1902e-03, -2.1608e-03,
         -3.3510e-02, -5.2966e-02, -4.3519e-03,  1.6995e-02, -3.4545e-01,
         -1.0765e-02, -9.9989e-02, -1.0011e-01, -3.7249e-04,  1.4691e+00,
          6.2912e-02,  7.5636e-02, -1.8203e-02, -3.4694e-05, -1.5133e-01,
          5.7953e-02,  4.8545e-02,  1.1030e-01, -2.1966e-02,  1.2101e-02,
          2.0885e-01,  4.8995e-04, -2.5285e-01, -8.6774e-03,  1.6156e-01,
         -2.6201e-01,  5.0716e-02, -2.1648e-02,  7.5976e-05,  5.0144e-02,
         -3.0721e-01,  9.3588e-03,  2.2135e-02,  2.4501e-01, -1.6055e-01,
         -1.7303e-01,  1.0592e-01, -1.7351e-02,  2.7989e-01,  1.6549e-01,
         -1.4505e-02, -6.8942e-02, -2.8199e-02, -3.4159e-02,  1.5674e-01,
          4.9527e-02, -3.7857e-01, -2.0091e+00, -1.6511e-01, -4.3431e-02,
          1.8875e-01,  1.4727e-01,  6.1401e-02,  2.8085e-01, -3.0314e-02,
         -9.8940e-04,  1.0362e-03, -1.3967e-01, -2.3157e-01,  1.5339e-01,
         -5.1018e-02,  1.2747e+00,  2.7447e-02, -8.5800e-02,  6.3348e-08,
          9.1292e-02,  1.2371e+00,  1.7142e-01,  5.6892e-02,  1.7983e-01,
         -9.7535e-02, -3.6000e-02, -2.8899e-02, -8.8156e-04,  3.4648e-02,
          5.3204e-03, -1.2650e-03,  1.2894e-01,  3.2040e-02,  1.0288e+00,
         -9.3169e-02, -4.7260e-02, -2.0237e-02,  6.1971e-02,  6.7866e-04,
         -2.3165e-02,  2.7206e-04,  1.3437e-01,  1.5250e-01, -2.0520e-01,
         -2.9144e+00, -1.2166e-01, -3.6379e-01, -8.6383e-02,  1.1694e-01,
         -6.7763e-02,  2.5856e-01, -1.2188e-01,  1.2351e+00, -1.1463e-02,
         -1.1956e-01,  7.6797e-02, -5.6778e-02, -1.7778e-01,  2.7893e-05,
         -2.1287e-01,  5.7611e-05,  1.1503e-01, -2.5430e-06, -1.0140e-01,
         -8.6593e-02, -5.6613e-02, -2.7060e-01,  2.0958e-02, -1.5935e-01,
         -3.1985e-01, -9.1810e-02,  9.0315e-02, -1.3853e-01, -6.4920e-03,
          1.4635e-02, -4.4417e-03, -4.0407e-02, -1.7282e-03,  5.1148e-02,
          5.4351e-03, -2.3672e-02,  2.8737e-02,  1.8557e-01,  1.8615e-01,
```

```
         1.2621e-01,  4.8470e-03,  3.5655e-02,  1.0349e-02,  1.2326e-01,
        -5.1941e-03, -1.3111e-01,  3.2170e-01, -2.5745e-01,  1.7181e-01,
         1.4770e-01, -1.0386e-01,  1.2461e+00,  5.6442e-02, -1.3732e-01,
        -1.4433e-01, -8.3810e-02,  4.5593e-02,  5.7387e-02,  5.9003e-02,
         1.2781e-03, -1.3120e-03, -6.5275e-02,  2.8374e-01,  7.0375e-02,
        -2.9655e-03,  1.6092e-01, -1.9024e-02,  2.7824e-02,  9.1401e-02,
         1.6738e-01]])
predict.bias tensor([0.0961])
```



```
Trained Network:
Number of nodes in hidden layer: 201 (num_points+1)
hidden.weight tensor([[ 2.2640],
        [ 2.2682],
        [-2.2102],
        [ 2.1456],
        [ 2.2325],
        [ 2.1960],
        [ 1.9217],
        [ 1.8311],
        [-0.2790],
        [-2.1236],
        [ 1.7509],
        [-2.5615],
        [-2.5464],
        [-2.3858],
        [-2.1987],
        [-1.9360],
        [-0.2531],
        [-0.2341],
        [ 2.0732],
        [-2.5390],
        [-2.1655],
        [-2.4247],
        [ 0.2884],
        [-2.0343],
        [ 2.1491],
        [ 0.0513],
        [ 2.0059],
        [-2.3880],
        [-2.2995],
        [-2.2110],
        [-2.0001],
        [ 0.1494],
        [-1.7846],
        [ 2.3365],
        [-2.5566],
        [ 1.9818],
        [ 2.3006],
        [ 2.4181],
        [-2.1219],
        [ 2.4394],
        [ 2.5569],
        [ 0.6407],
        [ 1.0179],
        [-2.4287],
        [-1.8511],
        [ 2.4057],
        [ 2.0283],
        [-0.1727],
        [ 1.9221],
        [ 2.2159],
        [ 2.2008],
        [ 1.9727],
        [ 2.3105],
        [-1.8555],
```

```
[ 2.4032],
[ 2.2665],
[ 2.1856],
[-2.1920],
[ 2.3237],
[-2.5432],
[ 2.3878],
[-2.2717],
[ 2.0675],
[-2.0982],
[-1.9113],
[ 0.2376],
[-0.3447],
[ 1.9606],
[-1.9459],
[-2.2281],
[-2.4145],
[-0.2883],
[-0.1387],
[-0.2884],
[ 2.0532],
[ 0.0885],
[ 0.3771],
[-2.0020],
[-2.0602],
[ 0.0793],
[-2.1068],
[ 2.4802],
[ 0.1176],
[ 2.1540],
[ 2.0318],
[ 2.2921],
[-2.0526],
[ 0.3453],
[ 0.3201],
[ 2.1079],
[ 1.9512],
[-0.0066],
[ 0.2160],
[ 2.5418],
[ 0.2912],
[-0.3443],
[ 2.1733],
[-2.0516],
[ 2.3936],
[ 0.2081],
[ 0.1150],
[-2.3808],
[ 2.5973],
[-0.1984],
[ 2.0452],
[ 0.3082],
[ 2.4439],
[-2.1169],
[-0.0302],
[-0.0125],
```

```
[-2.0368],
[-0.1207],
[ 0.1743],
[ 0.0566],
[ 2.5165],
[-2.2386],
[-2.1970],
[ 2.0702],
[ 2.1704],
[-2.1316],
[-2.1495],
[-2.5376],
[ 2.3376],
[-2.0670],
[-0.0080],
[ 0.4354],
[-2.2845],
[ 2.1486],
[-2.0764],
[-2.3129],
[-2.3603],
[ 2.0152],
[ 2.2753],
[ 1.9320],
[-0.0070],
[-2.1379],
[ 1.9975],
[ 1.8831],
[-2.5679],
[-2.2107],
[-0.1034],
[ 1.9940],
[-2.4233],
[ 2.1507],
[ 0.2560],
[-2.4288],
[-2.2566],
[-1.9456],
[ 0.0523],
[-2.2091],
[ 0.0539],
[ 1.8903],
[-2.5515],
[-2.4661],
[-2.1495],
[-2.2339],
[-0.1094],
[-2.2681],
[-2.3914],
[-2.2423],
[ 2.1219],
[-2.1746],
[ 2.2597],
[ 0.7660],
[ 1.8812],
[ 0.3812],
```

```
        [-2.5992],
        [-2.1340],
        [ 2.0847],
        [ 2.2543],
        [-2.2719],
        [-2.4967],
        [ 2.4937],
        [-2.1943],
        [-2.0031],
        [-2.1413],
        [ 2.0644],
        [-2.1701],
        [ 1.9584],
        [ 2.0553],
        [ 1.8908],
        [-1.7654],
        [ 2.3636],
        [-1.9666],
        [ 2.2118],
        [-0.0523],
        [ 2.5036],
        [-2.5745],
        [ 1.9147],
        [-2.4951],
        [-2.3060],
        [ 1.9879],
        [-1.9672],
        [ 2.3076],
        [-1.8208],
        [ 2.3467],
        [ 2.0130],
        [-1.9162],
        [-0.3098],
        [-2.1460],
        [ 2.1318]])
hidden.bias tensor([-1.6514, -3.6998, -3.8108, -3.5729, -3.6446, -1.9178, -3.3719,
-3.2201,
        -0.8558, -3.6857, -3.0838, -3.5035, -3.9685, -3.9471, -3.7603, -3.3816,
        -0.6798, -0.6969, -3.5033, -3.6471, -3.7504, -3.8444,  1.2132, -3.5485,
        -2.7237, -0.7502, -1.8458,  0.0667, -3.6490, -3.3022, -3.4926,  1.2090,
        -3.1260, -3.6501, -3.1352, -3.4754, -3.1860, -3.6708, -0.6259, -2.6362,
        -3.8018,  0.9731,  1.5617, -3.6609, -3.2391, -3.0782, -3.5332,  1.2723,
        -3.3706, -3.1488, -2.6775, -3.4576, -2.4270, -3.2426, -3.7303,  1.1301,
         1.1269, -1.5801, -2.8712, -3.6473, -3.8139, -3.7937, -3.5429, -3.6593,
        -3.3401,  1.1657, -0.9617, -3.4402, -2.9341, -2.1221, -2.3700, -0.7393,
        -0.5752, -0.7601, -3.5693, -0.2922, -0.8273, -3.4923, -3.5927,  0.9161,
        -3.6738, -1.5657,  0.8489, -3.6807, -2.3589, -1.9632, -1.8589,  0.9888,
        -0.7610, -2.3423, -3.4169, -0.1209, -0.8298, -3.7278, -0.7563,  1.2303,
        -3.0489, -3.0134, -1.3332, -0.4213,  0.8875, -3.8557, -3.7567, -0.8100,
        -3.5184,  0.7736, -2.7572, -0.5917,  1.0047,  0.6781, -2.0691, -0.3484,
         1.1643, -0.4763, -3.1331, -2.2457, -1.9643, -1.0612, -1.9991, -2.4262,
        -3.7233, -0.1821, -3.0773, -3.6034, -0.6824, -1.0207, -3.2561, -2.8704,
        -3.6182, -3.1230, -2.9158, -2.7467, -2.4080, -3.3889, -0.4021, -2.7037,
        -1.3237, -3.3120, -3.9920, -1.6658, -0.5371, -3.4947, -2.8544, -3.5969,
        -0.6178, -3.9188, -3.7439, -3.2893, -0.6536, -2.4396, -0.1540, -1.8419,
        -3.9219, -2.1833, -3.5202, -3.7301,  0.7708, -3.8349, -2.5762, -3.7243,
```

```
        -3.5859, -3.7729, -3.6381,  1.1652, -3.3019,  1.1850, -3.9138, -1.3985,
        -3.2311, -3.2469, -2.9105, -3.0595, -3.7491, -2.9375,  0.6107, -1.1682,
        -3.5360, -3.7562, -2.6311, -3.2931, -3.0296,  1.4930, -3.1933, -3.4352,
        -3.6782,  0.6088, -2.9021, -2.8313, -3.3388, -3.5537, -3.1416, -3.4840,
        -3.4351, -1.8050, -2.3842, -3.6397, -3.5287, -3.3525,  1.2963, -2.8954,
        -3.6434])
predict.weight tensor([[ 1.1167e+00,  4.9903e+00,  4.0940e+00,  4.8269e+00,  3.316
1e+00,
         1.3604e+00,  5.3746e+00,  2.8984e+00,  1.6688e-02,  5.4232e+00,
         2.8583e+00,  3.0381e+00,  3.8787e+00,  4.6006e+00,  3.2328e+00,
         2.9181e+00,  1.3272e-02,  6.9770e-02,  3.6499e+00,  3.2086e+00,
         4.9035e+00,  3.4458e+00, -3.2921e-01,  3.3388e+00,  2.1230e+00,
         5.6052e-02,  1.2493e+00,  8.9172e-01,  3.6095e+00,  2.5733e+00,
         3.3322e+00, -3.8959e-01,  3.6718e+00,  3.5499e+00,  2.4505e+00,
         4.9799e+00,  2.8612e+00,  3.9798e+00,  8.7240e-01,  2.0793e+00,
         3.7534e+00,  9.1300e-01,  6.8830e-01,  3.5918e+00,  4.6312e+00,
         2.4582e+00,  5.7432e+00, -4.2361e-01,  3.0679e+00,  2.7424e+00,
         2.0654e+00,  5.9076e+00,  1.8225e+00,  3.4031e+00,  4.1434e+00,
         9.9073e-01,  9.8859e-01,  1.1398e+00,  2.0946e+00,  3.2871e+00,
         4.8040e+00,  3.7096e+00,  4.2325e+00,  3.0764e+00,  4.0097e+00,
        -3.7753e-01, -6.7883e-02,  4.4705e+00,  2.4708e+00,  1.4431e+00,
         1.6190e+00,  7.2448e-02, -7.5715e-03,  3.4543e-02,  5.3976e+00,
         6.8102e-02,  4.2929e-02,  3.6681e+00,  3.7197e+00, -3.3476e-01,
         4.0334e+00,  1.2346e+00, -2.9294e-01,  3.9174e+00,  1.8053e+00,
         1.3134e+00,  1.3450e+00, -2.6648e-01,  5.0097e-02,  1.9448e+00,
         4.5382e+00,  3.2134e-03, -4.1632e-02,  3.6639e+00, -6.6884e-02,
        -3.5659e-01,  2.7312e+00,  2.4763e+00,  1.1643e+00, -4.8907e-02,
        -3.6374e-01,  4.5305e+00,  4.0659e+00,  4.8340e-02,  4.9035e+00,
        -1.3651e-01,  2.1207e+00,  9.2957e-01, -4.1520e-01, -3.1833e-01,
         1.6171e+00,  4.8791e-02, -3.7810e-01,  3.5699e-02,  2.3750e+00,
         1.5955e+00,  1.3418e+00,  1.0988e+00,  1.3809e+00,  1.8859e+00,
         4.9641e+00,  8.9451e-01,  2.5335e+00,  5.0913e+00,  6.5552e-02,
         2.5265e-02,  2.4523e+00,  2.4524e+00,  4.2140e+00,  2.4087e+00,
         2.1557e+00,  2.4243e+00,  1.8864e+00,  5.2478e+00,  2.7481e-03,
         2.2268e+00,  1.0340e+00,  3.1513e+00,  4.2152e+00,  1.1437e+00,
        -3.3993e-02,  4.7900e+00,  2.2481e+00,  3.4239e+00, -1.8984e-02,
         3.7141e+00,  3.9771e+00,  2.7272e+00,  4.5147e-02,  1.8989e+00,
        -3.0510e-02,  1.4626e+00,  4.0722e+00,  1.5207e+00,  3.4101e+00,
         3.9358e+00, -2.9697e-01,  4.9848e+00,  1.6994e+00,  3.3835e+00,
         3.2855e+00,  3.4451e+00,  4.8250e+00,  9.2693e-01,  4.5754e+00,
        -3.3235e-01,  3.8420e+00,  9.9374e-01,  2.5505e+00,  2.6322e+00,
         2.1605e+00,  2.3416e+00,  3.9977e+00,  2.3883e+00,  8.7738e-01,
         9.7596e-01,  4.6365e+00,  4.2845e+00,  2.0506e+00,  3.2478e+00,
         2.8341e+00,  8.9511e-01,  2.6753e+00,  4.1101e+00,  4.7667e+00,
        -2.3793e-01,  2.2149e+00,  2.0714e+00,  3.2315e+00,  3.0150e+00,
         2.6280e+00,  5.3950e+00,  3.2232e+00,  1.2384e+00,  2.1247e+00,
         3.4059e+00,  3.7128e+00,  3.0256e+00, -3.6668e-01,  2.3544e+00,
         4.9137e+00]])
predict.bias tensor([-0.4721])
```

Out[ ]:
```
(Net(
  (hidden): Linear(in_features=1, out_features=201, bias=True)
  (predict): Linear(in_features=201, out_features=1, bias=True)
 ),
 [38540.92724609375,
  39167.48327636719,
  34558.375,
  27355.077270507812,
  24621.923583984375,
  21481.571655273438,
  20271.326904296875,
  18084.59912109375,
  17451.994750976562,
  16675.09912109375,
  15489.106323242188,
  14563.004211425781,
  14100.592895507812,
  15008.484619140625,
  12936.868225097656,
  13147.6279296875,
  14530.686157226562,
  12065.708312988281,
  11630.297302246094,
  14595.795593261719,
  11040.173034667969,
  10872.522338867188,
  11053.007080078125,
  11404.056396484375,
  10376.840759277344,
  10724.175109863281,
  10974.770629882812,
  9006.164123535156,
  10476.71923828125,
  9159.3544921875,
  10181.3466796875,
  8612.829223632812,
  7953.230407714844,
  7932.117980957031,
  8529.108367919922,
  8058.507141113281,
  8601.933532714844,
  7460.373840332031,
  6957.68115234375,
  7639.367248535156,
  6594.123382568359,
  6249.291534423828,
  6893.129241943359,
  6206.659820556641,
  6949.921478271484,
  6110.1688232421875,
  6062.3050537109375,
  5759.814880371094,
  5729.371795654297,
  5691.010314941406,
  5339.003082275391,
  5191.349609375,
```

```
5140.797760009766,
4951.077667236328,
5485.395919799805,
4863.430023193359,
5913.135589599609,
4644.264892578125,
4581.950012207031,
4357.714340209961,
5332.4456787109375,
4446.922607421875,
4783.126968383789,
4202.712188720703,
3923.3968505859375,
4535.959701538086,
3842.4020080566406,
4107.484771728516,
3677.427490234375,
4049.5802001953125,
3794.9754638671875,
3466.2926025390625,
3842.2532958984375,
3652.8153228759766,
3358.6451416015625,
3784.6139068603516,
3322.8192138671875,
3122.817314147949,
3628.4579162597656,
2890.5261573791504,
3264.029556274414,
3428.21728515625,
3434.5731658935547,
2671.040277481079,
3139.869674682617,
2915.7626190185547,
2872.467788696289,
2558.794761657715,
2942.0558319091797,
2549.4256286621094,
2544.872848510742,
2342.0872802734375,
2388.8485717773438,
3776.4246978759766,
2506.6177139282227,
2541.3079376220703,
2893.303871154785,
2143.7527770996094,
2221.582290649414,
2300.496322631836,
2063.747528076172,
2166.4808654785156,
2247.815872192383,
2050.782730102539,
2003.0333709716797,
1869.43994140625,
1949.7063598632812,
1992.103614807129,
```

```
1806.2825393676758,
1826.053825378418,
1716.9828186035156,
1868.5805587768555,
1659.9712295532227,
1577.0327882766724,
1604.7871704101562,
1513.8278369903564,
1584.2446594238281,
1645.0620346069336,
1485.1327095031738,
1591.0189476013184,
1475.8282089233398,
1511.137092590332,
1483.3570175170898,
1457.9201431274414,
1335.6229915618896,
1769.3632888793945,
1392.4942474365234,
1338.280776977539,
1273.5768737792969,
1362.8616485595703,
1328.7848281860352,
1514.916160583496,
1236.7807807922363,
1482.0370292663574,
1257.9637069702148,
1431.7717247009277,
1203.8762741088867,
1341.9299812316895,
1232.6122207641602,
1249.3418312072754,
1156.7761764526367,
1060.112726688385,
1055.243808746338,
1149.2369995117188,
988.1659250259399,
1122.0796585083008,
1034.9523391723633,
1025.7322959899902,
1006.4914093017578,
969.2111320495605,
993.0946083068848,
913.7007684707642,
950.9476470947266,
929.6961479187012,
1209.2821960449219,
943.562837600708,
904.4072341918945,
875.7073822021484,
894.8316383361816,
866.0642366409302,
838.6824893951416,
834.7023544311523,
789.2168233394623,
845.5367126464844,
```

```
1103.7453804016113,
1010.3287906646729,
803.4740781784058,
794.1596240997314,
910.6847305297852,
774.4103469848633,
1079.612964630127,
746.1017227172852,
751.907600402832,
902.0896596908569,
797.1844863891602,
679.3686027526855,
684.8481941223145,
861.0653495788574,
814.054759979248,
695.0970401763916,
718.7634773254395,
829.1975517272949,
834.9750022888184,
628.3717041015625,
660.9902076721191,
619.4134254455566,
589.2678785324097,
745.0953598022461,
708.5590744018555,
622.0419807434082,
741.4557647705078,
562.0073931217194,
586.7891445159912,
672.5859079360962,
545.6573470830917,
624.5583782196045,
606.0834484100342,
752.9152221679688,
910.6231651306152,
538.3299907445908,
598.1644859313965,
505.28789710998535,
541.9736728668213,
548.699047088623,
494.2253189086914,
526.1923084259033,
619.4955863952637,
478.43579483032227,
476.3629467487335,
469.4148063659668,
464.55104637145996,
610.8855056762695,
617.22727394104,
495.885290145874,
504.6518249511719,
460.9801367223263,
487.6410026550293,
444.8062381744385,
416.3844526410103,
440.19746494293213,
```

```
468.31103324890137,
420.55295181274414,
432.3953342437744,
414.7808609008789,
408.374174118042,
512.1482067108154,
378.28537607192993,
531.3694496154785,
415.8124828338623,
410.10620307922363,
392.35535430908203,
372.1480917930603,
353.5122749209404,
436.05993461608887,
393.2579402923584,
476.0482654571533,
387.52681159973145,
402.2053031921387,
398.730411529541,
358.46756744384766,
328.6388385295868,
338.96981477737427,
342.58030128479004,
614.5719213485718,
330.98983001708984,
339.00731468200684,
409.83363246917725,
351.0737714767456,
423.4921073913574,
350.7643699645996,
311.81183981895447,
392.6485137939453,
289.74071884155273,
285.48689134418964,
375.88658380508423,
335.2752380371094,
309.7829866409302,
287.43779611587524,
325.3578071594238,
281.4480228424072,
281.41791120171547,
274.77225971221924,
276.389705657959,
300.5395517349243,
312.3586859703064,
306.9013481140137,
263.8912196159363,
257.94907304644585,
323.9818067550659,
344.66621685028076,
292.5588130950928,
258.99449729919434,
330.68597507476807,
252.07736897468567,
439.58826065063477,
273.54867935180664,
```

```
237.8410758972168,
227.94594717025757,
383.1591281890869,
265.6676387786865,
252.12993335723877,
340.1128177642822,
340.3573431968689,
241.4422688484192,
257.4676990509033,
223.30008935928345,
226.5422763824463,
206.94789040088654,
227.1832110285759,
212.60376662015915,
236.5202121734619,
335.85999298095703,
241.45107173919678,
199.93214976787567,
330.7647862434387,
267.6566905975342,
226.05096054077148,
244.0597915649414,
208.6587381362915,
189.87893137335777,
218.8087863922119,
198.9705729484558,
183.64833730459213,
200.47181034088135,
182.13279795646667,
229.86313247680664,
189.86981391906738,
176.99156984686852,
199.41229629516602,
184.89284205436707,
183.19159281253815,
172.6621595621109,
192.8820071220398,
188.28729438781738,
189.77325057983398,
167.20276421308517,
169.36763954162598,
212.79811477661133,
170.31087017059326,
184.54585647583008,
278.4924249649048,
196.93368983268738,
239.4286003112793,
169.16643404960632,
235.3758897781372,
198.50975620746613,
162.37632036209106,
171.76558303833008,
197.7760729789734,
185.46754837036133,
172.0076379776001,
173.76504611968994,
```

```
167.88720512390137,
144.54321330040693,
153.7423391342163,
170.093770891428,
159.0871558189392,
185.7781867980957,
152.67791557312012,
141.20256567001343,
148.07288718223572,
189.17652940750122,
158.3185851573944,
169.64210510253906,
147.69604873657227,
163.95388901233673,
134.06454876065254,
136.9115288257599,
131.86890810728073,
139.06861782073975,
124.58755898475647,
162.3802855014801,
157.45934295654297,
129.11797714233398,
134.85046529769897,
132.618670463562,
131.74246263504028,
130.90482187271118,
131.3353943824768,
127.15103697776794,
125.61831474304199,
122.10186402499676,
151.7353024482727,
130.58378720283508,
164.69711017608643,
152.07908630371094,
110.3743877671659,
123.65948051214218,
114.97288465499878,
108.9985283613205,
120.61898565292358,
117.29709351062775,
119.43744230270386,
166.44861698150635,
143.66879177093506,
123.33422636985779,
180.63961029052734,
122.33839637041092,
117.14643573760986,
105.23098051548004,
132.514233648777,
108.60192966461182,
128.09054577350616,
112.84247040748596,
105.89956557750702,
116.26983642578125,
104.34714794158936,
94.84463577717543,
```

```
97.78458762168884,
95.64810395240784,
131.74170470237732,
127.56163358688354,
94.79473781585693,
118.59981060028076,
92.99281167984009,
123.94401788711548,
103.0848009288311,
110.53248488903046,
138.92052936553955,
90.86117720603943,
184.72049951553345,
126.37107610702515,
100.86519765853882,
182.65836763381958,
156.50478386878967,
122.21875667572021,
101.38517594337463,
83.41922549530864,
92.19926956668496,
83.91518950462341,
85.11148178577423,
87.63093638420105,
79.97986635565758,
84.00372242927551,
79.0619592666626,
94.71891450881958,
91.06285965442657,
84.6315024793148,
90.81900024414062,
85.78595066070557,
79.40324658155441,
87.13124895095825,
73.88656663894653,
80.59240734577179,
80.27628421783447,
87.71291017532349,
88.14764940738678,
92.5671751499176,
74.57502317428589,
81.74683332443237,
75.89947271347046,
71.24308143556118,
78.42586660385132,
76.59887933731079,
85.3760378062725,
90.73390626907349,
68.5361533164978,
90.63635289669037,
86.34498333930969,
96.8373920917511,
98.20446538925171,
83.82624404132366,
85.69241392612457,
77.04287958145142,
```

```
91.91380834579468,
69.40205430984497,
70.60284614562988,
66.92071497440338,
65.61800661683083,
106.03177785873413,
64.28398889303207,
69.76310467720032,
64.69142532348633,
69.4935298487544,
63.17113983631134,
108.98427474498749,
73.78862619400024,
62.619184255599976,
63.098647594451904,
100.18910276889801,
84.80745935440063,
69.64771318435669,
59.014785408973694,
96.0047402381897,
100.45902466773987,
71.41562885046005,
64.87493073940277,
70.47738313674927,
62.01844525337219,
62.405444860458374,
56.44026482105255,
55.16595819592476,
54.35893052816391,
56.58323871344328,
56.64790388941765,
53.54219138622284,
52.84939044713974,
60.75202965736389,
47.922773629426956,
51.02022209763527,
48.77679068595171,
63.398184180259705,
61.64295622706413,
52.532081961631775,
60.911451518535614,
52.41591429710388,
57.99768304824829,
47.113478258252144,
50.75513172149658,
57.521285355091095,
48.16569846868515,
88.63700127601624,
69.14533615112305,
68.24127918481827,
52.07340621948242,
64.90801954269409,
52.963394463062286,
46.94602574408054,
49.93008017539978,
114.5305404663086])
```

Q3

In [ ]:
```
import tensorflow as tf
from tensorflow.keras.datasets import fashion_mnist
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense, Activatio
from tensorflow.keras import regularizers

# Load Fashion-MNIST data
(x_train, y_train), (x_test, y_test) = fashion_mnist.load_data()

# Normalize pixel values to be between 0 and 1
x_train = x_train.astype('float32') / 255.0
x_test = x_test.astype('float32') / 255.0

# Reshape data to be suitable for convolutional layers
x_train = x_train.reshape(-1, 28, 28, 1)
x_test = x_test.reshape(-1, 28, 28, 1)

# Define model architecture with L2 regularization
'''
model = Sequential()
model.add(Conv2D(32, (3, 3), activation='relu', kernel_regularizer=regularizers.l2(
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Conv2D(64, (3, 3), activation='relu', kernel_regularizer=regularizers.l2(
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Flatten())
model.add(Dense(128, activation='relu', kernel_regularizer=regularizers.l2(0.001)))
model.add(Dropout(0.5))
model.add(Dense(10, activation='softmax'))
'''

model = Sequential()
model.add(Conv2D(4, (2, 2), activation='relu', kernel_regularizer=regularizers.l2(0
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Conv2D(2, (2, 2), activation='relu', kernel_regularizer=regularizers.l2(0
#model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Flatten())
model.add(Dropout(0.2))
#model.add(Conv2D(10, (2, 2), activation='softmax'))
model.add(Dense(10, activation='softmax'))

# Compile model
model.compile(loss='sparse_categorical_crossentropy', optimizer='adam', metrics=['a

# Train model
history = model.fit(x_train, y_train, epochs=100, validation_data=(x_test, y_test))

# Evaluate model on test set
test_loss, test_acc = model.evaluate(x_test, y_test)
train_loss, train_acc = model.evaluate(x_train, y_train)

print('    loss     (train, test): {:.4f},  {:.4f}'.format(train_loss, test_loss))
print('    accuracy (train, test): {:.4f},  {:.4f}'.format(train_acc, test_acc))
```

```python
import matplotlib.pyplot as plt

plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('Model loss')
plt.ylabel('Loss')
plt.xlabel('Epoch')
plt.legend(['Train', 'Test'], loc='upper left')
plt.show()

plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])
plt.title('Model accuracy')
plt.ylabel('accuracy')
plt.xlabel('Epoch')
plt.legend(['Train', 'Test'], loc='upper left')
plt.show()

#print('Training accuracy:', history.history['accuracy'][-1])
print('Train accuracy:', train_acc)
print('Test accuracy:', test_acc)
```

```
Epoch 1/100
1875/1875 [==============================] - 5s 3ms/step - loss: 0.6882 - accurac
y: 0.7565 - val_loss: 0.5080 - val_accuracy: 0.8191
Epoch 2/100
1875/1875 [==============================] - 5s 3ms/step - loss: 0.5196 - accurac
y: 0.8166 - val_loss: 0.4747 - val_accuracy: 0.8344
Epoch 3/100
1875/1875 [==============================] - 5s 3ms/step - loss: 0.4992 - accurac
y: 0.8254 - val_loss: 0.4588 - val_accuracy: 0.8374
Epoch 4/100
1875/1875 [==============================] - 5s 3ms/step - loss: 0.4849 - accurac
y: 0.8271 - val_loss: 0.4595 - val_accuracy: 0.8420
Epoch 5/100
1875/1875 [==============================] - 5s 3ms/step - loss: 0.4755 - accurac
y: 0.8324 - val_loss: 0.4428 - val_accuracy: 0.8481
Epoch 6/100
1875/1875 [==============================] - 5s 3ms/step - loss: 0.4684 - accurac
y: 0.8336 - val_loss: 0.4457 - val_accuracy: 0.8460
Epoch 7/100
1875/1875 [==============================] - 5s 3ms/step - loss: 0.4672 - accurac
y: 0.8343 - val_loss: 0.4420 - val_accuracy: 0.8460
Epoch 8/100
1875/1875 [==============================] - 5s 3ms/step - loss: 0.4657 - accurac
y: 0.8347 - val_loss: 0.4412 - val_accuracy: 0.8464
Epoch 9/100
1875/1875 [==============================] - 5s 3ms/step - loss: 0.4605 - accurac
y: 0.8387 - val_loss: 0.4396 - val_accuracy: 0.8480
Epoch 10/100
1875/1875 [==============================] - 5s 3ms/step - loss: 0.4610 - accurac
y: 0.8377 - val_loss: 0.4325 - val_accuracy: 0.8500
Epoch 11/100
1875/1875 [==============================] - 5s 3ms/step - loss: 0.4555 - accurac
y: 0.8395 - val_loss: 0.4436 - val_accuracy: 0.8427
Epoch 12/100
1875/1875 [==============================] - 5s 3ms/step - loss: 0.4545 - accurac
y: 0.8379 - val_loss: 0.4294 - val_accuracy: 0.8491
Epoch 13/100
1875/1875 [==============================] - 5s 3ms/step - loss: 0.4540 - accurac
y: 0.8390 - val_loss: 0.4239 - val_accuracy: 0.8548
Epoch 14/100
1875/1875 [==============================] - 5s 3ms/step - loss: 0.4484 - accurac
y: 0.8414 - val_loss: 0.4306 - val_accuracy: 0.8454
Epoch 15/100
1875/1875 [==============================] - 5s 3ms/step - loss: 0.4484 - accurac
y: 0.8421 - val_loss: 0.4283 - val_accuracy: 0.8494
Epoch 16/100
1875/1875 [==============================] - 5s 3ms/step - loss: 0.4472 - accurac
y: 0.8411 - val_loss: 0.4259 - val_accuracy: 0.8494
Epoch 17/100
1875/1875 [==============================] - 5s 3ms/step - loss: 0.4454 - accurac
y: 0.8432 - val_loss: 0.4290 - val_accuracy: 0.8458
Epoch 18/100
1875/1875 [==============================] - 5s 3ms/step - loss: 0.4424 - accurac
y: 0.8435 - val_loss: 0.4196 - val_accuracy: 0.8517
Epoch 19/100
1875/1875 [==============================] - 5s 3ms/step - loss: 0.4425 - accurac
```

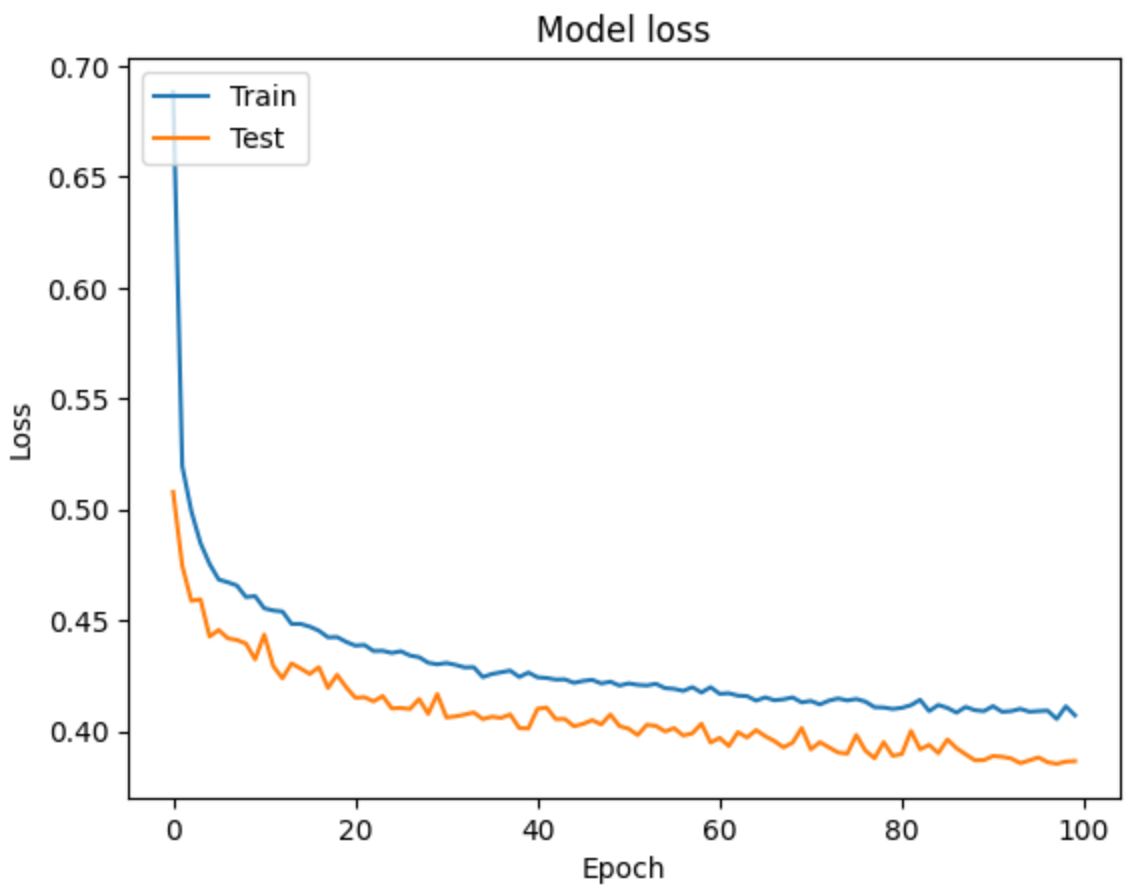
```
y: 0.8439 - val_loss: 0.4255 - val_accuracy: 0.8477
Epoch 20/100
1875/1875 [==============================] - 5s 3ms/step - loss: 0.4403 - accurac
y: 0.8451 - val_loss: 0.4199 - val_accuracy: 0.8539
Epoch 21/100
1875/1875 [==============================] - 5s 3ms/step - loss: 0.4387 - accurac
y: 0.8445 - val_loss: 0.4152 - val_accuracy: 0.8565
Epoch 22/100
1875/1875 [==============================] - 5s 3ms/step - loss: 0.4390 - accurac
y: 0.8456 - val_loss: 0.4154 - val_accuracy: 0.8551
Epoch 23/100
1875/1875 [==============================] - 5s 3ms/step - loss: 0.4362 - accurac
y: 0.8451 - val_loss: 0.4135 - val_accuracy: 0.8571
Epoch 24/100
1875/1875 [==============================] - 5s 3ms/step - loss: 0.4363 - accurac
y: 0.8458 - val_loss: 0.4160 - val_accuracy: 0.8554
Epoch 25/100
1875/1875 [==============================] - 5s 3ms/step - loss: 0.4354 - accurac
y: 0.8471 - val_loss: 0.4104 - val_accuracy: 0.8586
Epoch 26/100
1875/1875 [==============================] - 5s 3ms/step - loss: 0.4361 - accurac
y: 0.8463 - val_loss: 0.4106 - val_accuracy: 0.8561
Epoch 27/100
1875/1875 [==============================] - 5s 3ms/step - loss: 0.4342 - accurac
y: 0.8460 - val_loss: 0.4101 - val_accuracy: 0.8548
Epoch 28/100
1875/1875 [==============================] - 5s 3ms/step - loss: 0.4335 - accurac
y: 0.8467 - val_loss: 0.4146 - val_accuracy: 0.8571
Epoch 29/100
1875/1875 [==============================] - 5s 3ms/step - loss: 0.4310 - accurac
y: 0.8486 - val_loss: 0.4079 - val_accuracy: 0.8612
Epoch 30/100
1875/1875 [==============================] - 5s 3ms/step - loss: 0.4302 - accurac
y: 0.8484 - val_loss: 0.4168 - val_accuracy: 0.8565
Epoch 31/100
1875/1875 [==============================] - 5s 3ms/step - loss: 0.4308 - accurac
y: 0.8468 - val_loss: 0.4063 - val_accuracy: 0.8589
Epoch 32/100
1875/1875 [==============================] - 5s 3ms/step - loss: 0.4300 - accurac
y: 0.8491 - val_loss: 0.4068 - val_accuracy: 0.8602
Epoch 33/100
1875/1875 [==============================] - 5s 3ms/step - loss: 0.4288 - accurac
y: 0.8485 - val_loss: 0.4076 - val_accuracy: 0.8579
Epoch 34/100
1875/1875 [==============================] - 5s 3ms/step - loss: 0.4290 - accurac
y: 0.8490 - val_loss: 0.4085 - val_accuracy: 0.8597
Epoch 35/100
1875/1875 [==============================] - 5s 3ms/step - loss: 0.4245 - accurac
y: 0.8505 - val_loss: 0.4055 - val_accuracy: 0.8559
Epoch 36/100
1875/1875 [==============================] - 5s 3ms/step - loss: 0.4259 - accurac
y: 0.8505 - val_loss: 0.4066 - val_accuracy: 0.8577
Epoch 37/100
1875/1875 [==============================] - 5s 3ms/step - loss: 0.4267 - accurac
y: 0.8497 - val_loss: 0.4061 - val_accuracy: 0.8571
Epoch 38/100
```

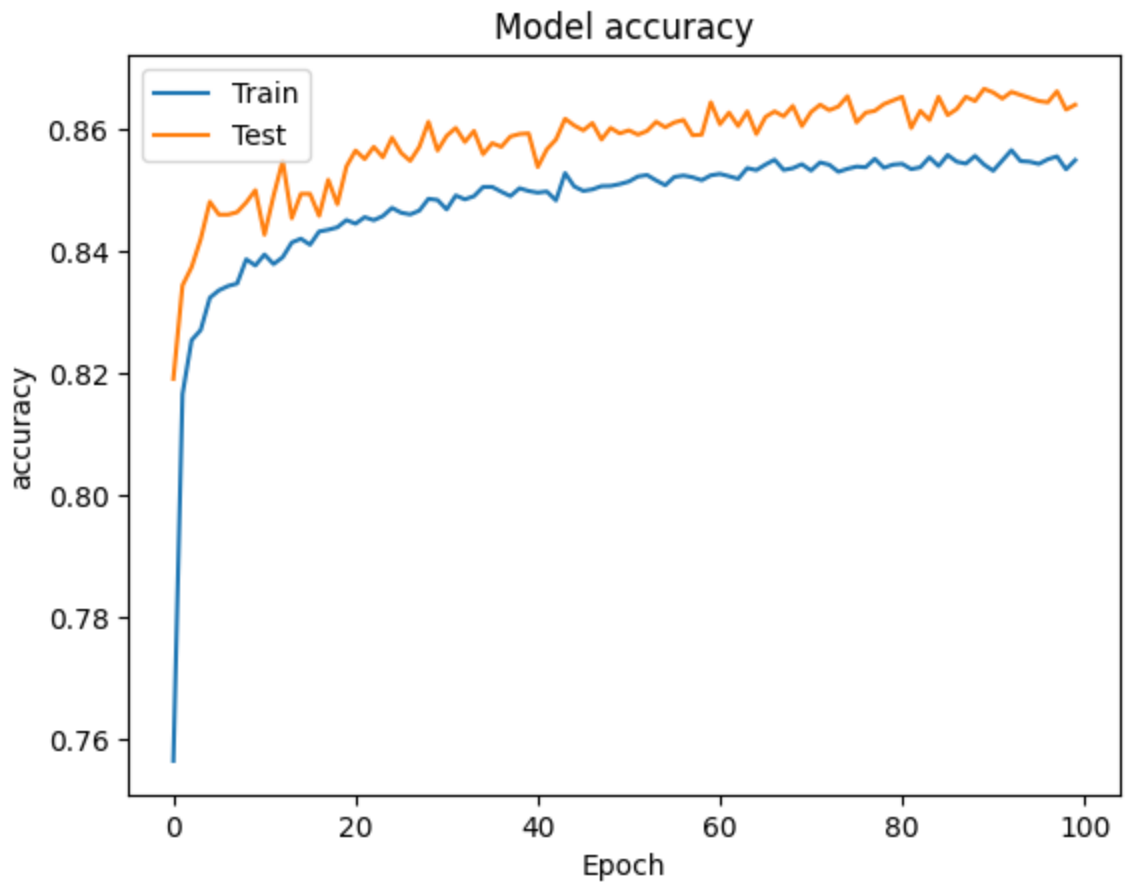
```
1875/1875 [==============================] - 5s 3ms/step - loss: 0.4274 - accurac
y: 0.8490 - val_loss: 0.4076 - val_accuracy: 0.8588
Epoch 39/100
1875/1875 [==============================] - 5s 3ms/step - loss: 0.4245 - accurac
y: 0.8503 - val_loss: 0.4015 - val_accuracy: 0.8592
Epoch 40/100
1875/1875 [==============================] - 5s 3ms/step - loss: 0.4265 - accurac
y: 0.8499 - val_loss: 0.4014 - val_accuracy: 0.8593
Epoch 41/100
1875/1875 [==============================] - 5s 3ms/step - loss: 0.4244 - accurac
y: 0.8496 - val_loss: 0.4102 - val_accuracy: 0.8538
Epoch 42/100
1875/1875 [==============================] - 5s 3ms/step - loss: 0.4241 - accurac
y: 0.8498 - val_loss: 0.4108 - val_accuracy: 0.8567
Epoch 43/100
1875/1875 [==============================] - 5s 3ms/step - loss: 0.4234 - accurac
y: 0.8483 - val_loss: 0.4056 - val_accuracy: 0.8583
Epoch 44/100
1875/1875 [==============================] - 5s 3ms/step - loss: 0.4235 - accurac
y: 0.8528 - val_loss: 0.4056 - val_accuracy: 0.8617
Epoch 45/100
1875/1875 [==============================] - 5s 3ms/step - loss: 0.4219 - accurac
y: 0.8506 - val_loss: 0.4023 - val_accuracy: 0.8606
Epoch 46/100
1875/1875 [==============================] - 5s 3ms/step - loss: 0.4227 - accurac
y: 0.8499 - val_loss: 0.4034 - val_accuracy: 0.8598
Epoch 47/100
1875/1875 [==============================] - 5s 3ms/step - loss: 0.4233 - accurac
y: 0.8501 - val_loss: 0.4051 - val_accuracy: 0.8610
Epoch 48/100
1875/1875 [==============================] - 5s 3ms/step - loss: 0.4216 - accurac
y: 0.8507 - val_loss: 0.4032 - val_accuracy: 0.8583
Epoch 49/100
1875/1875 [==============================] - 5s 3ms/step - loss: 0.4224 - accurac
y: 0.8507 - val_loss: 0.4076 - val_accuracy: 0.8602
Epoch 50/100
1875/1875 [==============================] - 5s 3ms/step - loss: 0.4207 - accurac
y: 0.8510 - val_loss: 0.4024 - val_accuracy: 0.8593
Epoch 51/100
1875/1875 [==============================] - 5s 3ms/step - loss: 0.4216 - accurac
y: 0.8514 - val_loss: 0.4012 - val_accuracy: 0.8598
Epoch 52/100
1875/1875 [==============================] - 5s 3ms/step - loss: 0.4210 - accurac
y: 0.8522 - val_loss: 0.3984 - val_accuracy: 0.8591
Epoch 53/100
1875/1875 [==============================] - 5s 3ms/step - loss: 0.4207 - accurac
y: 0.8525 - val_loss: 0.4030 - val_accuracy: 0.8597
Epoch 54/100
1875/1875 [==============================] - 5s 3ms/step - loss: 0.4215 - accurac
y: 0.8516 - val_loss: 0.4025 - val_accuracy: 0.8612
Epoch 55/100
1875/1875 [==============================] - 5s 3ms/step - loss: 0.4195 - accurac
y: 0.8508 - val_loss: 0.4000 - val_accuracy: 0.8603
Epoch 56/100
1875/1875 [==============================] - 5s 3ms/step - loss: 0.4192 - accurac
y: 0.8522 - val_loss: 0.4015 - val_accuracy: 0.8611
```

```
Epoch 57/100
1875/1875 [==============================] - 5s 3ms/step - loss: 0.4183 - accurac
y: 0.8524 - val_loss: 0.3981 - val_accuracy: 0.8615
Epoch 58/100
1875/1875 [==============================] - 5s 3ms/step - loss: 0.4200 - accurac
y: 0.8521 - val_loss: 0.3992 - val_accuracy: 0.8590
Epoch 59/100
1875/1875 [==============================] - 5s 3ms/step - loss: 0.4175 - accurac
y: 0.8516 - val_loss: 0.4035 - val_accuracy: 0.8591
Epoch 60/100
1875/1875 [==============================] - 5s 3ms/step - loss: 0.4199 - accurac
y: 0.8524 - val_loss: 0.3950 - val_accuracy: 0.8644
Epoch 61/100
1875/1875 [==============================] - 5s 3ms/step - loss: 0.4169 - accurac
y: 0.8527 - val_loss: 0.3971 - val_accuracy: 0.8608
Epoch 62/100
1875/1875 [==============================] - 5s 3ms/step - loss: 0.4172 - accurac
y: 0.8523 - val_loss: 0.3934 - val_accuracy: 0.8627
Epoch 63/100
1875/1875 [==============================] - 5s 3ms/step - loss: 0.4161 - accurac
y: 0.8519 - val_loss: 0.3997 - val_accuracy: 0.8605
Epoch 64/100
1875/1875 [==============================] - 5s 3ms/step - loss: 0.4159 - accurac
y: 0.8536 - val_loss: 0.3973 - val_accuracy: 0.8629
Epoch 65/100
1875/1875 [==============================] - 5s 3ms/step - loss: 0.4139 - accurac
y: 0.8533 - val_loss: 0.4006 - val_accuracy: 0.8592
Epoch 66/100
1875/1875 [==============================] - 5s 3ms/step - loss: 0.4154 - accurac
y: 0.8542 - val_loss: 0.3979 - val_accuracy: 0.8620
Epoch 67/100
1875/1875 [==============================] - 5s 3ms/step - loss: 0.4141 - accurac
y: 0.8550 - val_loss: 0.3957 - val_accuracy: 0.8629
Epoch 68/100
1875/1875 [==============================] - 5s 3ms/step - loss: 0.4145 - accurac
y: 0.8533 - val_loss: 0.3928 - val_accuracy: 0.8621
Epoch 69/100
1875/1875 [==============================] - 5s 3ms/step - loss: 0.4153 - accurac
y: 0.8536 - val_loss: 0.3949 - val_accuracy: 0.8638
Epoch 70/100
1875/1875 [==============================] - 5s 3ms/step - loss: 0.4131 - accurac
y: 0.8543 - val_loss: 0.4015 - val_accuracy: 0.8605
Epoch 71/100
1875/1875 [==============================] - 5s 3ms/step - loss: 0.4137 - accurac
y: 0.8532 - val_loss: 0.3919 - val_accuracy: 0.8628
Epoch 72/100
1875/1875 [==============================] - 5s 3ms/step - loss: 0.4122 - accurac
y: 0.8546 - val_loss: 0.3952 - val_accuracy: 0.8640
Epoch 73/100
1875/1875 [==============================] - 5s 3ms/step - loss: 0.4139 - accurac
y: 0.8542 - val_loss: 0.3928 - val_accuracy: 0.8631
Epoch 74/100
1875/1875 [==============================] - 5s 3ms/step - loss: 0.4149 - accurac
y: 0.8530 - val_loss: 0.3905 - val_accuracy: 0.8637
Epoch 75/100
1875/1875 [==============================] - 5s 3ms/step - loss: 0.4140 - accurac
```

```
y: 0.8535 - val_loss: 0.3900 - val_accuracy: 0.8654
Epoch 76/100
1875/1875 [==============================] - 5s 3ms/step - loss: 0.4146 - accurac
y: 0.8539 - val_loss: 0.3984 - val_accuracy: 0.8611
Epoch 77/100
1875/1875 [==============================] - 5s 3ms/step - loss: 0.4134 - accurac
y: 0.8538 - val_loss: 0.3913 - val_accuracy: 0.8627
Epoch 78/100
1875/1875 [==============================] - 5s 3ms/step - loss: 0.4109 - accurac
y: 0.8551 - val_loss: 0.3879 - val_accuracy: 0.8630
Epoch 79/100
1875/1875 [==============================] - 5s 3ms/step - loss: 0.4108 - accurac
y: 0.8537 - val_loss: 0.3952 - val_accuracy: 0.8641
Epoch 80/100
1875/1875 [==============================] - 5s 3ms/step - loss: 0.4101 - accurac
y: 0.8542 - val_loss: 0.3890 - val_accuracy: 0.8647
Epoch 81/100
1875/1875 [==============================] - 5s 3ms/step - loss: 0.4106 - accurac
y: 0.8543 - val_loss: 0.3898 - val_accuracy: 0.8653
Epoch 82/100
1875/1875 [==============================] - 6s 3ms/step - loss: 0.4118 - accurac
y: 0.8535 - val_loss: 0.4002 - val_accuracy: 0.8602
Epoch 83/100
1875/1875 [==============================] - 5s 3ms/step - loss: 0.4142 - accurac
y: 0.8537 - val_loss: 0.3919 - val_accuracy: 0.8630
Epoch 84/100
1875/1875 [==============================] - 5s 3ms/step - loss: 0.4092 - accurac
y: 0.8554 - val_loss: 0.3940 - val_accuracy: 0.8615
Epoch 85/100
1875/1875 [==============================] - 5s 3ms/step - loss: 0.4119 - accurac
y: 0.8540 - val_loss: 0.3902 - val_accuracy: 0.8653
Epoch 86/100
1875/1875 [==============================] - 5s 3ms/step - loss: 0.4106 - accurac
y: 0.8558 - val_loss: 0.3964 - val_accuracy: 0.8623
Epoch 87/100
1875/1875 [==============================] - 5s 3ms/step - loss: 0.4085 - accurac
y: 0.8547 - val_loss: 0.3924 - val_accuracy: 0.8633
Epoch 88/100
1875/1875 [==============================] - 5s 3ms/step - loss: 0.4110 - accurac
y: 0.8543 - val_loss: 0.3897 - val_accuracy: 0.8653
Epoch 89/100
1875/1875 [==============================] - 5s 3ms/step - loss: 0.4096 - accurac
y: 0.8556 - val_loss: 0.3870 - val_accuracy: 0.8646
Epoch 90/100
1875/1875 [==============================] - 5s 3ms/step - loss: 0.4091 - accurac
y: 0.8542 - val_loss: 0.3871 - val_accuracy: 0.8666
Epoch 91/100
1875/1875 [==============================] - 5s 3ms/step - loss: 0.4114 - accurac
y: 0.8532 - val_loss: 0.3889 - val_accuracy: 0.8660
Epoch 92/100
1875/1875 [==============================] - 5s 3ms/step - loss: 0.4088 - accurac
y: 0.8549 - val_loss: 0.3886 - val_accuracy: 0.8650
Epoch 93/100
1875/1875 [==============================] - 5s 3ms/step - loss: 0.4092 - accurac
y: 0.8565 - val_loss: 0.3878 - val_accuracy: 0.8661
Epoch 94/100
```

```
1875/1875 [==============================] - 5s 3ms/step - loss: 0.4102 - accurac
y: 0.8548 - val_loss: 0.3857 - val_accuracy: 0.8656
Epoch 95/100
1875/1875 [==============================] - 5s 3ms/step - loss: 0.4089 - accurac
y: 0.8547 - val_loss: 0.3870 - val_accuracy: 0.8651
Epoch 96/100
1875/1875 [==============================] - 5s 3ms/step - loss: 0.4092 - accurac
y: 0.8543 - val_loss: 0.3883 - val_accuracy: 0.8646
Epoch 97/100
1875/1875 [==============================] - 5s 3ms/step - loss: 0.4094 - accurac
y: 0.8551 - val_loss: 0.3861 - val_accuracy: 0.8644
Epoch 98/100
1875/1875 [==============================] - 5s 3ms/step - loss: 0.4056 - accurac
y: 0.8555 - val_loss: 0.3854 - val_accuracy: 0.8662
Epoch 99/100
1875/1875 [==============================] - 5s 3ms/step - loss: 0.4113 - accurac
y: 0.8534 - val_loss: 0.3864 - val_accuracy: 0.8632
Epoch 100/100
1875/1875 [==============================] - 5s 3ms/step - loss: 0.4073 - accurac
y: 0.8549 - val_loss: 0.3866 - val_accuracy: 0.8640
313/313 [==============================] - 0s 1ms/step - loss: 0.3866 - accuracy:
0.8640
1875/1875 [==============================] - 2s 1ms/step - loss: 0.3596 - accurac
y: 0.8762
    loss     (train, test): 0.3596,  0.3866
    accuracy (train, test): 0.8762,  0.8640
```






```
Train accuracy: 0.8761833310127258
Test accuracy: 0.8640000224113464
```