ECE285 Assignment 1: Neural Network in NumPy

Use this notebook to build your neural network by implementing the following functions in the python files under `ece285/algorithms` directory:

1. `linear.py`
2. `relu.py`
3. `softmax.py`
4. `loss_func.py`

You will be testing your 2 layer neural network implementation on a toy dataset.

TO SUBMIT: PDF of this notebook with all the required outputs and answers.

In [ ]:
```python
# Setup
import matplotlib.pyplot as plt
import numpy as np

from ece285.layers.sequential import Sequential
from ece285.layers.linear import Linear
from ece285.layers.relu import ReLU
from ece285.layers.softmax import Softmax
from ece285.layers.loss_func import CrossEntropyLoss
from ece285.utils.optimizer import SGD

%matplotlib inline
plt.rcParams["figure.figsize"] = (10.0, 8.0)  # set default size of plots

# For auto-reloading external modules
# See http://stackoverflow.com/questions/1907993/autoreload-of-modules-in-ipython
%load_ext autoreload
%autoreload 2
```

We will use the class Sequential as implemented in the file assignment2/layers/sequential.py to build a layer by layer model of our neural network. Below we initialize the toy model and the toy random data that you will use to develop your implementation.

In [ ]:
```python
# Create a small net and some toy data to check your implementations.
# Note that we set the random seed for repeatable experiments.

input_size = 4
hidden_size = 10
num_classes = 3  # Output
num_inputs = 10  # N


def init_toy_model():
    np.random.seed(0)
    l1 = Linear(input_size, hidden_size)
    l2 = Linear(hidden_size, num_classes)

    r1 = ReLU()
    softmax = Softmax()
    return Sequential([l1, r1, l2, softmax])


def init_toy_data():
    np.random.seed(0)
    X = 10 * np.random.randn(num_inputs, input_size)
    y = np.random.randint(num_classes, size=num_inputs)
    # y = np.array([0, 1, 2, 2, 1])
    return X, y


net = init_toy_model()
X, y = init_toy_data()
```

Forward Pass: Compute Scores (20%)

Implement the forward functions in Linear, Relu and Softmax layers and get the output by passing our toy data X The output must match the given output scores

In [ ]:
```
scores = net.forward(X)
print("Your scores:")
print(scores)
print()
print("correct scores:")
correct_scores = np.asarray(
    [
        [0.33333514, 0.33333826, 0.33332661],
        [0.3333351, 0.33333828, 0.33332661],
        [0.3333351, 0.33333828, 0.33332662],
        [0.3333351, 0.33333828, 0.33332662],
        [0.33333509, 0.33333829, 0.33332662],
        [0.33333508, 0.33333829, 0.33332662],
        [0.33333511, 0.33333828, 0.33332661],
        [0.33333512, 0.33333827, 0.33332661],
        [0.33333508, 0.33333829, 0.33332662],
        [0.33333511, 0.33333828, 0.33332662],
    ]
)
print(correct_scores)

# The difference should be very small. We get < 1e-7
print("Difference between your scores and correct scores:")
print(np.sum(np.abs(scores - correct_scores)))
```

```
Your scores:
[[0.33333514 0.33333826 0.33332661]
 [0.3333351  0.33333828 0.33332661]
 [0.3333351  0.33333828 0.33332662]
 [0.3333351  0.33333828 0.33332662]
 [0.33333509 0.33333829 0.33332662]
 [0.33333508 0.33333829 0.33332662]
 [0.33333511 0.33333828 0.33332661]
 [0.33333512 0.33333827 0.33332661]
 [0.33333508 0.33333829 0.33332662]
 [0.33333511 0.33333828 0.33332662]]

correct scores:
[[0.33333514 0.33333826 0.33332661]
 [0.3333351  0.33333828 0.33332661]
 [0.3333351  0.33333828 0.33332662]
 [0.3333351  0.33333828 0.33332662]
 [0.33333509 0.33333829 0.33332662]
 [0.33333508 0.33333829 0.33332662]
 [0.33333511 0.33333828 0.33332661]
 [0.33333512 0.33333827 0.33332661]
 [0.33333508 0.33333829 0.33332662]
 [0.33333511 0.33333828 0.33332662]]
Difference between your scores and correct scores:
8.799388540037256e-08
```

Forward Pass: Compute loss given the output scores from the previous step (10%)

Implement the forward function in the loss_func.py file, and output the loss value. The loss value must match the given loss value.

In [ ]:
```
Loss = CrossEntropyLoss()
loss = Loss.forward(scores, y)
correct_loss = 1.098612723362578
print(loss)
# should be very small, we get < 1e-12
print("Difference between your loss and correct loss:")
print(np.sum(np.abs(loss - correct_loss)))
```

```
1.098612723362578
Difference between your loss and correct loss:
0.0
```

Backward Pass (40%)

Implement the rest of the functions in the given files. Specifically, implement the backward function in all the 4 files as mentioned in the files. Note: No backward function in the softmax file, the gradient for softmax is jointly calculated with the cross entropy loss in the loss_func.backward function.

You will use the chain rule to calculate gradient individually for each layer. You can assume that this calculated gradeint then is passed to the next layers in a reversed manner due to the Sequential implementation. So all you need to worry about is implementing the gradient for the current layer and multiply it will the incoming gradient (passed to the backward function as dout) to calculate the total gradient for the parameters of that layer.

We check the values for these gradients by calculating the difference, it is expected to get difference < 1e-8.

In [ ]:
```python
# No need to edit anything in this block ( 20% of the above 40% )
net.backward(Loss.backward())

gradients = []
for module in net._modules:
    # print(module)
    for para, grad in zip(module.parameters, module.grads):
        assert grad is not None, "No Gradient"
        # Print gradients of the linear layer
        # print(grad)
        print(grad.shape)
        gradients.append(grad)

# Check shapes of your gradient. Note that only the linear layer has parameters
# (4, 10) -> Layer 1 W
# (10,)   -> Layer 1 b
# (10, 3) -> Layer 2 W
# (3,)    -> Layer 2 b
```

```
(4, 10)
(10,)
(10, 3)
(3,)
```

In [ ]:
```python
# No need to edit anything in this block ( 20% of the above 40% )
grad_w1 = np.array(
    [
        [
            -6.24320917e-05,
            3.41037180e-06,
            -1.69125969e-05,
            2.41514079e-05,
            3.88697976e-06,
            7.63842314e-05,
            -8.88925758e-05,
            3.34909890e-05,
            -1.42758303e-05,
            -4.74748560e-06,
        ],
        [
            -7.16182867e-05,
            4.63270039e-06,
            -2.20344270e-05,
            -2.72027034e-06,
            6.52903437e-07,
            8.97294847e-05,
            -1.05981609e-04,
            4.15825391e-05,
            -2.12210745e-05,
            3.06061658e-05,
        ],
        [
            -1.69074923e-05,
            -8.83185056e-06,
            3.10730840e-05,
            1.23010428e-05,
            5.25830316e-05,
            -7.82980115e-06,
            3.02117990e-05,
            -3.37645284e-05,
            6.17276346e-05,
            -1.10735656e-05,
```

```
        ],
        [
            -4.35902272e-05,
            3.71512704e-06,
            -1.66837877e-05,
            2.54069557e-06,
            -4.33258099e-06,
            5.72310022e-05,
            -6.94881762e-05,
            2.92408329e-05,
            -1.89369767e-05,
            2.01692516e-05,
        ],
    ]
)
grad_b1 = np.array(
    [
        -2.27150209e-06,
        5.14674340e-07,
        -2.04284403e-06,
        6.08849787e-07,
        -1.92177796e-06,
        3.92085824e-06,
        -5.40772636e-06,
        2.93354593e-06,
        -3.14568138e-06,
        5.27501592e-11,
    ]
)

grad_w2 = np.array(
    [
        [1.28932983e-04, 1.19946731e-04, -2.48879714e-04],
        [1.08784150e-04, 1.55140199e-04, -2.63924349e-04],
        [6.96017544e-05, 1.42748410e-04, -2.12350164e-04],
        [9.92512487e-05, 1.73257611e-04, -2.72508860e-04],
        [2.05484895e-05, 4.96161144e-05, -7.01646039e-05],
        [8.20539510e-05, 9.37063861e-05, -1.75760337e-04],
        [2.45831715e-05, 8.74369112e-05, -1.12020083e-04],
        [1.34073379e-04, 1.86253064e-04, -3.20326443e-04],
        [8.86473128e-05, 2.35554414e-04, -3.24201726e-04],
        [3.57433149e-05, 1.91164061e-04, -2.26907376e-04],
    ]
)

grad_b2 = np.array([-0.1666649, 0.13333828, 0.03332662])

difference = (
    np.sum(np.abs(gradients[0] - grad_w1))
    + np.sum(np.abs(gradients[1] - grad_b1))
    + np.sum(np.abs(gradients[2] - grad_w2))
    + np.sum(np.abs(gradients[3] - grad_b2))
)
print("Difference in Gradient values", difference)
```

```
Difference in Gradient values 7.70191643436727e-09
```

Train the complete network on the toy data. (30%)

To train the network we will use stochastic gradient descent (SGD), we have implemented the optimizer for you. You do not implement any more functions in the python files. Below we implement the training procedure, you should get yourself familiar with the training process. Specifically looking at which functions to call and when.

Once you have implemented the method and tested various parts in the above blocks, run the code below to train a two-layer network on toy data. You should see your training loss decrease below 0.01.

In [ ]:

```
# Training Procedure
# Initialize the optimizer. DO NOT change any of the hyper-parameters here or above.
# We have implemented the SGD optimizer class for you here, which visits each layer sequentially to
# get the gradients and optimize the respective parameters.
# You should work with the given parameters and only edit your implementation in the .py files

epochs = 1000
```

```python
optim = SGD(net, lr=0.1, weight_decay=0.00001)

epoch_loss = []
for epoch in range(epochs):
    # Get output scores from the network
    output_x = net(X)
    # Calculate the loss for these output scores, given the true labels
    # print(output_x, y)
    loss = Loss.forward(output_x, y)
    # print(loss)
    # Initialize your gradients to None in each epoch
    optim.zero_grad()
    # Make a backward pass to update the internal gradients in the layers
    net.backward(Loss.backward())
    # call the step function in the optimizer to update the values of the params with the gradients
    optim.step()
    # Append the loss at each iteration
    epoch_loss.append(loss)
    print("Epoch {}, loss={:3f}".format(epoch + 1, epoch_loss[-1]))

    if (epoch + 1) % 50 == 0:
        print("Epoch {}, loss={:3f}".format(epoch + 1, epoch_loss[-1]))
```

```
Epoch 1, loss=1.098613
Epoch 2, loss=1.094024
Epoch 3, loss=1.089737
Epoch 4, loss=1.085733
Epoch 5, loss=1.081994
Epoch 6, loss=1.078505
Epoch 7, loss=1.075248
Epoch 8, loss=1.072208
Epoch 9, loss=1.069372
Epoch 10, loss=1.066726
Epoch 11, loss=1.064257
Epoch 12, loss=1.061952
Epoch 13, loss=1.059799
Epoch 14, loss=1.057785
Epoch 15, loss=1.055899
Epoch 16, loss=1.054124
Epoch 17, loss=1.052445
Epoch 18, loss=1.050839
Epoch 19, loss=1.049277
Epoch 20, loss=1.047713
Epoch 21, loss=1.046081
Epoch 22, loss=1.044279
Epoch 23, loss=1.042147
Epoch 24, loss=1.039435
Epoch 25, loss=1.035774
Epoch 26, loss=1.030637
Epoch 27, loss=1.023365
Epoch 28, loss=1.013337
Epoch 29, loss=1.000400
Epoch 30, loss=0.985502
Epoch 31, loss=0.970780
Epoch 32, loss=0.958224
Epoch 33, loss=0.948146
Epoch 34, loss=0.939687
Epoch 35, loss=0.932411
Epoch 36, loss=0.926597
Epoch 37, loss=0.920919
Epoch 38, loss=0.914630
Epoch 39, loss=0.908446
Epoch 40, loss=0.902708
Epoch 41, loss=0.895794
Epoch 42, loss=0.889273
Epoch 43, loss=0.882132
Epoch 44, loss=0.875647
Epoch 45, loss=0.870536
Epoch 46, loss=0.861775
Epoch 47, loss=0.855117
Epoch 48, loss=0.848626
Epoch 49, loss=0.839904
Epoch 50, loss=0.832706
Epoch 50, loss=0.832706
Epoch 51, loss=0.827365
Epoch 52, loss=0.815973
Epoch 53, loss=0.810844
Epoch 54, loss=0.802194
Epoch 55, loss=0.790601
Epoch 56, loss=0.783502
Epoch 57, loss=0.770632
Epoch 58, loss=0.760160
Epoch 59, loss=0.749472
Epoch 60, loss=0.739295
Epoch 61, loss=0.732478
Epoch 62, loss=0.719142
Epoch 63, loss=0.713844
Epoch 64, loss=0.700037
Epoch 65, loss=0.693629
Epoch 66, loss=0.689958
Epoch 67, loss=0.674298
Epoch 68, loss=0.659023
Epoch 69, loss=0.647852
Epoch 70, loss=0.638275
Epoch 71, loss=0.628526
Epoch 72, loss=0.622772
Epoch 73, loss=0.617140
Epoch 74, loss=0.608889
Epoch 75, loss=0.601315
```

```
Epoch 76, loss=0.590965
Epoch 77, loss=0.588483
Epoch 78, loss=0.580111
Epoch 79, loss=0.580237
Epoch 80, loss=0.579590
Epoch 81, loss=0.575265
Epoch 82, loss=0.583167
Epoch 83, loss=0.569299
Epoch 84, loss=0.567051
Epoch 85, loss=0.558750
Epoch 86, loss=0.570455
Epoch 87, loss=0.550303
Epoch 88, loss=0.556647
Epoch 89, loss=0.528476
Epoch 90, loss=0.523165
Epoch 91, loss=0.503276
Epoch 92, loss=0.508981
Epoch 93, loss=0.493973
Epoch 94, loss=0.512211
Epoch 95, loss=0.477031
Epoch 96, loss=0.483331
Epoch 97, loss=0.456475
Epoch 98, loss=0.472357
Epoch 99, loss=0.443569
Epoch 100, loss=0.454687
Epoch 100, loss=0.454687
Epoch 101, loss=0.429413
Epoch 102, loss=0.441666
Epoch 103, loss=0.413838
Epoch 104, loss=0.443494
Epoch 105, loss=0.407146
Epoch 106, loss=0.444991
Epoch 107, loss=0.403453
Epoch 108, loss=0.436572
Epoch 109, loss=0.397764
Epoch 110, loss=0.439797
Epoch 111, loss=0.422846
Epoch 112, loss=0.526893
Epoch 113, loss=0.553876
Epoch 114, loss=0.728167
Epoch 115, loss=0.582263
Epoch 116, loss=0.547702
Epoch 117, loss=0.416240
Epoch 118, loss=0.453693
Epoch 119, loss=0.381732
Epoch 120, loss=0.362521
Epoch 121, loss=0.324897
Epoch 122, loss=0.338047
Epoch 123, loss=0.342623
Epoch 124, loss=0.326819
Epoch 125, loss=0.392589
Epoch 126, loss=0.292861
Epoch 127, loss=0.314854
Epoch 128, loss=0.235608
Epoch 129, loss=0.227234
Epoch 130, loss=0.207276
Epoch 131, loss=0.220121
Epoch 132, loss=0.208397
Epoch 133, loss=0.249017
Epoch 134, loss=0.226502
Epoch 135, loss=0.284894
Epoch 136, loss=0.233826
Epoch 137, loss=0.259128
Epoch 138, loss=0.195131
Epoch 139, loss=0.177242
Epoch 140, loss=0.156995
Epoch 141, loss=0.147287
Epoch 142, loss=0.142642
Epoch 143, loss=0.139515
Epoch 144, loss=0.136126
Epoch 145, loss=0.135978
Epoch 146, loss=0.130053
Epoch 147, loss=0.127937
Epoch 148, loss=0.122477
Epoch 149, loss=0.120641
Epoch 150, loss=0.118350
```

```
Epoch 150, loss=0.118350
Epoch 151, loss=0.117382
Epoch 152, loss=0.112601
Epoch 153, loss=0.111264
Epoch 154, loss=0.106784
Epoch 155, loss=0.105601
Epoch 156, loss=0.105261
Epoch 157, loss=0.103690
Epoch 158, loss=0.099740
Epoch 159, loss=0.097608
Epoch 160, loss=0.094972
Epoch 161, loss=0.094597
Epoch 162, loss=0.093439
Epoch 163, loss=0.092069
Epoch 164, loss=0.090610
Epoch 165, loss=0.087682
Epoch 166, loss=0.085767
Epoch 167, loss=0.084664
Epoch 168, loss=0.083181
Epoch 169, loss=0.082531
Epoch 170, loss=0.081955
Epoch 171, loss=0.079550
Epoch 172, loss=0.078298
Epoch 173, loss=0.077149
Epoch 174, loss=0.075943
Epoch 175, loss=0.074927
Epoch 176, loss=0.074027
Epoch 177, loss=0.074400
Epoch 178, loss=0.072842
Epoch 179, loss=0.071362
Epoch 180, loss=0.070140
Epoch 181, loss=0.069109
Epoch 182, loss=0.068189
Epoch 183, loss=0.067438
Epoch 184, loss=0.066542
Epoch 185, loss=0.065878
Epoch 186, loss=0.065619
Epoch 187, loss=0.065236
Epoch 188, loss=0.064023
Epoch 189, loss=0.063155
Epoch 190, loss=0.062142
Epoch 191, loss=0.061411
Epoch 192, loss=0.060913
Epoch 193, loss=0.060049
Epoch 194, loss=0.059368
Epoch 195, loss=0.059265
Epoch 196, loss=0.058653
Epoch 197, loss=0.057884
Epoch 198, loss=0.057724
Epoch 199, loss=0.056640
Epoch 200, loss=0.055911
Epoch 200, loss=0.055911
Epoch 201, loss=0.055581
Epoch 202, loss=0.054974
Epoch 203, loss=0.054336
Epoch 204, loss=0.053731
Epoch 205, loss=0.053237
Epoch 206, loss=0.052698
Epoch 207, loss=0.052610
Epoch 208, loss=0.052149
Epoch 209, loss=0.051605
Epoch 210, loss=0.051018
Epoch 211, loss=0.050427
Epoch 212, loss=0.050176
Epoch 213, loss=0.050089
Epoch 214, loss=0.049610
Epoch 215, loss=0.048936
Epoch 216, loss=0.048407
Epoch 217, loss=0.048042
Epoch 218, loss=0.047693
Epoch 219, loss=0.047232
Epoch 220, loss=0.047006
Epoch 221, loss=0.046590
Epoch 222, loss=0.046342
Epoch 223, loss=0.045836
Epoch 224, loss=0.045452
```

```
Epoch 225, loss=0.045112
Epoch 226, loss=0.044795
Epoch 227, loss=0.044464
Epoch 228, loss=0.044038
Epoch 229, loss=0.043724
Epoch 230, loss=0.043653
Epoch 231, loss=0.043454
Epoch 232, loss=0.042933
Epoch 233, loss=0.042565
Epoch 234, loss=0.042280
Epoch 235, loss=0.041901
Epoch 236, loss=0.041709
Epoch 237, loss=0.041528
Epoch 238, loss=0.041169
Epoch 239, loss=0.040963
Epoch 240, loss=0.040683
Epoch 241, loss=0.040507
Epoch 242, loss=0.040178
Epoch 243, loss=0.039961
Epoch 244, loss=0.039609
Epoch 245, loss=0.039391
Epoch 246, loss=0.039159
Epoch 247, loss=0.038825
Epoch 248, loss=0.038546
Epoch 249, loss=0.038335
Epoch 250, loss=0.038039
Epoch 250, loss=0.038039
Epoch 251, loss=0.037768
Epoch 252, loss=0.037602
Epoch 253, loss=0.037523
Epoch 254, loss=0.037224
Epoch 255, loss=0.036954
Epoch 256, loss=0.036836
Epoch 257, loss=0.036784
Epoch 258, loss=0.036457
Epoch 259, loss=0.036186
Epoch 260, loss=0.035966
Epoch 261, loss=0.035764
Epoch 262, loss=0.035549
Epoch 263, loss=0.035444
Epoch 264, loss=0.035394
Epoch 265, loss=0.035143
Epoch 266, loss=0.034874
Epoch 267, loss=0.034649
Epoch 268, loss=0.034474
Epoch 269, loss=0.034245
Epoch 270, loss=0.034032
Epoch 271, loss=0.033865
Epoch 272, loss=0.033680
Epoch 273, loss=0.033470
Epoch 274, loss=0.033298
Epoch 275, loss=0.033177
Epoch 276, loss=0.032968
Epoch 277, loss=0.032779
Epoch 278, loss=0.032638
Epoch 279, loss=0.032442
Epoch 280, loss=0.032259
Epoch 281, loss=0.032117
Epoch 282, loss=0.032015
Epoch 283, loss=0.032165
Epoch 284, loss=0.031888
Epoch 285, loss=0.031721
Epoch 286, loss=0.031566
Epoch 287, loss=0.031528
Epoch 288, loss=0.031292
Epoch 289, loss=0.031094
Epoch 290, loss=0.030956
Epoch 291, loss=0.030888
Epoch 292, loss=0.030697
Epoch 293, loss=0.030521
Epoch 294, loss=0.030383
Epoch 295, loss=0.030241
Epoch 296, loss=0.030111
Epoch 297, loss=0.029963
Epoch 298, loss=0.029830
Epoch 299, loss=0.029668
```

```
Epoch 300, loss=0.029528
Epoch 300, loss=0.029528
Epoch 301, loss=0.029408
Epoch 302, loss=0.029255
Epoch 303, loss=0.029114
Epoch 304, loss=0.029005
Epoch 305, loss=0.028858
Epoch 306, loss=0.028718
Epoch 307, loss=0.028612
Epoch 308, loss=0.028480
Epoch 309, loss=0.028342
Epoch 310, loss=0.028228
Epoch 311, loss=0.028112
Epoch 312, loss=0.028042
Epoch 313, loss=0.028009
Epoch 314, loss=0.027869
Epoch 315, loss=0.027722
Epoch 316, loss=0.027666
Epoch 317, loss=0.027744
Epoch 318, loss=0.027542
Epoch 319, loss=0.027385
Epoch 320, loss=0.027244
Epoch 321, loss=0.027115
Epoch 322, loss=0.027015
Epoch 323, loss=0.026908
Epoch 324, loss=0.026801
Epoch 325, loss=0.026707
Epoch 326, loss=0.026660
Epoch 327, loss=0.026527
Epoch 328, loss=0.026404
Epoch 329, loss=0.026311
Epoch 330, loss=0.026192
Epoch 331, loss=0.026078
Epoch 332, loss=0.025984
Epoch 333, loss=0.025883
Epoch 334, loss=0.025771
Epoch 335, loss=0.025673
Epoch 336, loss=0.025583
Epoch 337, loss=0.025474
Epoch 338, loss=0.025383
Epoch 339, loss=0.025316
Epoch 340, loss=0.025206
Epoch 341, loss=0.025106
Epoch 342, loss=0.025025
Epoch 343, loss=0.024921
Epoch 344, loss=0.024822
Epoch 345, loss=0.024753
Epoch 346, loss=0.024686
Epoch 347, loss=0.024655
Epoch 348, loss=0.024557
Epoch 349, loss=0.024513
Epoch 350, loss=0.024400
Epoch 350, loss=0.024400
Epoch 351, loss=0.024295
Epoch 352, loss=0.024202
Epoch 353, loss=0.024120
Epoch 354, loss=0.024023
Epoch 355, loss=0.023949
Epoch 356, loss=0.023882
Epoch 357, loss=0.023873
Epoch 358, loss=0.023803
Epoch 359, loss=0.023696
Epoch 360, loss=0.023608
Epoch 361, loss=0.023513
Epoch 362, loss=0.023421
Epoch 363, loss=0.023337
Epoch 364, loss=0.023263
Epoch 365, loss=0.023175
Epoch 366, loss=0.023089
Epoch 367, loss=0.023058
Epoch 368, loss=0.022986
Epoch 369, loss=0.022898
Epoch 370, loss=0.022820
Epoch 371, loss=0.022764
Epoch 372, loss=0.022686
Epoch 373, loss=0.022602
```

```
Epoch 374, loss=0.022536
Epoch 375, loss=0.022452
Epoch 376, loss=0.022371
Epoch 377, loss=0.022310
Epoch 378, loss=0.022230
Epoch 379, loss=0.022152
Epoch 380, loss=0.022091
Epoch 381, loss=0.022017
Epoch 382, loss=0.021940
Epoch 383, loss=0.021877
Epoch 384, loss=0.021808
Epoch 385, loss=0.021733
Epoch 386, loss=0.021676
Epoch 387, loss=0.021642
Epoch 388, loss=0.021625
Epoch 389, loss=0.021543
Epoch 390, loss=0.021474
Epoch 391, loss=0.021424
Epoch 392, loss=0.021348
Epoch 393, loss=0.021282
Epoch 394, loss=0.021211
Epoch 395, loss=0.021137
Epoch 396, loss=0.021074
Epoch 397, loss=0.021011
Epoch 398, loss=0.020940
Epoch 399, loss=0.020878
Epoch 400, loss=0.020819
Epoch 400, loss=0.020819
Epoch 401, loss=0.020751
Epoch 402, loss=0.020688
Epoch 403, loss=0.020633
Epoch 404, loss=0.020566
Epoch 405, loss=0.020504
Epoch 406, loss=0.020451
Epoch 407, loss=0.020385
Epoch 408, loss=0.020313
Epoch 409, loss=0.020251
Epoch 410, loss=0.020197
Epoch 411, loss=0.020169
Epoch 412, loss=0.020110
Epoch 413, loss=0.020046
Epoch 414, loss=0.019971
Epoch 415, loss=0.019898
Epoch 416, loss=0.019833
Epoch 417, loss=0.019767
Epoch 418, loss=0.019698
Epoch 419, loss=0.019641
Epoch 420, loss=0.019610
Epoch 421, loss=0.019540
Epoch 422, loss=0.019472
Epoch 423, loss=0.019407
Epoch 424, loss=0.019350
Epoch 425, loss=0.019300
Epoch 426, loss=0.019235
Epoch 427, loss=0.019172
Epoch 428, loss=0.019116
Epoch 429, loss=0.019053
Epoch 430, loss=0.018993
Epoch 431, loss=0.018957
Epoch 432, loss=0.018905
Epoch 433, loss=0.018855
Epoch 434, loss=0.018792
Epoch 435, loss=0.018782
Epoch 436, loss=0.018733
Epoch 437, loss=0.018664
Epoch 438, loss=0.018605
Epoch 439, loss=0.018545
Epoch 440, loss=0.018484
Epoch 441, loss=0.018426
Epoch 442, loss=0.018374
Epoch 443, loss=0.018317
Epoch 444, loss=0.018260
Epoch 445, loss=0.018212
Epoch 446, loss=0.018156
Epoch 447, loss=0.018101
Epoch 448, loss=0.018054
```

```
Epoch 449, loss=0.018001
Epoch 450, loss=0.017947
Epoch 450, loss=0.017947
Epoch 451, loss=0.017900
Epoch 452, loss=0.017862
Epoch 453, loss=0.017808
Epoch 454, loss=0.017761
Epoch 455, loss=0.017711
Epoch 456, loss=0.017659
Epoch 457, loss=0.017613
Epoch 458, loss=0.017564
Epoch 459, loss=0.017513
Epoch 460, loss=0.017468
Epoch 461, loss=0.017421
Epoch 462, loss=0.017371
Epoch 463, loss=0.017327
Epoch 464, loss=0.017281
Epoch 465, loss=0.017232
Epoch 466, loss=0.017188
Epoch 467, loss=0.017143
Epoch 468, loss=0.017096
Epoch 469, loss=0.017052
Epoch 470, loss=0.017009
Epoch 471, loss=0.016962
Epoch 472, loss=0.016919
Epoch 473, loss=0.016877
Epoch 474, loss=0.016831
Epoch 475, loss=0.016825
Epoch 476, loss=0.016779
Epoch 477, loss=0.016733
Epoch 478, loss=0.016688
Epoch 479, loss=0.016648
Epoch 480, loss=0.016615
Epoch 481, loss=0.016571
Epoch 482, loss=0.016538
Epoch 483, loss=0.016511
Epoch 484, loss=0.016489
Epoch 485, loss=0.016449
Epoch 486, loss=0.016401
Epoch 487, loss=0.016354
Epoch 488, loss=0.016312
Epoch 489, loss=0.016269
Epoch 490, loss=0.016226
Epoch 491, loss=0.016184
Epoch 492, loss=0.016159
Epoch 493, loss=0.016143
Epoch 494, loss=0.016097
Epoch 495, loss=0.016054
Epoch 496, loss=0.016014
Epoch 497, loss=0.015972
Epoch 498, loss=0.015933
Epoch 499, loss=0.015906
Epoch 500, loss=0.015866
Epoch 500, loss=0.015866
Epoch 501, loss=0.015825
Epoch 502, loss=0.015785
Epoch 503, loss=0.015748
Epoch 504, loss=0.015709
Epoch 505, loss=0.015670
Epoch 506, loss=0.015635
Epoch 507, loss=0.015597
Epoch 508, loss=0.015559
Epoch 509, loss=0.015523
Epoch 510, loss=0.015487
Epoch 511, loss=0.015450
Epoch 512, loss=0.015415
Epoch 513, loss=0.015380
Epoch 514, loss=0.015343
Epoch 515, loss=0.015308
Epoch 516, loss=0.015275
Epoch 517, loss=0.015238
Epoch 518, loss=0.015203
Epoch 519, loss=0.015171
Epoch 520, loss=0.015135
Epoch 521, loss=0.015100
Epoch 522, loss=0.015069
```

```
Epoch 523, loss=0.015042
Epoch 524, loss=0.015007
Epoch 525, loss=0.014976
Epoch 526, loss=0.014941
Epoch 527, loss=0.014907
Epoch 528, loss=0.014876
Epoch 529, loss=0.014842
Epoch 530, loss=0.014809
Epoch 531, loss=0.014777
Epoch 532, loss=0.014745
Epoch 533, loss=0.014712
Epoch 534, loss=0.014680
Epoch 535, loss=0.014649
Epoch 536, loss=0.014617
Epoch 537, loss=0.014585
Epoch 538, loss=0.014555
Epoch 539, loss=0.014523
Epoch 540, loss=0.014495
Epoch 541, loss=0.014474
Epoch 542, loss=0.014441
Epoch 543, loss=0.014409
Epoch 544, loss=0.014380
Epoch 545, loss=0.014349
Epoch 546, loss=0.014318
Epoch 547, loss=0.014288
Epoch 548, loss=0.014259
Epoch 549, loss=0.014228
Epoch 550, loss=0.014198
Epoch 550, loss=0.014198
Epoch 551, loss=0.014170
Epoch 552, loss=0.014139
Epoch 553, loss=0.014110
Epoch 554, loss=0.014082
Epoch 555, loss=0.014052
Epoch 556, loss=0.014022
Epoch 557, loss=0.014014
Epoch 558, loss=0.014001
Epoch 559, loss=0.013977
Epoch 560, loss=0.013963
Epoch 561, loss=0.013930
Epoch 562, loss=0.013898
Epoch 563, loss=0.013888
Epoch 564, loss=0.013871
Epoch 565, loss=0.013836
Epoch 566, loss=0.013803
Epoch 567, loss=0.013771
Epoch 568, loss=0.013741
Epoch 569, loss=0.013710
Epoch 570, loss=0.013680
Epoch 571, loss=0.013653
Epoch 572, loss=0.013631
Epoch 573, loss=0.013602
Epoch 574, loss=0.013579
Epoch 575, loss=0.013554
Epoch 576, loss=0.013525
Epoch 577, loss=0.013498
Epoch 578, loss=0.013470
Epoch 579, loss=0.013442
Epoch 580, loss=0.013414
Epoch 581, loss=0.013388
Epoch 582, loss=0.013362
Epoch 583, loss=0.013334
Epoch 584, loss=0.013308
Epoch 585, loss=0.013283
Epoch 586, loss=0.013257
Epoch 587, loss=0.013230
Epoch 588, loss=0.013204
Epoch 589, loss=0.013180
Epoch 590, loss=0.013154
Epoch 591, loss=0.013128
Epoch 592, loss=0.013104
Epoch 593, loss=0.013079
Epoch 594, loss=0.013053
Epoch 595, loss=0.013029
Epoch 596, loss=0.013005
Epoch 597, loss=0.012980
```

```
Epoch 598, loss=0.012956
Epoch 599, loss=0.012941
Epoch 600, loss=0.012916
Epoch 600, loss=0.012916
Epoch 601, loss=0.012891
Epoch 602, loss=0.012867
Epoch 603, loss=0.012843
Epoch 604, loss=0.012818
Epoch 605, loss=0.012797
Epoch 606, loss=0.012779
Epoch 607, loss=0.012754
Epoch 608, loss=0.012730
Epoch 609, loss=0.012707
Epoch 610, loss=0.012682
Epoch 611, loss=0.012658
Epoch 612, loss=0.012636
Epoch 613, loss=0.012612
Epoch 614, loss=0.012589
Epoch 615, loss=0.012567
Epoch 616, loss=0.012544
Epoch 617, loss=0.012520
Epoch 618, loss=0.012498
Epoch 619, loss=0.012476
Epoch 620, loss=0.012453
Epoch 621, loss=0.012431
Epoch 622, loss=0.012409
Epoch 623, loss=0.012386
Epoch 624, loss=0.012364
Epoch 625, loss=0.012343
Epoch 626, loss=0.012320
Epoch 627, loss=0.012298
Epoch 628, loss=0.012277
Epoch 629, loss=0.012255
Epoch 630, loss=0.012233
Epoch 631, loss=0.012212
Epoch 632, loss=0.012191
Epoch 633, loss=0.012169
Epoch 634, loss=0.012148
Epoch 635, loss=0.012127
Epoch 636, loss=0.012106
Epoch 637, loss=0.012090
Epoch 638, loss=0.012080
Epoch 639, loss=0.012064
Epoch 640, loss=0.012042
Epoch 641, loss=0.012024
Epoch 642, loss=0.012014
Epoch 643, loss=0.011991
Epoch 644, loss=0.011968
Epoch 645, loss=0.011947
Epoch 646, loss=0.011928
Epoch 647, loss=0.011928
Epoch 648, loss=0.011904
Epoch 649, loss=0.011881
Epoch 650, loss=0.011859
Epoch 650, loss=0.011859
Epoch 651, loss=0.011837
Epoch 652, loss=0.011815
Epoch 653, loss=0.011794
Epoch 654, loss=0.011773
Epoch 655, loss=0.011752
Epoch 656, loss=0.011732
Epoch 657, loss=0.011711
Epoch 658, loss=0.011691
Epoch 659, loss=0.011670
Epoch 660, loss=0.011651
Epoch 661, loss=0.011631
Epoch 662, loss=0.011611
Epoch 663, loss=0.011597
Epoch 664, loss=0.011578
Epoch 665, loss=0.011559
Epoch 666, loss=0.011544
Epoch 667, loss=0.011529
Epoch 668, loss=0.011510
Epoch 669, loss=0.011490
Epoch 670, loss=0.011471
Epoch 671, loss=0.011451
```

```
Epoch 672, loss=0.011433
Epoch 673, loss=0.011414
Epoch 674, loss=0.011395
Epoch 675, loss=0.011377
Epoch 676, loss=0.011359
Epoch 677, loss=0.011340
Epoch 678, loss=0.011322
Epoch 679, loss=0.011304
Epoch 680, loss=0.011286
Epoch 681, loss=0.011267
Epoch 682, loss=0.011250
Epoch 683, loss=0.011232
Epoch 684, loss=0.011214
Epoch 685, loss=0.011197
Epoch 686, loss=0.011180
Epoch 687, loss=0.011162
Epoch 688, loss=0.011149
Epoch 689, loss=0.011132
Epoch 690, loss=0.011114
Epoch 691, loss=0.011097
Epoch 692, loss=0.011080
Epoch 693, loss=0.011062
Epoch 694, loss=0.011045
Epoch 695, loss=0.011029
Epoch 696, loss=0.011011
Epoch 697, loss=0.010994
Epoch 698, loss=0.010977
Epoch 699, loss=0.010961
Epoch 700, loss=0.010943
Epoch 700, loss=0.010943
Epoch 701, loss=0.010927
Epoch 702, loss=0.010910
Epoch 703, loss=0.010893
Epoch 704, loss=0.010877
Epoch 705, loss=0.010861
Epoch 706, loss=0.010844
Epoch 707, loss=0.010828
Epoch 708, loss=0.010812
Epoch 709, loss=0.010795
Epoch 710, loss=0.010779
Epoch 711, loss=0.010763
Epoch 712, loss=0.010747
Epoch 713, loss=0.010731
Epoch 714, loss=0.010716
Epoch 715, loss=0.010700
Epoch 716, loss=0.010684
Epoch 717, loss=0.010676
Epoch 718, loss=0.010663
Epoch 719, loss=0.010647
Epoch 720, loss=0.010632
Epoch 721, loss=0.010616
Epoch 722, loss=0.010600
Epoch 723, loss=0.010584
Epoch 724, loss=0.010569
Epoch 725, loss=0.010553
Epoch 726, loss=0.010538
Epoch 727, loss=0.010523
Epoch 728, loss=0.010510
Epoch 729, loss=0.010497
Epoch 730, loss=0.010482
Epoch 731, loss=0.010467
Epoch 732, loss=0.010451
Epoch 733, loss=0.010436
Epoch 734, loss=0.010421
Epoch 735, loss=0.010406
Epoch 736, loss=0.010391
Epoch 737, loss=0.010378
Epoch 738, loss=0.010369
Epoch 739, loss=0.010354
Epoch 740, loss=0.010338
Epoch 741, loss=0.010324
Epoch 742, loss=0.010309
Epoch 743, loss=0.010294
Epoch 744, loss=0.010280
Epoch 745, loss=0.010272
Epoch 746, loss=0.010257
```

```
Epoch 747, loss=0.010242
Epoch 748, loss=0.010227
Epoch 749, loss=0.010213
Epoch 750, loss=0.010198
Epoch 750, loss=0.010198
Epoch 751, loss=0.010183
Epoch 752, loss=0.010169
Epoch 753, loss=0.010159
Epoch 754, loss=0.010145
Epoch 755, loss=0.010131
Epoch 756, loss=0.010116
Epoch 757, loss=0.010102
Epoch 758, loss=0.010088
Epoch 759, loss=0.010080
Epoch 760, loss=0.010072
Epoch 761, loss=0.010057
Epoch 762, loss=0.010043
Epoch 763, loss=0.010028
Epoch 764, loss=0.010013
Epoch 765, loss=0.009999
Epoch 766, loss=0.009985
Epoch 767, loss=0.009971
Epoch 768, loss=0.009957
Epoch 769, loss=0.009943
Epoch 770, loss=0.009929
Epoch 771, loss=0.009915
Epoch 772, loss=0.009902
Epoch 773, loss=0.009888
Epoch 774, loss=0.009874
Epoch 775, loss=0.009861
Epoch 776, loss=0.009847
Epoch 777, loss=0.009834
Epoch 778, loss=0.009821
Epoch 779, loss=0.009807
Epoch 780, loss=0.009794
Epoch 781, loss=0.009781
Epoch 782, loss=0.009768
Epoch 783, loss=0.009755
Epoch 784, loss=0.009741
Epoch 785, loss=0.009728
Epoch 786, loss=0.009716
Epoch 787, loss=0.009702
Epoch 788, loss=0.009690
Epoch 789, loss=0.009681
Epoch 790, loss=0.009668
Epoch 791, loss=0.009654
Epoch 792, loss=0.009642
Epoch 793, loss=0.009629
Epoch 794, loss=0.009616
Epoch 795, loss=0.009603
Epoch 796, loss=0.009591
Epoch 797, loss=0.009578
Epoch 798, loss=0.009565
Epoch 799, loss=0.009553
Epoch 800, loss=0.009540
Epoch 800, loss=0.009540
Epoch 801, loss=0.009528
Epoch 802, loss=0.009516
Epoch 803, loss=0.009503
Epoch 804, loss=0.009491
Epoch 805, loss=0.009478
Epoch 806, loss=0.009468
Epoch 807, loss=0.009462
Epoch 808, loss=0.009450
Epoch 809, loss=0.009437
Epoch 810, loss=0.009425
Epoch 811, loss=0.009412
Epoch 812, loss=0.009400
Epoch 813, loss=0.009388
Epoch 814, loss=0.009375
Epoch 815, loss=0.009363
Epoch 816, loss=0.009351
Epoch 817, loss=0.009340
Epoch 818, loss=0.009331
Epoch 819, loss=0.009319
Epoch 820, loss=0.009307
```

```
Epoch 821, loss=0.009295
Epoch 822, loss=0.009289
Epoch 823, loss=0.009277
Epoch 824, loss=0.009264
Epoch 825, loss=0.009252
Epoch 826, loss=0.009240
Epoch 827, loss=0.009228
Epoch 828, loss=0.009216
Epoch 829, loss=0.009205
Epoch 830, loss=0.009193
Epoch 831, loss=0.009181
Epoch 832, loss=0.009169
Epoch 833, loss=0.009158
Epoch 834, loss=0.009146
Epoch 835, loss=0.009134
Epoch 836, loss=0.009123
Epoch 837, loss=0.009111
Epoch 838, loss=0.009100
Epoch 839, loss=0.009089
Epoch 840, loss=0.009077
Epoch 841, loss=0.009066
Epoch 842, loss=0.009055
Epoch 843, loss=0.009044
Epoch 844, loss=0.009032
Epoch 845, loss=0.009021
Epoch 846, loss=0.009011
Epoch 847, loss=0.009004
Epoch 848, loss=0.008992
Epoch 849, loss=0.008981
Epoch 850, loss=0.008970
Epoch 850, loss=0.008970
Epoch 851, loss=0.008959
Epoch 852, loss=0.008948
Epoch 853, loss=0.008937
Epoch 854, loss=0.008926
Epoch 855, loss=0.008914
Epoch 856, loss=0.008904
Epoch 857, loss=0.008895
Epoch 858, loss=0.008885
Epoch 859, loss=0.008874
Epoch 860, loss=0.008863
Epoch 861, loss=0.008852
Epoch 862, loss=0.008841
Epoch 863, loss=0.008831
Epoch 864, loss=0.008820
Epoch 865, loss=0.008809
Epoch 866, loss=0.008799
Epoch 867, loss=0.008788
Epoch 868, loss=0.008777
Epoch 869, loss=0.008767
Epoch 870, loss=0.008756
Epoch 871, loss=0.008745
Epoch 872, loss=0.008735
Epoch 873, loss=0.008725
Epoch 874, loss=0.008714
Epoch 875, loss=0.008704
Epoch 876, loss=0.008694
Epoch 877, loss=0.008683
Epoch 878, loss=0.008673
Epoch 879, loss=0.008663
Epoch 880, loss=0.008652
Epoch 881, loss=0.008642
Epoch 882, loss=0.008632
Epoch 883, loss=0.008622
Epoch 884, loss=0.008611
Epoch 885, loss=0.008601
Epoch 886, loss=0.008591
Epoch 887, loss=0.008581
Epoch 888, loss=0.008571
Epoch 889, loss=0.008564
Epoch 890, loss=0.008553
Epoch 891, loss=0.008543
Epoch 892, loss=0.008534
Epoch 893, loss=0.008523
Epoch 894, loss=0.008513
Epoch 895, loss=0.008504
```

```
Epoch 896, loss=0.008494
Epoch 897, loss=0.008483
Epoch 898, loss=0.008474
Epoch 899, loss=0.008464
Epoch 900, loss=0.008454
Epoch 900, loss=0.008454
Epoch 901, loss=0.008448
Epoch 902, loss=0.008441
Epoch 903, loss=0.008431
Epoch 904, loss=0.008422
Epoch 905, loss=0.008413
Epoch 906, loss=0.008403
Epoch 907, loss=0.008393
Epoch 908, loss=0.008383
Epoch 909, loss=0.008373
Epoch 910, loss=0.008368
Epoch 911, loss=0.008364
Epoch 912, loss=0.008358
Epoch 913, loss=0.008348
Epoch 914, loss=0.008337
Epoch 915, loss=0.008327
Epoch 916, loss=0.008317
Epoch 917, loss=0.008307
Epoch 918, loss=0.008297
Epoch 919, loss=0.008287
Epoch 920, loss=0.008277
Epoch 921, loss=0.008267
Epoch 922, loss=0.008258
Epoch 923, loss=0.008248
Epoch 924, loss=0.008238
Epoch 925, loss=0.008229
Epoch 926, loss=0.008219
Epoch 927, loss=0.008210
Epoch 928, loss=0.008201
Epoch 929, loss=0.008194
Epoch 930, loss=0.008185
Epoch 931, loss=0.008175
Epoch 932, loss=0.008166
Epoch 933, loss=0.008157
Epoch 934, loss=0.008147
Epoch 935, loss=0.008138
Epoch 936, loss=0.008129
Epoch 937, loss=0.008120
Epoch 938, loss=0.008110
Epoch 939, loss=0.008102
Epoch 940, loss=0.008092
Epoch 941, loss=0.008083
Epoch 942, loss=0.008074
Epoch 943, loss=0.008065
Epoch 944, loss=0.008056
Epoch 945, loss=0.008047
Epoch 946, loss=0.008039
Epoch 947, loss=0.008030
Epoch 948, loss=0.008021
Epoch 949, loss=0.008012
Epoch 950, loss=0.008003
Epoch 950, loss=0.008003
Epoch 951, loss=0.007994
Epoch 952, loss=0.007985
Epoch 953, loss=0.007977
Epoch 954, loss=0.007968
Epoch 955, loss=0.007959
Epoch 956, loss=0.007951
Epoch 957, loss=0.007943
Epoch 958, loss=0.007935
Epoch 959, loss=0.007927
Epoch 960, loss=0.007918
Epoch 961, loss=0.007909
Epoch 962, loss=0.007901
Epoch 963, loss=0.007892
Epoch 964, loss=0.007884
Epoch 965, loss=0.007875
Epoch 966, loss=0.007867
Epoch 967, loss=0.007858
Epoch 968, loss=0.007850
Epoch 969, loss=0.007844
```

```
Epoch 970, loss=0.007836
Epoch 971, loss=0.007828
Epoch 972, loss=0.007820
Epoch 973, loss=0.007811
Epoch 974, loss=0.007802
Epoch 975, loss=0.007794
Epoch 976, loss=0.007786
Epoch 977, loss=0.007777
Epoch 978, loss=0.007769
Epoch 979, loss=0.007761
Epoch 980, loss=0.007752
Epoch 981, loss=0.007744
Epoch 982, loss=0.007736
Epoch 983, loss=0.007728
Epoch 984, loss=0.007719
Epoch 985, loss=0.007711
Epoch 986, loss=0.007703
Epoch 987, loss=0.007695
Epoch 988, loss=0.007687
Epoch 989, loss=0.007679
Epoch 990, loss=0.007671
Epoch 991, loss=0.007662
Epoch 992, loss=0.007655
Epoch 993, loss=0.007646
Epoch 994, loss=0.007638
Epoch 995, loss=0.007630
Epoch 996, loss=0.007622
Epoch 997, loss=0.007614
Epoch 998, loss=0.007606
Epoch 999, loss=0.007601
Epoch 1000, loss=0.007593
Epoch 1000, loss=0.007593
```

In [ ]:

```
# Test your predictions. The predictions must match the labels
print(net.predict(X))
print(y)
```

```
[2 1 0 1 2 0 0 2 0 0]
[2 1 0 1 2 0 0 2 0 0]

[2 1 0 1 2 0 0 2 0 0]
```

In [ ]:

```
# You should be able to achieve a training loss of less than 0.02 (10%)
print("Final training loss", epoch_loss[-1])
```

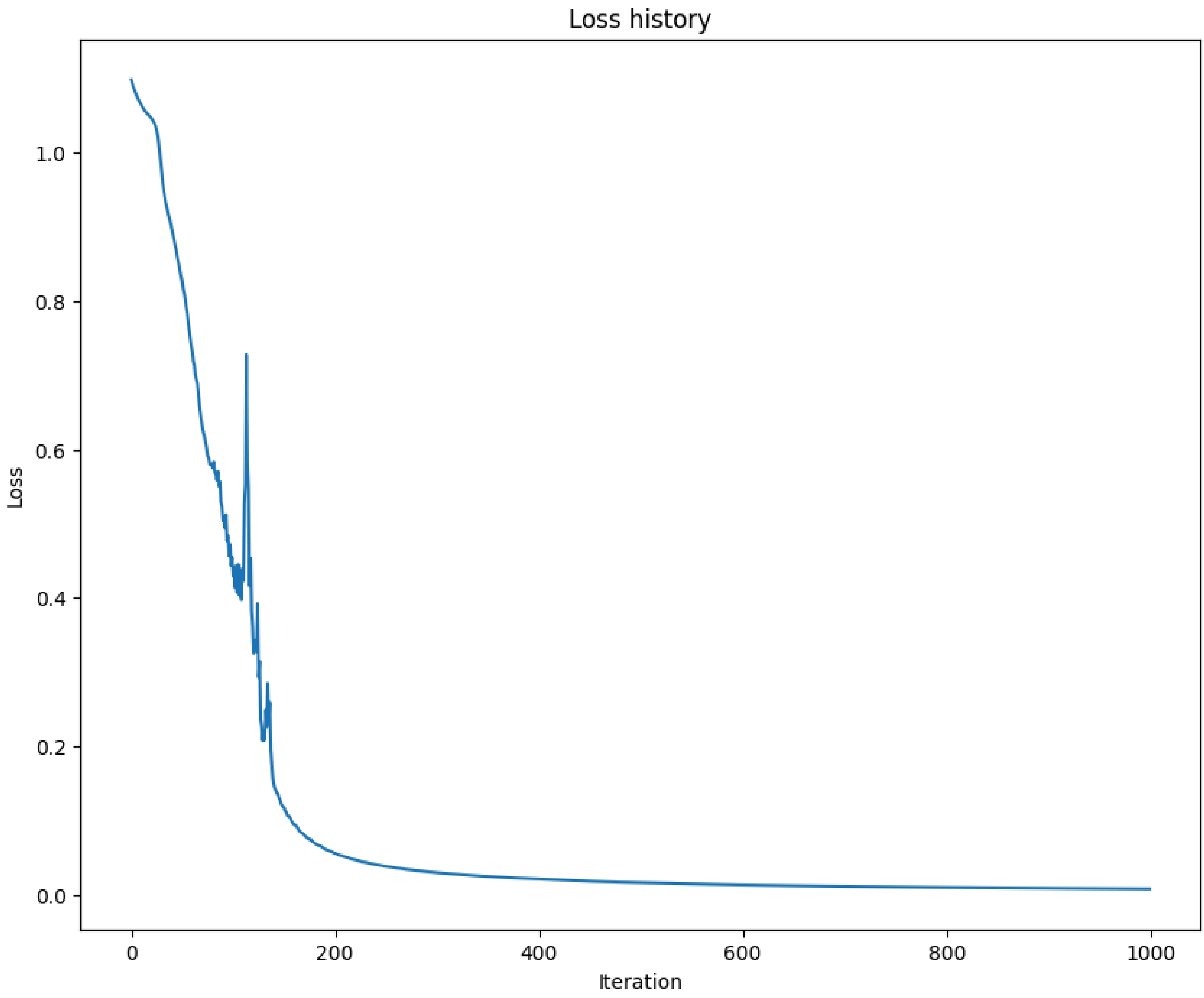
```
Final training loss 0.007593419801731252
```

In [ ]:

```
# Plot the training loss curve. The loss in the curve should be decreasing (20%)
plt.plot(epoch_loss)
plt.title("Loss history")
plt.xlabel("Iteration")
plt.ylabel("Loss")
```

Out[ ]:

```
Text(0, 0.5, 'Loss')
```


Loss history
Loss
Iteration
0.0
0.2
0.4
0.6
0.8
1.0
0
200
400
600
800
1000