```
!pip install pennylane
```

```
Requirement already satisfied: pennylane in /usr/local/lib/python3.10/dist-packages (0.33.1)
Requirement already satisfied: numpy in /usr/local/lib/python3.10/dist-packages (from pennylane) (1.23.5)
Requirement already satisfied: scipy in /usr/local/lib/python3.10/dist-packages (from pennylane) (1.11.4)
Requirement already satisfied: networkx in /usr/local/lib/python3.10/dist-packages (from pennylane) (3.2.1)
Requirement already satisfied: rustworkx in /usr/local/lib/python3.10/dist-packages (from pennylane) (0.13.2)
Requirement already satisfied: autograd in /usr/local/lib/python3.10/dist-packages (from pennylane) (1.6.2)
Requirement already satisfied: toml in /usr/local/lib/python3.10/dist-packages (from pennylane) (0.10.2)
Requirement already satisfied: appdirs in /usr/local/lib/python3.10/dist-packages (from pennylane) (1.4.4)
Requirement already satisfied: semantic-version>=2.7 in /usr/local/lib/python3.10/dist-packages (from pennylane) (2.10.0)
Requirement already satisfied: autoray>=0.6.1 in /usr/local/lib/python3.10/dist-packages (from pennylane) (0.6.7)
Requirement already satisfied: cachetools in /usr/local/lib/python3.10/dist-packages (from pennylane) (5.3.2)
Requirement already satisfied: pennylane-lightning>=0.33 in /usr/local/lib/python3.10/dist-packages (from pennylane) (0.33.1)
Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-packages (from pennylane) (2.31.0)
Requirement already satisfied: typing-extensions in /usr/local/lib/python3.10/dist-packages (from pennylane) (4.5.0)
Requirement already satisfied: future>=0.15.2 in /usr/local/lib/python3.10/dist-packages (from autograd->pennylane) (0.18.3)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests->pennylane) (3.3.2)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests->pennylane) (3.6)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests->pennylane) (2.0.7)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests->pennylane) (2023.11.17)
```

```
from sklearn import datasets

iris = datasets.load_iris()
X = np.array(iris['data'], requires_grad=True)
y = np.array(iris['target'], requires_grad=True)
X, y
```

```
                0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
                0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
                1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
                1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
                2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
                2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2], requires_grad=True))
```

```
import pennylane as qml
import torch
from torch.autograd import Variable
from pennylane import numpy as np

num_wires = 4
num_layers = 2

dev = qml.device("default.qubit", wires=num_wires)

coeffs = [1]
obs = [qml.PauliZ(0) @ qml.PauliZ(1)]
H = qml.Hamiltonian(coeffs, obs)

#START HERE
'''
@qml.qnode(dev, interface='autograd')
def circuit(params, x=x):
  qml.RX(params[0], wires=0)
  qml.RZ(params[1], wires=0)
  for i in range(1, num_wires):
    qml.CNOT([i-1, i])
    qml.RX(params[0], wires=i)
    qml.RZ(params[1], wires=i)
  return qml.expval(qml.PauliZ(0))

def circuit_0_1(params):
  print('check i', circuit(params))
  return (circuit(params) + 1)/2
'''

@qml.qnode(dev, interface='autograd')
def circuit(params, x):
  x /= np.linalg.norm(x)
  #qml.AmplitudeEmbedding(x, wires=list(range(num_wires)))
  qml.AngleEmbedding(features=x, wires=range(num_wires), rotation='Y')
  qml.BasicEntanglerLayers(weights=params, wires=range(num_wires))

  return qml.expval(H)

  #print('state', qml.state())
  ##print('full_circuit_1', circuit_0_1(params))
  #return circuit_0_1(params)

#def binary_cost(params):
#  return circuit_0_1

def full_circuit(params, x):
  ###print('circuit result', circuit(params, x))
  return (np.array(circuit(params, x)).mean() + 1)/2

def square_loss(labels, predictions):
    loss = 0
    ##print(labels, predictions)
    for true, pred in zip(labels, predictions):
        ##print(true, pred)
        loss = loss + (true - pred) ** 2

    loss = loss / len(labels)
    return loss

def cross_loss(labels, predictions):
    loss = 0
    for true, pred in zip(labels, predictions):
      #print(true*np.log(pred + 1))
      loss = loss + true*np.log(pred) + (1-true)*np.log(1 - pred)
    loss = - loss/len(labels)
    return loss

def accuracy(labels, predictions):
    #loss = (labels - predictions).sum()
```

```
        loss1 = 0
        loss2 = 0
        ##print('accuracy', labels, predictions)
        for true, pred in zip(labels, predictions):
            if true - pred < 0.5:
                loss1 = loss1 + 1
            if true - pred > 0.5:
                loss2 = loss2 + 1
        acc1 = loss1 / len(labels)
        acc2 = loss2 / len(labels)
        return max(acc1, acc2)

    def cost(params, X_batch, y_batch):
        #predictions = full_circuit(params, X_batch)
        ##print('X_batch.shape',X_batch.shape)
        data_indices = [np.random.randint(0, len(X)) for _ in range(1)]
        predictions = [full_circuit(params, X_batch[index]) for index in data_indices]
        ##print('len(predictions)',len(predictions))
        #predictions = full_circuit(params, X_batch.flatten())
        ##print('Cost',y_batch, predictions)
        return cross_loss(y_batch[data_indices], predictions)

    '''
    def circuit_probs(params):
      qml.RX(params[0], wires=0)
      qml.RZ(params[1], wires=0)
      return qml.probs(wires=0)

    def cost_probs(params):
      return circuit_probs - data'''

    optimizer = qml.AdamOptimizer(stepsize=0.01)
    params = np.array([[0.3, 0.5, 0.3, 0.5], [0.5, 0.5, 0.3, 0.5]], requires_grad=True)

    #X = np.array([[1., 1., 0., 0.], [0., 0., 1., 1.], [1., 0., 1., 0.], [0., 1., 0., 1.]], requires_grad=True)
    #y = np.array([0, 0, 1, 1], requires_grad=True)

    losses = []

    for it in range(100):
      #data_index = np.random.randint(0, len(X))
      #X_batch = X[data_index]
      #y_batch = y[data_index]

      X_batch = X
      y_batch = y
      params, loss = optimizer.step_and_cost(cost, params, X_batch=X_batch, y_batch=y_batch)
      print('step', it, params, loss)
      losses.append(loss)

      predictions = [full_circuit(params, x) for x in X]
      print('step', it, 'params', params, 'loss', loss, 'predictions', predictions)
      #predictions = full_circuit(params, X.flatten())
      acc = accuracy(y, predictions)

      print(
          "Iter: {:5d} | Cost: {:0.7f} | Accuracy: {:0.7f} ".format(
              it + 1, cost(params, X, y), acc
          )
      )

    '''state_preparation(x)

    losses = []
    for i in range(400):
      params, loss = optimizer.step_and_cost(circuit, params)
      losses.append(loss)
      print(loss)'''


    #END HERE
```

```
step 0 [[0.3        0.49       0.29000003 0.5       ]
 [0.49       0.5        0.29000003 0.49000001]] 0.35658584877518806
step 0 params [[0.3        0.49       0.29000003 0.5       ]
 [0.49       0.5        0.29000003 0.49000001]] loss 0.35658584877518806 predictions [tensor(0.70546527, requires_grad=True), tensor(0.6
Iter:     1 | Cost: 1.1636282 | Accuracy: 1.0000000
step 1 [[0.3        0.49425228 0.29451558 0.5       ]
 [0.49425941 0.5        0.29451558 0.49434707]] 1.222358361430949
step 1 params [[0.3        0.49425228 0.29451558 0.5       ]
 [0.49425941 0.5        0.29451558 0.49434707]] loss 1.222358361430949 predictions [tensor(0.70313793, requires_grad=True), tensor(0.684
Iter:     2 | Cost: 1.1545816 | Accuracy: 1.0000000
step 2 [[0.3        0.49502091 0.2955271  0.5       ]
 [0.49503476 0.5        0.2955271  0.49520382]] 0.35220220859754986
step 2 params [[0.3        0.49502091 0.2955271  0.5       ]
 [0.49503476 0.5        0.2955271  0.49520382]] loss 0.35220220859754986 predictions [tensor(0.7027022, requires_grad=True), tensor(0.68
Iter:     3 | Cost: 1.2130210 | Accuracy: 1.0000000
step 3 [[0.3        0.49926405 0.29987381 0.5       ]
 [0.49928055 0.5        0.29987381 0.49948335]] 1.2130209614783634
step 3 params [[0.3        0.49926405 0.29987381 0.5       ]
 [0.49928055 0.5        0.29987381 0.49948335]] loss 1.2130209614783634 predictions [tensor(0.70040824, requires_grad=True), tensor(0.68
Iter:     4 | Cost: 0.3835079 | Accuracy: 1.0000000
step 4 [[0.3        0.50209257 0.29866599 0.5       ]
 [0.50133781 0.5        0.29866599 0.49959834]] 0.3835079052488763
step 4 params [[0.3        0.50209257 0.29866599 0.5       ]
 [0.50133781 0.5        0.29866599 0.49959834]] loss 0.3835079052488763 predictions [tensor(0.69945193, requires_grad=True), tensor(0.68
Iter:     5 | Cost: 0.3842324 | Accuracy: 1.0000000
step 5 [[0.3        0.5030599  0.29670457 0.5       ]
 [0.5017019  0.5        0.29670457 0.49853771]] 0.3574582145445693
step 5 params [[0.3        0.5030599  0.29670457 0.5       ]
 [0.5017019  0.5        0.29670457 0.49853771]] loss 0.3574582145445693 predictions [tensor(0.6993523, requires_grad=True), tensor(0.681
Iter:     6 | Cost: 1.2018161 | Accuracy: 1.0000000
step 6 [[0.3        0.50643901 0.2970607  0.5       ]
 [0.50458754 0.5        0.2970607  0.50003496]] 1.201816121062332
step 6 params [[0.3        0.50643901 0.2970607  0.5       ]
 [0.50458754 0.5        0.2970607  0.50003496]] loss 1.201816121062332 predictions [tensor(0.69792117, requires_grad=True), tensor(0.680
Iter:     7 | Cost: 1.1970673 | Accuracy: 1.0000000
step 7 [[0.3        0.50824707 0.29658629 0.5       ]
 [0.50599255 0.5        0.29658629 0.50037701]] 0.3596491252803576
step 7 params [[0.3        0.50824707 0.29658629 0.5       ]
 [0.50599255 0.5        0.29658629 0.50037701]] loss 0.3596491252803576 predictions [tensor(0.69724851, requires_grad=True), tensor(0.67
Iter:     8 | Cost: 1.1383762 | Accuracy: 1.0000000
step 8 [[0.3        0.51189984 0.29784311 0.5       ]
 [0.50930072 0.5        0.29784311 0.50263207]] 1.1948429721867726
step 8 params [[0.3        0.51189984 0.29784311 0.5       ]
 [0.50930072 0.5        0.29784311 0.50263207]] loss 1.1948429721867726 predictions [tensor(0.69556949, requires_grad=True), tensor(0.67
Iter:     9 | Cost: 1.1893124 | Accuracy: 1.0000000
step 9 [[0.3        0.51607022 0.30224785 0.5       ]
 [0.51383268 0.5        0.30224785 0.50731878]] 1.133274713878167
step 9 params [[0.3        0.51607022 0.30224785 0.5       ]
 [0.51383268 0.5        0.30224785 0.50731878]] loss 1.133274713878167 predictions [tensor(0.69321225, requires_grad=True), tensor(0.675
Iter:    10 | Cost: 0.3664191 | Accuracy: 1.0000000
step 10 [[0.3        0.51934116 0.30431257 0.5       ]
 [0.517002   0.5        0.30431257 0.50988277]] 0.39285571931707836
step 10 params [[0.3        0.51934116 0.30431257 0.5       ]
 [0.517002   0.5        0.30431257 0.50988277]] loss 0.39285571931707836 predictions [tensor(0.69159334, requires_grad=True), tensor(0.6
Iter:    11 | Cost: 0.3687572 | Accuracy: 1.0000000
step 11 [[0.3        0.52181908 0.30451967 0.5       ]
 [0.51898748 0.5        0.30451967 0.51070788]] 0.39545794517087896
step 11 params [[0.3        0.52181908 0.30451967 0.5       ]
 [0.51898748 0.5        0.30451967 0.51070788]] loss 0.39545794517087896 predictions [tensor(0.69059227, requires_grad=True), tensor(0.6
Iter:    12 | Cost: 1.1164353 | Accuracy: 1.0000000
step 12 [[0.3        0.52493822 0.30730797 0.5       ]
 [0.52227721 0.5        0.30730797 0.51378367]] 1.116435255939546
step 12 params [[0.3        0.52493822 0.30730797 0.5       ]
 [0.52227721 0.5        0.30730797 0.51378367]] loss 1.116435255939546 predictions [tensor(0.68891912, requires_grad=True), tensor(0.670
Iter:    13 | Cost: 1.1677023 | Accuracy: 1.0000000
step 13 [[0.3        0.5294228  0.31042029 0.5       ]
 [0.5267486  0.5        0.31042029 0.51753321]] 1.1677023241536197
step 13 params [[0.3        0.5294228  0.31042029 0.5       ]
 [0.5267486  0.5        0.31042029 0.51753321]] loss 1.1677023241536197 predictions [tensor(0.68666022, requires_grad=True), tensor(0.66
Iter:    14 | Cost: 1.1028681 | Accuracy: 1.0000000
step 14 [[0.3        0.53306029 0.31190753 0.5       ]
 [0.53002361 0.5        0.31190753 0.51969072]] 0.40334395168050297
step 14 params [[0.3        0.53306029 0.31190753 0.5       ]
 [0.53002361 0.5        0.31190753 0.51969072]] loss 0.40334395168050297 predictions [tensor(0.68500712, requires_grad=True), tensor(0.6
Iter:    15 | Cost: 0.4057804 | Accuracy: 1.0000000
step 15 [[0.3        0.53713813 0.3153017  0.5       ]
 [0.53427389 0.5        0.3153017  0.52352498]] 1.097981914270131
step 15 params [[0.3        0.53713813 0.3153017  0.5       ]
 [0.53427389 0.5        0.3153017  0.52352498]] loss 1.097981914270131 predictions [tensor(0.6828795, requires_grad=True), tensor(0.6639
Iter:    16 | Cost: 1.0906175 | Accuracy: 1.0000000
step 16 [[0.3        0.54233924 0.31881362 0.5       ]
 [0.53946072 0.5        0.31881362 0.52781256]] 1.1484734555169875
step 16 params [[0.3        0.54233924 0.31881362 0.5       ]
 [0.53946072 0.5        0.31881362 0.52781256]] loss 1.1484734555169875 predictions [tensor(0.68029748, requires_grad=True), tensor(0.66
```

```
Iter:    17 | Cost: 1.1403643 | Accuracy: 1.0000000
step 17 [[0.3        0.54776732 0.32369767 0.5       ]
 [0.54529576 0.5        0.32369767 0.53327865]] 1.0821856873921665
step 17 params [[0.3        0.54776732 0.32369767 0.5       ]
 [0.54529576 0.5        0.32369767 0.53327865]] loss 1.0821856873921665 predictions [tensor(0.67743037, requires_grad=True), tensor(0.65
Iter:    18 | Cost: 0.4189450 | Accuracy: 1.0000000
step 18 [[0.3        0.55182201 0.32798854 0.5       ]
 [0.54985989 0.5        0.32798854 0.53788184]] 0.38944850666298636
step 18 params [[0.3        0.55182201 0.32798854 0.5       ]
 [0.54985989 0.5        0.32798854 0.53788184]] loss 0.38944850666298636 predictions [tensor(0.67521308, requires_grad=True), tensor(0.6
Iter:    19 | Cost: 1.1245859 | Accuracy: 1.0000000
step 19 [[0.3        0.55464666 0.33175513 0.5       ]
 [0.55327908 0.5        0.33175513 0.54170587]] 0.39272696158189135
step 19 params [[0.3        0.55464666 0.33175513 0.5       ]
 [0.55327908 0.5        0.33175513 0.54170587]] loss 0.39272696158189135 predictions [tensor(0.67357266, requires_grad=True), tensor(0.6
Iter:    20 | Cost: 1.1195479 | Accuracy: 1.0000000
step 20 [[0.3        0.55683179 0.33410342 0.5       ]
 [0.55570313 0.5        0.33410342 0.54415049]] 0.4265032552863242
step 20 params [[0.3        0.55683179 0.33410342 0.5       ]
 [0.55570313 0.5        0.33410342 0.54415049]] loss 0.4265032552863242 predictions [tensor(0.67240409, requires_grad=True), tensor(0.65
Iter:    21 | Cost: 1.0536050 | Accuracy: 1.0000000
step 21 [[0.3        0.55952032 0.33788498 0.5       ]
 [0.55906672 0.5        0.33788498 0.54796672]] 1.0536050058893218
step 21 params [[0.3        0.55952032 0.33788498 0.5       ]
 [0.55906672 0.5        0.33788498 0.54796672]] loss 1.0536050058893218 predictions [tensor(0.67082725, requires_grad=True), tensor(0.64
Iter:    22 | Cost: 1.0474575 | Accuracy: 1.0000000
step 22 [[0.3        0.56265742 0.34281541 0.5       ]
 [0.56323738 0.5        0.34281541 0.5528935 ]] 1.047457509096443
step 22 params [[0.3        0.56265742 0.34281541 0.5       ]
 [0.56323738 0.5        0.34281541 0.5528935 ]] loss 1.047457509096443 predictions [tensor(0.66891688, requires_grad=True), tensor(0.646
Iter:    23 | Cost: 1.0397811 | Accuracy: 1.0000000
step 23 [[0.3        0.56698621 0.34748934 0.5       ]
 [0.56824423 0.5        0.34748934 0.55798536]] 1.1053858116264026
step 23 params [[0.3        0.56698621 0.34748934 0.5       ]
 [0.56824423 0.5        0.34748934 0.55798536]] loss 1.1053858116264026 predictions [tensor(0.66660604, requires_grad=True), tensor(0.64
Iter:    24 | Cost: 1.0314993 | Accuracy: 1.0000000
step 24 [[0.3        0.57229851 0.35189744 0.5       ]
 [0.5739613  0.5        0.35189744 0.5632154 ]] 1.0984304393085051
step 24 params [[0.3        0.57229851 0.35189744 0.5       ]
 [0.5739613  0.5        0.35189744 0.5632154 ]] loss 1.0984304393085051 predictions [tensor(0.663968, requires_grad=True), tensor(0.6404
Iter:    25 | Cost: 0.4456534 | Accuracy: 1.0000000
step 25 [[0.3        0.57678513 0.35494736 0.5       ]
 [0.57852471 0.5        0.35494736 0.56702524]] 0.4456533954501497
step 25 params [[0.3        0.57678513 0.35494736 0.5       ]
 [0.57852471 0.5        0.35494736 0.56702524]] loss 0.4456533954501497 predictions [tensor(0.66185638, requires_grad=True), tensor(0.63
Iter:    26 | Cost: 1.0162302 | Accuracy: 1.0000000
step 26 [[0.3        0.58004413 0.35769691 0.5       ]
 [0.58199823 0.5        0.35769691 0.57020029]] 0.41270669916309977
step 26 params [[0.3        0.58004413 0.35769691 0.5       ]
 [0.58199823 0.5        0.35769691 0.57020029]] loss 0.41270669916309977 predictions [tensor(0.66027331, requires_grad=True), tensor(0.6
Iter:    27 | Cost: 1.0110406 | Accuracy: 1.0000000
step 27 [[0.3        0.58361867 0.36163646 0.5       ]
 [0.5861823  0.5        0.36163646 0.57450446]] 1.0110406003999757
step 27 params [[0.3        0.58361867 0.36163646 0.5       ]
 [0.5861823  0.5        0.36163646 0.57450446]] loss 1.0110406003999757 predictions [tensor(0.65842511, requires_grad=True), tensor(0.63
Iter:    28 | Cost: 0.4560785 | Accuracy: 1.0000000
step 28 [[0.3        0.58747489 0.36656368 0.5       ]
 [0.5909865  0.5        0.36656368 0.57975347]] 1.0044776145277858
step 28 params [[0.3        0.58747489 0.36656368 0.5       ]
 [0.5909865  0.5        0.36656368 0.57975347]] loss 1.0044776145277858 predictions [tensor(0.65636316, requires_grad=True), tensor(0.63
Iter:    29 | Cost: 1.0681699 | Accuracy: 1.0000000
step 29 [[0.3        0.59235043 0.37104173 0.5       ]
 [0.59648274 0.5        0.37104173 0.58503765]] 1.068169869758839
step 29 params [[0.3        0.59235043 0.37104173 0.5       ]
 [0.59648274 0.5        0.37104173 0.58503765]] loss 1.068169869758839 predictions [tensor(0.65398087, requires_grad=True), tensor(0.628
Iter:    30 | Cost: 0.4652093 | Accuracy: 1.0000000
step 30 [[0.3        0.59644964 0.37423707 0.5       ]
 [0.60086265 0.5        0.37423707 0.58894355]] 0.4652092629119201
step 30 params [[0.3        0.59644964 0.37423707 0.5       ]
 [0.60086265 0.5        0.37423707 0.58894355]] loss 0.4652092629119201 predictions [tensor(0.65207459, requires_grad=True), tensor(0.62
Iter:    31 | Cost: 1.0557672 | Accuracy: 1.0000000
step 31 [[0.3        0.59935456 0.37720464 0.5       ]
 [0.604181   0.5        0.37720464 0.59224748]] 0.42759632056442304
step 31 params [[0.3        0.59935456 0.37720464 0.5       ]
 [0.604181   0.5        0.37720464 0.59224748]] loss 0.42759632056442304 predictions [tensor(0.65067063, requires_grad=True), tensor(0.6
Iter:    32 | Cost: 1.0517400 | Accuracy: 1.0000000
step 32 [[0.3        0.60256508 0.38119418 0.5       ]
 [0.60816825 0.5        0.38119418 0.59657241]] 0.9780682860578241
step 32 params [[0.3        0.60256508 0.38119418 0.5       ]
 [0.60816825 0.5        0.38119418 0.59657241]] loss 0.9780682860578241 predictions [tensor(0.64904576, requires_grad=True), tensor(0.62
Iter:    33 | Cost: 0.9721336 | Accuracy: 1.0000000
step 33 [[0.3        0.60684954 0.3846887  0.5       ]
 [0.61291285 0.5        0.3846887  0.60096241]] 1.0470994349097267
step 33 params [[0.3        0.60684954 0.3846887  0.5       ]
 [0.61291285 0.5        0.3846887  0.60096241]] loss 1.0470994349097267 predictions [tensor(0.6470646, requires_grad=True), tensor(0.619
```

```
Iter:    34 | Cost: 0.4353091 | Accuracy: 1.0000000
step 34 [[0.3        0.61205106 0.38769397 0.5       ]
 [0.61831767 0.5        0.38769397 0.60540076]] 1.0414702359092833
step 34 params [[0.3        0.61205106 0.38769397 0.5       ]
 [0.61831767 0.5        0.38769397 0.60540076]] loss 1.0414702359092833 predictions [tensor(0.644778, requires_grad=True), tensor(0.6168
Iter:    35 | Cost: 0.9592893 | Accuracy: 1.0000000
step 35 [[0.3        0.61803813 0.39021088 0.5       ]
 [0.62429864 0.5        0.39021088 0.60987054]] 1.0350123200957273
step 35 params [[0.3        0.61803813 0.39021088 0.5       ]
 [0.62429864 0.5        0.39021088 0.60987054]] loss 1.0350123200957273 predictions [tensor(0.64222702, requires_grad=True), tensor(0.61
Iter:    36 | Cost: 0.9525546 | Accuracy: 1.0000000
step 36 [[0.3        0.62469968 0.39223668 0.5       ]
 [0.63078276 0.5        0.39223668 0.61435497]] 1.0278566409103849
step 36 params [[0.3        0.62469968 0.39223668 0.5       ]
 [0.63078276 0.5        0.39223668 0.61435497]] loss 1.0278566409103849 predictions [tensor(0.63944489, requires_grad=True), tensor(0.61
Iter:    37 | Cost: 0.4916875 | Accuracy: 1.0000000
step 37 [[0.3        0.62996325 0.39426328 0.5       ]
 [0.63602367 0.5        0.39426328 0.61821851]] 0.4471548414654349
step 37 params [[0.3        0.62996325 0.39426328 0.5       ]
 [0.63602367 0.5        0.39426328 0.61821851]] loss 0.4471548414654349 predictions [tensor(0.63723371, requires_grad=True), tensor(0.60
Iter:    38 | Cost: 1.0139965 | Accuracy: 1.0000000
step 38 [[0.3        0.63444442 0.39531444 0.5       ]
 [0.64018658 0.5        0.39531444 0.62081132]] 0.49529393518953696
step 38 params [[0.3        0.63444442 0.39531444 0.5       ]
 [0.64018658 0.5        0.39531444 0.62081132]] loss 0.49529393518953696 predictions [tensor(0.63542729, requires_grad=True), tensor(0.6
Iter:    39 | Cost: 1.0090293 | Accuracy: 1.0000000
step 39 [[0.3        0.63820918 0.39549517 0.5       ]
 [0.64336551 0.5        0.39549517 0.62225779]] 0.4979541924916149
step 39 params [[0.3        0.63820918 0.39549517 0.5       ]
 [0.64336551 0.5        0.39549517 0.62225779]] loss 0.4979541924916149 predictions [tensor(0.63398097, requires_grad=True), tensor(0.60
Iter:    40 | Cost: 0.4997683 | Accuracy: 1.0000000
step 40 [[0.3        0.64290237 0.3952515  0.5       ]
 [0.64734348 0.5        0.3952515  0.62394132]] 1.005069964965299
step 40 params [[0.3        0.64290237 0.3952515  0.5       ]
 [0.64734348 0.5        0.3952515  0.62394132]] loss 1.005069964965299 predictions [tensor(0.63216164, requires_grad=True), tensor(0.605
Iter:    41 | Cost: 1.0001117 | Accuracy: 1.0000000
step 41 [[0.3        0.64840508 0.39459337 0.5       ]
 [0.6520302  0.5        0.39459337 0.62583321]] 1.0001116769309575
step 41 params [[0.3        0.64840508 0.39459337 0.5       ]
 [0.6520302  0.5        0.39459337 0.62583321]] loss 1.0001116769309575 predictions [tensor(0.63000846, requires_grad=True), tensor(0.60
Iter:    42 | Cost: 0.9942751 | Accuracy: 1.0000000
step 42 [[0.3        0.65259708 0.39426937 0.5       ]
 [0.65561511 0.5        0.39426937 0.62735176]] 0.46202202536240206
step 42 params [[0.3        0.65259708 0.39426937 0.5       ]
 [0.65561511 0.5        0.39426937 0.62735176]] loss 0.46202202536240206 predictions [tensor(0.62836539, requires_grad=True), tensor(0.6
Iter:    43 | Cost: 0.5062854 | Accuracy: 1.0000000
step 43 [[0.3        0.65559588 0.39426767 0.5       ]
 [0.65819292 0.5        0.39426767 0.62853196]] 0.4646334429914637
step 43 params [[0.3        0.65559588 0.39426767 0.5       ]
 [0.65819292 0.5        0.39426767 0.62853196]] loss 0.4646334429914637 predictions [tensor(0.627192, requires_grad=True), tensor(0.6018
Iter:    44 | Cost: 0.4665026 | Accuracy: 1.0000000
step 44 [[0.3        0.65750961 0.39457378 0.5       ]
 [0.65985107 0.5        0.39457378 0.62940481]] 0.466502565059358
step 44 params [[0.3        0.65750961 0.39457378 0.5       ]
 [0.65985107 0.5        0.39457378 0.62940481]] loss 0.466502565059358 predictions [tensor(0.62644929, requires_grad=True), tensor(0.601
Iter:    45 | Cost: 0.9847015 | Accuracy: 1.0000000
step 45 [[0.3        0.66053695 0.39431873 0.5       ]
 [0.66246937 0.5        0.39431873 0.63053832]] 0.9847015021993034
step 45 params [[0.3        0.66053695 0.39431873 0.5       ]
 [0.66246937 0.5        0.39431873 0.63053832]] loss 0.9847015021993034 predictions [tensor(0.62525182, requires_grad=True), tensor(0.60
Iter:    46 | Cost: 0.4696008 | Accuracy: 1.0000000
step 46 [[0.3        0.66454207 0.39353402 0.5       ]
 [0.665942   0.5        0.39353402 0.63190511]] 0.9815010000746512
step 46 params [[0.3        0.66454207 0.39353402 0.5       ]
 [0.665942   0.5        0.39353402 0.63190511]] loss 0.9815010000746512 predictions [tensor(0.62364754, requires_grad=True), tensor(0.59
Iter:    47 | Cost: 0.9146856 | Accuracy: 1.0000000
step 47 [[0.3        0.66940619 0.39224473 0.5       ]
 [0.67017495 0.5        0.39224473 0.63347838]] 0.9772291706423565
step 47 params [[0.3        0.66940619 0.39224473 0.5       ]
 [0.67017495 0.5        0.39224473 0.63347838]] loss 0.9772291706423565 predictions [tensor(0.62167709, requires_grad=True), tensor(0.59
Iter:    48 | Cost: 0.9116014 | Accuracy: 1.0000000
step 48 [[0.3        0.67301386 0.39144089 0.5       ]
 [0.67332435 0.5        0.39144089 0.63471512]] 0.4753344700690587
step 48 params [[0.3        0.67301386 0.39144089 0.5       ]
 [0.67332435 0.5        0.39144089 0.63471512]] loss 0.4753344700690587 predictions [tensor(0.62021419, requires_grad=True), tensor(0.59
Iter:    49 | Cost: 0.9092485 | Accuracy: 1.0000000
step 49 [[0.3        0.67674695 0.39176455 0.5       ]
 [0.67707158 0.5        0.39176455 0.63726503]] 0.909248461162876
step 49 params [[0.3        0.67674695 0.39176455 0.5       ]
 [0.67707158 0.5        0.39176455 0.63726503]] loss 0.909248461162876 predictions [tensor(0.61861898, requires_grad=True), tensor(0.595
Iter:    50 | Cost: 0.9639564 | Accuracy: 1.0000000
step 50 [[0.3        0.68059348 0.39307039 0.5       ]
 [0.68135525 0.5        0.39307039 0.64095431]] 0.9058404460167373
step 50 params [[0.3        0.68059348 0.39307039 0.5       ]
 [0.68135525 0.5        0.39307039 0.64095431]] loss 0.9058404460167373 predictions [tensor(0.61691467, requires_grad=True), tensor(0.59
```

```
Iter:    51 | Cost: 0.9015418 | Accuracy: 1.0000000
step 51 [[0.3        0.68532086 0.39357393 0.5       ]
 [0.68631021 0.5        0.39357393 0.64459692]] 0.9594975106294343
step 51 params [[0.3        0.68532086 0.39357393 0.5       ]
 [0.68631021 0.5        0.39357393 0.64459692]] loss 0.9594975106294343 predictions [tensor(0.61485781, requires_grad=True), tensor(0.59
Iter:    52 | Cost: 0.9541427 | Accuracy: 1.0000000
step 52 [[0.3        0.68878119 0.39448058 0.5       ]
 [0.69009924 0.5        0.39448058 0.64774058]] 0.4863642344293955
step 52 params [[0.3        0.68878119 0.39448058 0.5       ]
 [0.69009924 0.5        0.39448058 0.64774058]] loss 0.4863642344293955 predictions [tensor(0.61334285, requires_grad=True), tensor(0.59
Iter:    53 | Cost: 0.4888312 | Accuracy: 1.0000000
step 53 [[0.3        0.69314856 0.39455452 0.5       ]
 [0.69460324 0.5        0.39455452 0.65086836]] 0.9502168893358355
step 53 params [[0.3        0.69314856 0.39455452 0.5       ]
 [0.69460324 0.5        0.39455452 0.65086836]] loss 0.9502168893358355 predictions [tensor(0.61145135, requires_grad=True), tensor(0.58
Iter:    54 | Cost: 0.8894927 | Accuracy: 1.0000000
step 54 [[0.3        0.69754502 0.39551393 0.5       ]
 [0.69954935 0.5        0.39551393 0.65506284]] 0.8894927294586289
step 54 params [[0.3        0.69754502 0.39551393 0.5       ]
 [0.69954935 0.5        0.39551393 0.65506284]] loss 0.8894927294586289 predictions [tensor(0.60949941, requires_grad=True), tensor(0.58
Iter:    55 | Cost: 0.8848593 | Accuracy: 1.0000000
step 55 [[0.3        0.70196424 0.39723408 0.5       ]
 [0.70488989 0.5        0.39723408 0.66018213]] 0.88485928655843
step 55 params [[0.3        0.70196424 0.39723408 0.5       ]
 [0.70488989 0.5        0.39723408 0.66018213]] loss 0.88485928655843 predictions [tensor(0.60750532, requires_grad=True), tensor(0.5850
Iter:    56 | Cost: 0.4983944 | Accuracy: 1.0000000
step 56 [[0.3        0.70568451 0.39821587 0.5       ]
 [0.70910894 0.5        0.39821587 0.66381759]] 0.5360377429271953
step 56 params [[0.3        0.70568451 0.39821587 0.5       ]
 [0.70910894 0.5        0.39821587 0.66381759]] loss 0.5360377429271953 predictions [tensor(0.60587826, requires_grad=True), tensor(0.58
Iter:    57 | Cost: 0.9310954 | Accuracy: 1.0000000
step 57 [[0.3        0.70948164 0.3998868  0.5       ]
 [0.71378704 0.5        0.3998868  0.66840701]] 0.8757242534764772
step 57 params [[0.3        0.70948164 0.3998868  0.5       ]
 [0.71378704 0.5        0.3998868  0.66840701]] loss 0.8757242534764772 predictions [tensor(0.60418424, requires_grad=True), tensor(0.58
Iter:    58 | Cost: 0.5038761 | Accuracy: 1.0000000
step 58 [[0.3        0.71206311 0.40200585 0.5       ]
 [0.71730629 0.5        0.40200585 0.6724526 ]] 0.5038760878274203
step 58 params [[0.3        0.71206311 0.40200585 0.5       ]
 [0.71730629 0.5        0.40200585 0.6724526 ]] loss 0.5038760878274203 predictions [tensor(0.60303126, requires_grad=True), tensor(0.58
Iter:    59 | Cost: 0.5057863 | Accuracy: 1.0000000
step 59 [[0.3        0.71354564 0.40455534 0.5       ]
 [0.71977443 0.5        0.40455534 0.67600782]] 0.5057862503070278
step 59 params [[0.3        0.71354564 0.40455534 0.5       ]
 [0.71977443 0.5        0.40455534 0.67600782]] loss 0.5057862503070278 predictions [tensor(0.60236811, requires_grad=True), tensor(0.57
Iter:    60 | Cost: 0.5068865 | Accuracy: 1.0000000
step 60 [[0.3        0.71614224 0.40587066 0.5       ]
 [0.72306997 0.5        0.40587066 0.67942529]] 0.9222286055464863
step 60 params [[0.3        0.71614224 0.40587066 0.5       ]
 [0.72306997 0.5        0.40587066 0.67942529]] loss 0.9222286055464863 predictions [tensor(0.60122948, requires_grad=True), tensor(0.57
Iter:    61 | Cost: 0.5491328 | Accuracy: 1.0000000
step 61 [[0.3        0.71889415 0.40773653 0.5       ]
 [0.72689628 0.5        0.40773653 0.68380983]] 0.8614483140015595
step 61 params [[0.3        0.71889415 0.40773653 0.5       ]
 [0.72689628 0.5        0.40773653 0.68380983]] loss 0.8614483140015595 predictions [tensor(0.59999358, requires_grad=True), tensor(0.57
Iter:    62 | Cost: 0.5519109 | Accuracy: 1.0000000
step 62 [[0.3        0.72261504 0.40837688 0.5       ]
 [0.73140071 0.5        0.40837688 0.68796162]] 0.9162746933142506
step 62 params [[0.3        0.72261504 0.40837688 0.5       ]
 [0.73140071 0.5        0.40837688 0.68796162]] loss 0.9162746933142506 predictions [tensor(0.59834125, requires_grad=True), tensor(0.57
Iter:    63 | Cost: 0.5135940 | Accuracy: 1.0000000
step 63 [[0.3        0.72637417 0.40956331 0.5       ]
 [0.7363066  0.5        0.40956331 0.69298588]] 0.8537582774311447
step 63 params [[0.3        0.72637417 0.40956331 0.5       ]
 [0.7363066  0.5        0.40956331 0.69298588]] loss 0.8537582774311447 predictions [tensor(0.59664834, requires_grad=True), tensor(0.57
Iter:    64 | Cost: 0.9079465 | Accuracy: 1.0000000
step 64 [[0.3        0.72891588 0.41138671 0.5       ]
 [0.74002745 0.5        0.41138671 0.69745856]] 0.5164273853171242
step 64 params [[0.3        0.72891588 0.41138671 0.5       ]
 [0.74002745 0.5        0.41138671 0.69745856]] loss 0.5164273853171242 predictions [tensor(0.59551268, requires_grad=True), tensor(0.57
Iter:    65 | Cost: 0.5183326 | Accuracy: 1.0000000
step 65 [[0.3        0.73159243 0.41357614 0.5       ]
 [0.74421833 0.5        0.41357614 0.70275349]] 0.8457568947703011
step 65 params [[0.3        0.73159243 0.41357614 0.5       ]
 [0.74421833 0.5        0.41357614 0.70275349]] loss 0.8457568947703011 predictions [tensor(0.59429793, requires_grad=True), tensor(0.56
Iter:    66 | Cost: 0.8416820 | Accuracy: 1.0000000
step 66 [[0.3        0.73374901 0.41518411 0.5       ]
 [0.74738626 0.5        0.41518411 0.70654613]] 0.5638484901443899
step 66 params [[0.3        0.73374901 0.41518411 0.5       ]
 [0.74738626 0.5        0.41518411 0.70654613]] loss 0.5638484901443899 predictions [tensor(0.5933637, requires_grad=True), tensor(0.567
Iter:    67 | Cost: 0.5661114 | Accuracy: 1.0000000
step 67 [[0.3        0.73483084 0.41747065 0.5       ]
 [0.74953003 0.5        0.41747065 0.70991648]] 0.5219477450051945
step 67 params [[0.3        0.73483084 0.41747065 0.5       ]
 [0.74953003 0.5        0.41747065 0.70991648]] loss 0.5219477450051945 predictions [tensor(0.59292223, requires_grad=True), tensor(0.56
```

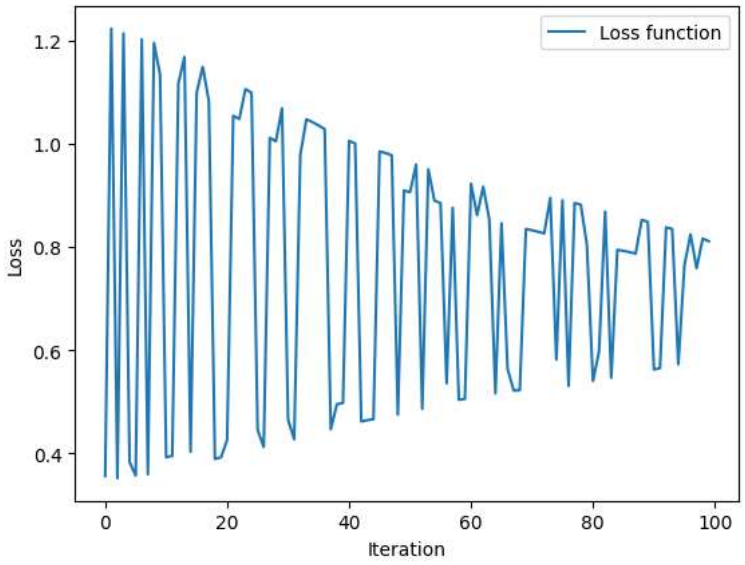
```
 [0.74993665 0.5        0.41747669 0.76991648]] loss 0.9219477430651945 predictions [tensor(0.59292225, requires_grad=True), tensor(0.56
Iter:    68 | Cost: 0.5678934 | Accuracy: 1.0000000
step 68 [[0.3        0.73494349 0.42038747 0.5       ]
 [0.75074696 0.5        0.42038747 0.7129065 ]] 0.5226920341606585
step 68 params [[0.3        0.73494349 0.42038747 0.5       ]
 [0.75074696 0.5        0.42038747 0.7129065 ]] loss 0.5226920341606585 predictions [tensor(0.59292615, requires_grad=True), tensor(0.56
Iter:    69 | Cost: 0.5692427 | Accuracy: 1.0000000
step 69 [[0.3        0.73539764 0.42347488 0.5       ]
 [0.75266467 0.5        0.42347488 0.71685636]] 0.8346045182377501
step 69 params [[0.3        0.73539764 0.42347488 0.5       ]
 [0.75266467 0.5        0.42347488 0.71685636]] loss 0.8346045182377501 predictions [tensor(0.59274603, requires_grad=True), tensor(0.56
Iter:    70 | Cost: 0.5229893 | Accuracy: 1.0000000
step 70 [[0.3        0.73615877 0.42669776 0.5       ]
 [0.75521004 0.5        0.42669776 0.72164813]] 0.8322189232067577
step 70 params [[0.3        0.73615877 0.42669776 0.5       ]
 [0.75521004 0.5        0.42669776 0.72164813]] loss 0.8322189232067577 predictions [tensor(0.59240658, requires_grad=True), tensor(0.56
Iter:    71 | Cost: 0.5235621 | Accuracy: 1.0000000
step 71 [[0.3        0.73719418 0.43001755 0.5       ]
 [0.75831625 0.5        0.43001755 0.72717677]] 0.8293040704678951
step 71 params [[0.3        0.73719418 0.43001755 0.5       ]
 [0.75831625 0.5        0.43001755 0.72717677]] loss 0.8293040704678951 predictions [tensor(0.59193013, requires_grad=True), tensor(0.56
Iter:    72 | Cost: 0.8259465 | Accuracy: 1.0000000
step 72 [[0.3        0.73847307 0.4333931  0.5       ]
 [0.76192238 0.5        0.4333931  0.73334876]] 0.8259464890881831
step 72 params [[0.3        0.73847307 0.4333931  0.5       ]
 [0.76192238 0.5        0.4333931  0.73334876]] loss 0.8259464890881831 predictions [tensor(0.5913365, requires_grad=True), tensor(0.560
Iter:    73 | Cost: 0.5788421 | Accuracy: 1.0000000
step 73 [[0.3        0.74086951 0.43498772 0.5       ]
 [0.76618951 0.5        0.43498772 0.73902461]] 0.8948632113454448
step 73 params [[0.3        0.74086951 0.43498772 0.5       ]
 [0.76618951 0.5        0.43498772 0.73902461]] loss 0.8948632113454448 predictions [tensor(0.59023207, requires_grad=True), tensor(0.55
Iter:    74 | Cost: 0.8921643 | Accuracy: 1.0000000
step 74 [[0.3        0.74279131 0.4362165  0.5       ]
 [0.76943091 0.5        0.4362165  0.74318299]] 0.5817944097128707
step 74 params [[0.3        0.74279131 0.4362165  0.5       ]
 [0.76943091 0.5        0.4362165  0.74318299]] loss 0.5817944097128707 predictions [tensor(0.58938768, requires_grad=True), tensor(0.55
Iter:    75 | Cost: 0.5286711 | Accuracy: 1.0000000
step 75 [[0.3        0.74575222 0.4358418  0.5       ]
 [0.77336152 0.5        0.4358418  0.74701263]] 0.8901057599706381
step 75 params [[0.3        0.74575222 0.4358418  0.5       ]
 [0.77336152 0.5        0.4358418  0.74701263]] loss 0.8901057599706381 predictions [tensor(0.58804158, requires_grad=True), tensor(0.55
Iter:    76 | Cost: 0.5309576 | Accuracy: 1.0000000
step 76 [[0.3        0.74755648 0.43654777 0.5       ]
 [0.77620501 0.5        0.43654777 0.75046013]] 0.5309576234157145
step 76 params [[0.3        0.74755648 0.43654777 0.5       ]
 [0.77620501 0.5        0.43654777 0.75046013]] loss 0.5309576234157145 predictions [tensor(0.58724899, requires_grad=True), tensor(0.55
Iter:    77 | Cost: 0.5323064 | Accuracy: 1.0000000
step 77 [[0.3        0.7503984  0.43568476 0.5       ]
 [0.77977265 0.5        0.43568476 0.75363075]] 0.8849107601115023
step 77 params [[0.3        0.7503984  0.43568476 0.5       ]
 [0.77977265 0.5        0.43568476 0.75363075]] loss 0.8849107601115023 predictions [tensor(0.58594402, requires_grad=True), tensor(0.55
Iter:    78 | Cost: 0.5345310 | Accuracy: 1.0000000
step 78 [[0.3        0.75416028 0.43342644 0.5       ]
 [0.7839884  0.5        0.43342644 0.75654372]] 0.8817540912329753
step 78 params [[0.3        0.75416028 0.43342644 0.5       ]
 [0.7839884  0.5        0.43342644 0.75654372]] loss 0.8817540912329753 predictions [tensor(0.58417638, requires_grad=True), tensor(0.55
Iter:    79 | Cost: 0.8774941 | Accuracy: 1.0000000
step 79 [[0.3        0.75786695 0.43149718 0.5       ]
 [0.78857391 0.5        0.43149718 0.76037704]] 0.8051774338579346
step 79 params [[0.3        0.75786695 0.43149718 0.5       ]
 [0.78857391 0.5        0.43149718 0.76037704]] loss 0.8051774338579346 predictions [tensor(0.58238826, requires_grad=True), tensor(0.55
Iter:    80 | Cost: 0.5406179 | Accuracy: 1.0000000
step 80 [[0.3        0.76033619 0.43086262 0.5       ]
 [0.7919968  0.5        0.43086262 0.76384407]] 0.5406179422710996
step 80 params [[0.3        0.76033619 0.43086262 0.5       ]
 [0.7919968  0.5        0.43086262 0.76384407]] loss 0.5406179422710996 predictions [tensor(0.58121087, requires_grad=True), tensor(0.55
Iter:    81 | Cost: 0.7994967 | Accuracy: 1.0000000
step 81 [[0.3        0.76234596 0.43023776 0.5       ]
 [0.79445578 0.5        0.43023776 0.7659611 ]] 0.5970285583643278
step 81 params [[0.3        0.76234596 0.43023776 0.5       ]
 [0.79445578 0.5        0.43023776 0.7659611 ]] loss 0.5970285583643278 predictions [tensor(0.58030017, requires_grad=True), tensor(0.54
Iter:    82 | Cost: 0.5984290 | Accuracy: 1.0000000
step 82 [[0.3        0.76536214 0.4281304  0.5       ]
 [0.79767403 0.5        0.4281304  0.76789297]] 0.8682155123366534
step 82 params [[0.3        0.76536214 0.4281304  0.5       ]
 [0.79767403 0.5        0.4281304  0.76789297]] loss 0.8682155123366534 predictions [tensor(0.57886991, requires_grad=True), tensor(0.54
Iter:    83 | Cost: 0.8648135 | Accuracy: 1.0000000
step 83 [[0.3        0.76719831 0.42737011 0.5       ]
 [0.79984906 0.5        0.42737011 0.76964364]] 0.5466775137470777
step 83 params [[0.3        0.76719831 0.42737011 0.5       ]
 [0.79984906 0.5        0.42737011 0.76964364]] loss 0.5466775137470777 predictions [tensor(0.57802635, requires_grad=True), tensor(0.54
Iter:    84 | Cost: 0.5481358 | Accuracy: 1.0000000
step 84 [[0.3        0.76913852 0.4266849  0.5       ]
 [0.8025979  0.5        0.4266849  0.77245078]] 0.7943096756450108
step 84 params [[0.3        0.76913852 0.4266849  0.5       ]
```

```
 [0.8025979  0.5        0.4266849  0.77245078]] loss 0.7943096756450108 predictions [tensor(0.57707007, requires_grad=True), tensor(0.54
Iter:    85 | Cost: 0.5497916 | Accuracy: 1.0000000
step 85 [[0.3        0.77117051 0.42604623 0.5       ]
 [0.80586563 0.5        0.42604623 0.77619633]] 0.7922922451500121
step 85 params [[0.3        0.77117051 0.42604623 0.5       ]
 [0.80586563 0.5        0.42604623 0.77619633]] loss 0.7922922451500121 predictions [tensor(0.5760117, requires_grad=True), tensor(0.546
Iter:    86 | Cost: 0.6050209 | Accuracy: 1.0000000
step 86 [[0.3        0.77328177 0.42542238 0.5       ]
 [0.80960191 0.5        0.42542238 0.78077452]] 0.7897968974778024
step 86 params [[0.3        0.77328177 0.42542238 0.5       ]
 [0.80960191 0.5        0.42542238 0.78077452]] loss 0.7897968974778024 predictions [tensor(0.57486073, requires_grad=True), tensor(0.54
Iter:    87 | Cost: 0.8553385 | Accuracy: 1.0000000
step 87 [[0.3        0.77545959 0.42477934 0.5       ]
 [0.81376073 0.5        0.42477934 0.78609065]] 0.7868807008923013
step 87 params [[0.3        0.77545959 0.42477934 0.5       ]
 [0.81376073 0.5        0.42477934 0.78609065]] loss 0.7868807008923013 predictions [tensor(0.5736255, requires_grad=True), tensor(0.543
Iter:    88 | Cost: 0.5557785 | Accuracy: 1.0000000
step 88 [[0.3        0.77863238 0.42255096 0.5       ]
 [0.81850356 0.5        0.42255096 0.79089619]] 0.8524372037202042
step 88 params [[0.3        0.77863238 0.42255096 0.5       ]
 [0.81850356 0.5        0.42255096 0.79089619]] loss 0.8524372037202042 predictions [tensor(0.57188357, requires_grad=True), tensor(0.54
Iter:    89 | Cost: 0.5588199 | Accuracy: 1.0000000
step 89 [[0.3        0.78268561 0.41892675 0.5       ]
 [0.82376747 0.5        0.41892675 0.79523184]] 0.848360091247953
step 89 params [[0.3        0.78268561 0.41892675 0.5       ]
 [0.82376747 0.5        0.41892675 0.79523184]] loss 0.848360091247953 predictions [tensor(0.56968686, requires_grad=True), tensor(0.540
Iter:    90 | Cost: 0.6161338 | Accuracy: 1.0000000
step 90 [[0.3        0.78542928 0.41692643 0.5       ]
 [0.82777103 0.5        0.41692643 0.79919147]] 0.56266843130107
step 90 params [[0.3        0.78542928 0.41692643 0.5       ]
 [0.82777103 0.5        0.41692643 0.79919147]] loss 0.56266843130107 predictions [tensor(0.56818763, requires_grad=True), tensor(0.5387
Iter:    91 | Cost: 0.8397641 | Accuracy: 1.0000000
step 91 [[0.3        0.78699031 0.41639036 0.5       ]
 [0.83063172 0.5        0.41639036 0.80281406]] 0.5653035862875575
step 91 params [[0.3        0.78699031 0.41639036 0.5       ]
 [0.83063172 0.5        0.41639036 0.80281406]] loss 0.5653035862875575 predictions [tensor(0.56732301, requires_grad=True), tensor(0.53
Iter:    92 | Cost: 0.8377638 | Accuracy: 1.0000000
step 92 [[0.3        0.78958031 0.41427477 0.5       ]
 [0.83419665 0.5        0.41427477 0.80605857]] 0.8377638031527926
step 92 params [[0.3        0.78958031 0.41427477 0.5       ]
 [0.83419665 0.5        0.41427477 0.80605857]] loss 0.8377638031527926 predictions [tensor(0.56591425, requires_grad=True), tensor(0.53
Iter:    93 | Cost: 0.5693127 | Accuracy: 1.0000000
step 93 [[0.3        0.79308461 0.41077278 0.5       ]
 [0.8383925  0.5        0.41077278 0.80895465]] 0.8345131762313364
step 93 params [[0.3        0.79308461 0.41077278 0.5       ]
 [0.8383925  0.5        0.41077278 0.80895465]] loss 0.8345131762313364 predictions [tensor(0.5640194, requires_grad=True), tensor(0.535
Iter:    94 | Cost: 0.5726666 | Accuracy: 1.0000000
step 94 [[0.3        0.79533249 0.40888673 0.5       ]
 [0.8414184  0.5        0.40888673 0.81162418]] 0.5726666260631433
step 94 params [[0.3        0.79533249 0.40888673 0.5       ]
 [0.8414184  0.5        0.40888673 0.81162418]] loss 0.5726666260631433 predictions [tensor(0.56280025, requires_grad=True), tensor(0.53
Iter:    95 | Cost: 0.8273651 | Accuracy: 1.0000000
step 95 [[0.3        0.79757617 0.4068964  0.5       ]
 [0.84493742 0.5        0.4068964  0.81527953]] 0.7641381555254311
step 95 params [[0.3        0.79757617 0.4068964  0.5       ]
 [0.84493742 0.5        0.4068964  0.81527953]] loss 0.7641381555254311 predictions [tensor(0.56148623, requires_grad=True), tensor(0.53
Iter:    96 | Cost: 0.5771680 | Accuracy: 1.0000000
step 96 [[0.3        0.80076251 0.40350945 0.5       ]
 [0.84908937 0.5        0.40350945 0.81853309]] 0.8243640574753519
step 96 params [[0.3        0.80076251 0.40350945 0.5       ]
 [0.84908937 0.5        0.40350945 0.81853309]] loss 0.8243640574753519 predictions [tensor(0.55967614, requires_grad=True), tensor(0.53
Iter:    97 | Cost: 0.7587343 | Accuracy: 1.0000000
step 97 [[0.3        0.80384413 0.40011839 0.5       ]
 [0.85362675 0.5        0.40011839 0.82271409]] 0.758734273924844
step 97 params [[0.3        0.80384413 0.40011839 0.5       ]
 [0.85362675 0.5        0.40011839 0.82271409]] loss 0.758734273924844 predictions [tensor(0.55782359, requires_grad=True), tensor(0.530
Iter:    98 | Cost: 0.5837125 | Accuracy: 1.0000000
step 98 [[0.3        0.80777603 0.39549169 0.5       ]
 [0.85869155 0.5        0.39549169 0.82643337]] 0.8160463555123018
step 98 params [[0.3        0.80777603 0.39549169 0.5       ]
 [0.85869155 0.5        0.39549169 0.82643337]] loss 0.8160463555123018 predictions [tensor(0.55553151, requires_grad=True), tensor(0.52
Iter:    99 | Cost: 0.7520821 | Accuracy: 1.0000000
step 99 [[0.3        0.8124619  0.38978669 0.5       ]
 [0.86422814 0.5        0.38978669 0.8297268 ]] 0.8108761107812491
step 99 params [[0.3        0.8124619  0.38978669 0.5       ]
 [0.86422814 0.5        0.38978669 0.8297268 ]] loss 0.8108761107812491 predictions [tensor(0.55284917, requires_grad=True), tensor(0.52
Iter:   100 | Cost: 0.6406895 | Accuracy: 1.0000000
'state_preparation(x)\n\nlosses = []\nfor i in range(400):\n  params, loss = optimizer.step_and_cost(circuit, params)\n  losses.append
(loss)\n  print(loss)'
```

```
import matplotlib.pyplot as plt

plt.plot(losses, label='Loss function')
plt.xlabel('Iteration')
plt.ylabel('Loss')
plt.legend()
```

```
<matplotlib.legend.Legend at 0x78d6193080a0>
```



```
import pennylane as qml
import torch
from torch.autograd import Variable
from pennylane import numpy as np

num_wires = 4
num_layers = 2

dev = qml.device("default.qubit", wires=num_wires)

coeffs = [1]
obs = [qml.PauliZ(0) @ qml.PauliZ(1)]
H = qml.Hamiltonian(coeffs, obs)

#START HERE
'''
@qml.qnode(dev, interface='autograd')
def circuit(params, x=x):
  qml.RX(params[0], wires=0)
  qml.RZ(params[1], wires=0)
  for i in range(1, num_wires):
    qml.CNOT([i-1, i])
    qml.RX(params[0], wires=i)
    qml.RZ(params[1], wires=i)
  return qml.expval(qml.PauliZ(0))

def circuit_0_1(params):
  print('check i', circuit(params))
  return (circuit(params) + 1)/2
'''

@qml.qnode(dev, interface='autograd')
def circuit(params, x):
  x /= np.linalg.norm(x)
  #qml.AmplitudeEmbedding(x, wires=list(range(num_wires)))
  qml.AngleEmbedding(features=x, wires=range(num_wires), rotation='Y')
  qml.BasicEntanglerLayers(weights=params, wires=range(num_wires))

  return qml.expval(H)

  #print('state', qml.state())
  ##print('full_circuit_1', circuit_0_1(params))
  #return circuit_0_1(params)

#def binary_cost(params):
#  return circuit_0_1

def full_circuit(params, x):
  ###print('circuit result', circuit(params, x))
  return (np.array(circuit(params, x)).mean() + 1)/2

def square_loss(labels, predictions):
    loss = 0
    ##print(labels, predictions)
    for true, pred in zip(labels, predictions):
        ##print(true, pred)
        loss = loss + (true - pred) ** 2

    loss = loss / len(labels)
    return loss

def cross_loss(labels, predictions):
    loss = 0
    for true, pred in zip(labels, predictions):
      #print(true*np.log(pred + 1))
      loss = loss + true*np.log(pred) + (1-true)*np.log(1 - pred)
    loss = - loss/len(labels)
    return loss

def accuracy(labels, predictions):
    #loss = (labels - predictions).sum()
    loss1 = 0
    loss2 = 0
    ##print('accuracy', labels, predictions)
    for true, pred in zip(labels, predictions):
        if true - pred < 0.5:
            loss1 = loss1 + 1
```

```
            if true - pred > 0.5:
                loss2 = loss2 + 1
        acc1 = loss1 / len(labels)
        acc2 = loss2 / len(labels)
        return max(acc1, acc2)

    def cost(params, X_batch, y_batch):
        #predictions = full_circuit(params, X_batch)
        ##print('X_batch.shape',X_batch.shape)
        data_indices = [np.random.randint(0, len(X)) for _ in range(1)]
        predictions = [full_circuit(params, X_batch[index]) for index in data_indices]
        ##print('len(predictions)',len(predictions))
        #predictions = full_circuit(params, X_batch.flatten())
        ##print('Cost',y_batch, predictions)
        return cross_loss(y_batch[data_indices], predictions)

    '''
    def circuit_probs(params):
      qml.RX(params[0], wires=0)
      qml.RZ(params[1], wires=0)
      return qml.probs(wires=0)

    def cost_probs(params):
      return circuit_probs - data'''

    optimizer = qml.AdamOptimizer(stepsize=0.01)
    params = np.array([[0.3, 0.5, 0.3, 0.5], [0.5, 0.5, 0.3, 0.5]], requires_grad=True)

    X = np.array([[1., 1., 0., 0.], [0., 0., 1., 1.], [1., 0., 1., 0.], [0., 1., 0., 1.]], requires_grad=True)
    y = np.array([0, 0, 1, 1], requires_grad=True)

    losses = []

    for it in range(100):
      #data_index = np.random.randint(0, len(X))
      #X_batch = X[data_index]
      #y_batch = y[data_index]

      X_batch = X
      y_batch = y
      params, loss = optimizer.step_and_cost(cost, params, X_batch=X_batch, y_batch=y_batch)
      print('step', it, params, loss)
      losses.append(loss)

      predictions = [full_circuit(params, x) for x in X]
      print('step', it, 'params', params, 'loss', loss, 'predictions', predictions)
      #predictions = full_circuit(params, X.flatten())
      acc = accuracy(y, predictions)

      print(
          "Iter: {:5d} | Cost: {:0.7f} | Accuracy: {:0.7f} ".format(
              it + 1, cost(params, X, y), acc
          )
      )

    '''state_preparation(x)

    losses = []
    for i in range(400):
      params, loss = optimizer.step_and_cost(circuit, params)
      losses.append(loss)
      print(loss)'''


    #END HERE
```

```
step 0 [[0.3        0.49000001 0.29000001 0.5       ]
 [0.49       0.5        0.29000001 0.49      ]] 0.38399258872749886
step 0 params [[0.3        0.49000001 0.29000001 0.5       ]
 [0.49       0.5        0.29000001 0.49      ]] loss 0.38399258872749886 predictions [te
Iter:     1 | Cost: 1.1636282 | Accuracy: 1.0000000
step 1 [[0.3        0.47999957 0.28000087 0.5       ]
 [0.48000094 0.5        0.28000087 0.48000094]] 0.3744765445158933
step 1 params [[0.3        0.47999957 0.28000087 0.5       ]
 [0.48000094 0.5        0.28000087 0.48000094]] loss 0.3744765445158933 predictions [ter
Iter:     2 | Cost: 0.3411845 | Accuracy: 1.0000000
step 2 [[0.3        0.4699988  0.27000406 0.5       ]
 [0.47000386 0.5        0.27000406 0.47000386]] 0.36499634184488255
step 2 params [[0.3        0.4699988  0.27000406 0.5       ]
 [0.47000386 0.5        0.27000406 0.47000386]] loss 0.36499634184488255 predictions [te
Iter:     3 | Cost: 1.2065602 | Accuracy: 1.0000000
step 3 [[0.3        0.46025726 0.260876   0.5       ]
 [0.46000331 0.5        0.260876   0.46060291]] 0.33346081351504936
step 3 params [[0.3        0.46025726 0.260876   0.5       ]
 [0.46000331 0.5        0.260876   0.46060291]] loss 0.33346081351504936 predictions [te
Iter:     4 | Cost: 0.3259593 | Accuracy: 1.0000000
step 4 [[0.3        0.45061614 0.2514856  0.5       ]
 [0.45001155 0.5        0.2514856  0.45101497]] 0.3466131691259224
step 4 params [[0.3        0.45061614 0.2514856  0.5       ]
 [0.45001155 0.5        0.2514856  0.45101497]] loss 0.3466131691259224 predictions [ter
Iter:     5 | Cost: 1.2992581 | Accuracy: 1.0000000
step 5 [[0.3        0.45120662 0.2463111  0.5       ]
 [0.44843885 0.5        0.2463111  0.4466259 ]] 1.2992581076106202
step 5 params [[0.3        0.45120662 0.2463111  0.5       ]
 [0.44843885 0.5        0.2463111  0.4466259 ]] loss 1.2992581076106202 predictions [ter
Iter:     6 | Cost: 0.3348848 | Accuracy: 1.0000000
step 6 [[0.3        0.45086659 0.24015987 0.5       ]
 [0.44574652 0.5        0.24015987 0.44116348]] 0.3348848438309738
step 6 params [[0.3        0.45086659 0.24015987 0.5       ]
 [0.44574652 0.5        0.24015987 0.44116348]] loss 0.3348848438309738 predictions [ter
Iter:     7 | Cost: 1.2660531 | Accuracy: 1.0000000
step 7 [[0.3        0.44974244 0.2333099  0.5       ]
 [0.44216053 0.5        0.2333099  0.43491012]] 0.3312052550543617
step 7 params [[0.3        0.44974244 0.2333099  0.5       ]
 [0.44216053 0.5        0.2333099  0.43491012]] loss 0.3312052550543617 predictions [ter
Iter:     8 | Cost: 0.3268315 | Accuracy: 1.0000000
step 8 [[0.3        0.45077747 0.23215451 0.5       ]
 [0.44224296 0.5        0.23215451 0.43408577]] 1.2772793490725443
step 8 params [[0.3        0.45077747 0.23215451 0.5       ]
 [0.44224296 0.5        0.23215451 0.43408577]] loss 1.2772793490725443 predictions [ter
Iter:     9 | Cost: 1.3151544 | Accuracy: 1.0000000
step 9 [[0.3        0.45094981 0.2299401  0.5       ]
 [0.44129639 0.5        0.2299401  0.43216221]] 0.32652655606184683
step 9 params [[0.3        0.45094981 0.2299401  0.5       ]
 [0.44129639 0.5        0.2299401  0.43216221]] loss 0.32652655606184683 predictions [te
Iter:    10 | Cost: 0.3252873 | Accuracy: 1.0000000
step 10 [[0.3        0.45405964 0.22920502 0.5       ]
 [0.44302176 0.5        0.22920502 0.43200839]] 1.3169110317785198
step 10 params [[0.3        0.45405964 0.22920502 0.5       ]
 [0.44302176 0.5        0.22920502 0.43200839]] loss 1.3169110317785198 predictions [ter
Iter:    11 | Cost: 0.3132097 | Accuracy: 1.0000000
step 11 [[0.3        0.45626663 0.22743499 0.5       ]
 [0.44369517 0.5        0.22743499 0.43075961]] 0.3260125319036725
step 11 params [[0.3        0.45626663 0.22743499 0.5       ]
 [0.44369517 0.5        0.22743499 0.43075961]] loss 0.3260125319036725 predictions [ter
Iter:    12 | Cost: 1.3119953 | Accuracy: 1.0000000
step 12 [[0.3        0.45960013 0.22855237 0.5       ]
 [0.44625986 0.5        0.22855237 0.4322684 ]] 1.2799491211934944
step 12 params [[0.3        0.45960013 0.22855237 0.5       ]
 [0.44625986 0.5        0.22855237 0.4322684 ]] loss 1.2799491211934944 predictions [ter
Iter:    13 | Cost: 1.3066853 | Accuracy: 1.0000000
step 13 [[0.3        0.46386297 0.23163681 0.5       ]
 [0.45024891 0.5        0.23163681 0.43566537]] 1.2751703824951732
step 13 params [[0.3        0.46386297 0.23163681 0.5       ]
 [0.45024891 0.5        0.23163681 0.43566537]] loss 1.2751703824951732 predictions [ter
Iter:    14 | Cost: 0.3187321 | Accuracy: 1.0000000
step 14 [[0.3        0.46889522 0.23616016 0.5       ]
 [0.45534241 0.5        0.23616016 0.44044442]] 1.266656014388765
step 14 params [[0.3        0.46889522 0.23616016 0.5       ]
 [0.45534241 0.5        0.23616016 0.44044442]] loss 1.266656014388765 predictions [tens
Iter:    15 | Cost: 0.3226087 | Accuracy: 1.0000000
step 15 [[0.3        0.47456633 0.24178126 0.5       ]
 [0.46130996 0.5        0.24178126 0.44627643]] 1.2554115443443568
step 15 params [[0.3        0.47456633 0.24178126 0.5       ]
 [0.46130996 0.5        0.24178126 0.44627643]] loss 1.2554115443443568 predictions [ter
Iter:    16 | Cost: 1.2764524 | Accuracy: 1.0000000
step 16 [[0.3        0.47918617 0.24609544 0.5       ]
 [0.4660355  0.5        0.24609544 0.45078526]] 0.34075413855408837
step 16 params [[0.3        0.47918617 0.24609544 0.5       ]
 [0.4660355  0.5        0.24609544 0.45078526]] loss 0.34075413855408837 predictions [te
```

```
Iter:    17 | Cost: 1.2318462 | Accuracy: 1.0000000
step 17 [[0.3        0.48232581 0.24976692 0.5       ]
 [0.46960674 0.5        0.24976692 0.45454355]] 0.33077007528609403
step 17 params [[0.3        0.48232581 0.24976692 0.5       ]
 [0.46960674 0.5        0.24976692 0.45454355]] loss 0.33077007528609403 predictions [te
Iter:    18 | Cost: 0.3482912 | Accuracy: 1.0000000
step 18 [[0.3        0.48466331 0.25231841 0.5       ]
 [0.47217173 0.5        0.25231841 0.45718222]] 0.3482912099484531
step 18 params [[0.3        0.48466331 0.25231841 0.5       ]
 [0.47217173 0.5        0.25231841 0.45718222]] loss 0.3482912099484531 predictions [ter
Iter:    19 | Cost: 1.2181386 | Accuracy: 1.0000000
step 19 [[0.3        0.48878859 0.25524109 0.5       ]
 [0.47602279 0.5        0.25524109 0.46038732]] 1.2553623747056557
step 19 params [[0.3        0.48878859 0.25524109 0.5       ]
 [0.47602279 0.5        0.25524109 0.46038732]] loss 1.2553623747056557 predictions [ter
Iter:    20 | Cost: 0.3539156 | Accuracy: 1.0000000
step 20 [[0.3        0.49425271 0.25848731 0.5       ]
 [0.48092455 0.5        0.25848731 0.46409078]] 1.2479639773503384
step 20 params [[0.3        0.49425271 0.25848731 0.5       ]
 [0.48092455 0.5        0.25848731 0.46409078]] loss 1.2479639773503384 predictions [ter
Iter:    21 | Cost: 0.3421640 | Accuracy: 1.0000000
step 21 [[0.3        0.50075631 0.26200916 0.5       ]
 [0.48669397 0.5        0.26200916 0.46822978]] 1.238673577682533
step 21 params [[0.3        0.50075631 0.26200916 0.5       ]
 [0.48669397 0.5        0.26200916 0.46822978]] loss 1.238673577682533 predictions [tens
Iter:    22 | Cost: 1.1906147 | Accuracy: 1.0000000
step 22 [[0.3        0.50583354 0.26498852 0.5       ]
 [0.49129245 0.5        0.26498852 0.47166976]] 0.34658603561176576
step 22 params [[0.3        0.50583354 0.26498852 0.5       ]
 [0.49129245 0.5        0.26498852 0.47166976]] loss 0.34658603561176576 predictions [te
Iter:    23 | Cost: 1.2194415 | Accuracy: 1.0000000
step 23 [[0.3        0.51195247 0.26823246 0.5       ]
 [0.49676733 0.5        0.26823246 0.47555275]] 1.2194414587570166
step 23 params [[0.3        0.51195247 0.26823246 0.5       ]
 [0.49676733 0.5        0.26823246 0.47555275]] loss 1.2194414587570166 predictions [ter
Iter:    24 | Cost: 0.3704855 | Accuracy: 1.0000000
step 24 [[0.3        0.51671982 0.27097117 0.5       ]
 [0.5011094  0.5        0.27097117 0.47876268]] 0.3543128066756202
step 24 params [[0.3        0.51671982 0.27097117 0.5       ]
 [0.5011094  0.5        0.27097117 0.47876268]] loss 0.3543128066756202 predictions [ter
Iter:    25 | Cost: 1.1647285 | Accuracy: 1.0000000
step 25 [[0.3        0.52067617 0.2726457  0.5       ]
 [0.50445887 0.5        0.2726457  0.48088081]] 0.3739771345722296
step 25 params [[0.3        0.52067617 0.2726457  0.5       ]
 [0.50445887 0.5        0.2726457  0.48088081]] loss 0.3739771345722296 predictions [ter
Iter:    26 | Cost: 1.1591482 | Accuracy: 1.0000000
step 26 [[0.3        0.52347119 0.27396331 0.5       ]
 [0.5068596  0.5        0.27396331 0.48248326]] 0.36024446400762833
step 26 params [[0.3        0.52347119 0.27396331 0.5       ]
 [0.5068596  0.5        0.27396331 0.48248326]] loss 0.36024446400762833 predictions [te
Iter:    27 | Cost: 1.1913969 | Accuracy: 1.0000000
step 27 [[0.3        0.52562456 0.27434618 0.5       ]
 [0.50843715 0.5        0.27434618 0.48313645]] 0.3783855228356731
step 27 params [[0.3        0.52562456 0.27434618 0.5       ]
 [0.50843715 0.5        0.27434618 0.48313645]] loss 0.3783855228356731 predictions [ter
Iter:    28 | Cost: 0.3633689 | Accuracy: 1.0000000
step 28 [[0.3        0.5292143  0.27516355 0.5       ]
 [0.51125889 0.5        0.27516355 0.48447456]] 1.1885257654452102
step 28 params [[0.3        0.5292143  0.27516355 0.5       ]
 [0.51125889 0.5        0.27516355 0.48447456]] loss 1.1885257654452102 predictions [ter
Iter:    29 | Cost: 0.3813689 | Accuracy: 1.0000000
step 29 [[0.3        0.53399443 0.27636525 0.5       ]
 [0.51514687 0.5        0.27636525 0.48642577]] 1.1835301718230606
step 29 params [[0.3        0.53399443 0.27636525 0.5       ]
 [0.51514687 0.5        0.27636525 0.48642577]] loss 1.1835301718230606 predictions [ter
Iter:    30 | Cost: 0.3840508 | Accuracy: 1.0000000
step 30 [[0.3        0.53797964 0.27663326 0.5       ]
 [0.51809425 0.5        0.27663326 0.48738153]] 0.3840507764610341
step 30 params [[0.3        0.53797964 0.27663326 0.5       ]
 [0.51809425 0.5        0.27663326 0.48738153]] loss 0.3840507764610341 predictions [ter
Iter:    31 | Cost: 1.1390292 | Accuracy: 1.0000000
step 31 [[0.3        0.54307782 0.27730219 0.5       ]
 [0.52207755 0.5        0.27730219 0.4889696 ]] 1.1715993158221336
step 31 params [[0.3        0.54307782 0.27730219 0.5       ]
 [0.52207755 0.5        0.27730219 0.4889696 ]] loss 1.1715993158221336 predictions [ter
Iter:    32 | Cost: 0.3884017 | Accuracy: 1.0000000
step 32 [[0.3        0.54911269 0.2783181  0.5       ]
 [0.52695525 0.5        0.2783181  0.49112062]] 1.1647436224569332
step 32 params [[0.3        0.54911269 0.2783181  0.5       ]
 [0.52695525 0.5        0.2783181  0.49112062]] loss 1.1647436224569332 predictions [ter
Iter:    33 | Cost: 0.3777467 | Accuracy: 1.0000000
step 33 [[0.3        0.55385023 0.27906886 0.5       ]
 [0.53075705 0.5        0.27906886 0.49275754]] 0.3777467361893918
step 33 params [[0.3        0.55385023 0.27906886 0.5       ]
 [0.53075705 0.5        0.27906886 0.49275754]] loss 0.3777467361893918 predictions [ter
```

```
Iter:    34 | Cost: 1.1500377 | Accuracy: 1.0000000
step 34 [[0.3        0.55780613 0.2789202  0.5       ]
 [0.53363139 0.5        0.2789202  0.49340213]] 0.39408279336162305
step 34 params [[0.3        0.55780613 0.2789202  0.5       ]
 [0.53363139 0.5        0.2789202  0.49340213]] loss 0.39408279336162305 predictions [te
Iter:    35 | Cost: 0.3829906 | Accuracy: 1.0000000
step 35 [[0.3        0.56213446 0.28043089 0.5       ]
 [0.53737991 0.5        0.28043089 0.49562889]] 1.1184327427594594
step 35 params [[0.3        0.56213446 0.28043089 0.5       ]
 [0.53737991 0.5        0.28043089 0.49562889]] loss 1.1184327427594594 predictions [ter
Iter:    36 | Cost: 1.1128068 | Accuracy: 1.0000000
step 36 [[0.3        0.56679896 0.28333286 0.5       ]
 [0.54189561 0.5        0.28333286 0.49918636]] 1.1128068213288826
step 36 params [[0.3        0.56679896 0.28333286 0.5       ]
 [0.54189561 0.5        0.28333286 0.49918636]] loss 1.1128068213288826 predictions [ter
Iter:    37 | Cost: 0.4021418 | Accuracy: 1.0000000
step 37 [[0.3        0.57245245 0.2862657  0.5       ]
 [0.54721871 0.5        0.2862657  0.50303701]] 1.131496858134743
step 37 params [[0.3        0.57245245 0.2862657  0.5       ]
 [0.54721871 0.5        0.2862657  0.50303701]] loss 1.131496858134743 predictions [tens
Iter:    38 | Cost: 1.1228998 | Accuracy: 1.0000000
step 38 [[0.3        0.5782886  0.2903607  0.5       ]
 [0.55310782 0.5        0.2903607  0.50797343]] 1.096890724409461
step 38 params [[0.3        0.5782886  0.2903607  0.5       ]
 [0.55310782 0.5        0.2903607  0.50797343]] loss 1.096890724409461 predictions [tens
Iter:    39 | Cost: 1.1135979 | Accuracy: 1.0000000
step 39 [[0.3        0.58428335 0.29543374 0.5       ]
 [0.55948902 0.5        0.29543374 0.51382628]] 1.0871130128011355
step 39 params [[0.3        0.58428335 0.29543374 0.5       ]
 [0.55948902 0.5        0.29543374 0.51382628]] loss 1.0871130128011355 predictions [ter
Iter:    40 | Cost: 1.0762460 | Accuracy: 1.0000000
step 40 [[0.3        0.59107491 0.30024699 0.5       ]
 [0.56642751 0.5        0.30024699 0.51969115]] 1.1037663627102001
step 40 params [[0.3        0.59107491 0.30024699 0.5       ]
 [0.56642751 0.5        0.30024699 0.51969115]] loss 1.1037663627102001 predictions [ter
Iter:    41 | Cost: 0.4081698 | Accuracy: 1.0000000
step 41 [[0.3        0.59690369 0.30385925 0.5       ]
 [0.57216378 0.5        0.30385925 0.52421884]] 0.42264680987362946
step 41 params [[0.3        0.59690369 0.30385925 0.5       ]
 [0.57216378 0.5        0.30385925 0.52421884]] loss 0.42264680987362946 predictions [te
Iter:    42 | Cost: 1.0845872 | Accuracy: 1.0000000
step 42 [[0.3        0.60140934 0.30709086 0.5       ]
 [0.57673006 0.5        0.30709086 0.52807867]] 0.4125521059591548
step 42 params [[0.3        0.60140934 0.30709086 0.5       ]
 [0.57673006 0.5        0.30709086 0.52807867]] loss 0.4125521059591548 predictions [ter
Iter:    43 | Cost: 1.0778338 | Accuracy: 1.0000000
step 43 [[0.3        0.60615579 0.31132108 0.5       ]
 [0.58188391 0.5        0.31132108 0.53293307]] 1.0491589147929414
step 43 params [[0.3        0.60615579 0.31132108 0.5       ]
 [0.58188391 0.5        0.31132108 0.53293307]] loss 1.0491589147929414 predictions [ter
Iter:    44 | Cost: 1.0408154 | Accuracy: 1.0000000
step 44 [[0.3        0.61181931 0.31524661 0.5       ]
 [0.58770036 0.5        0.31524661 0.53783176]] 1.070437831402362
step 44 params [[0.3        0.61181931 0.31524661 0.5       ]
 [0.58770036 0.5        0.31524661 0.53783176]] loss 1.070437831402362 predictions [tens
Iter:    45 | Cost: 0.4404725 | Accuracy: 1.0000000
step 45 [[0.3        0.61758793 0.32004736 0.5       ]
 [0.59394776 0.5        0.32004736 0.54357374]] 1.0320726259813653
step 45 params [[0.3        0.61758793 0.32004736 0.5       ]
 [0.59394776 0.5        0.32004736 0.54357374]] loss 1.0320726259813653 predictions [ter
Iter:    46 | Cost: 1.0224705 | Accuracy: 1.0000000
step 46 [[0.3        0.62248528 0.32367562 0.5       ]
 [0.59904936 0.5        0.32367562 0.54799176]] 0.4458265596953968
step 46 params [[0.3        0.62248528 0.32367562 0.5       ]
 [0.59904936 0.5        0.32367562 0.54799176]] loss 0.4458265596953968 predictions [ter
Iter:    47 | Cost: 1.0149463 | Accuracy: 1.0000000
step 47 [[0.3        0.62612427 0.32699659 0.5       ]
 [0.60302808 0.5        0.32699659 0.55177853]] 0.43257425018163564
step 47 params [[0.3        0.62612427 0.32699659 0.5       ]
 [0.60302808 0.5        0.32699659 0.55177853]] loss 0.43257425018163564 predictions [te
Iter:    48 | Cost: 0.4354657 | Accuracy: 1.0000000
step 48 [[0.3        0.63003892 0.33119814 0.5       ]
 [0.60760415 0.5        0.33119814 0.55649664]] 1.0088647595209117
step 48 params [[0.3        0.63003892 0.33119814 0.5       ]
 [0.60760415 0.5        0.33119814 0.55649664]] loss 1.0088647595209117 predictions [ter
Iter:    49 | Cost: 0.4576712 | Accuracy: 1.0000000
step 49 [[0.3        0.63277786 0.33506824 0.5       ]
 [0.61110132 0.5        0.33506824 0.56056576]] 0.4387030002756657
step 49 params [[0.3        0.63277786 0.33506824 0.5       ]
 [0.61110132 0.5        0.33506824 0.56056576]] loss 0.4387030002756657 predictions [ter
Iter:    50 | Cost: 1.0309416 | Accuracy: 1.0000000
step 50 [[0.3        0.63491609 0.3378615  0.5       ]
 [0.61369496 0.5        0.3378615  0.56346005]] 0.4610299735772336
step 50 params [[0.3        0.63491609 0.3378615  0.5       ]
 [0.61369496 0.5        0.3378615  0.56346005]] loss 0.4610299735772336 predictions [ter
```

```
Iter:    51 | Cost: 0.4634797 | Accuracy: 1.0000000
step 51 [[0.3        0.63824296 0.34028926 0.5       ]
 [0.61718502 0.5        0.34028926 0.56648981]] 1.0277186851400932
step 51 params [[0.3        0.63824296 0.34028926 0.5       ]
 [0.61718502 0.5        0.34028926 0.56648981]] loss 1.0277186851400932 predictions [ter
Iter:    52 | Cost: 0.4663332 | Accuracy: 1.0000000
step 52 [[0.3        0.64260318 0.34236176 0.5       ]
 [0.62146699 0.5        0.34236176 0.5696361 ]] 1.0232337590393403
step 52 params [[0.3        0.64260318 0.34236176 0.5       ]
 [0.62146699 0.5        0.34236176 0.5696361 ]] loss 1.0232337590393403 predictions [ter
Iter:    53 | Cost: 1.0176644 | Accuracy: 1.0000000
step 53 [[0.3        0.64712941 0.34539186 0.5       ]
 [0.62627949 0.5        0.34539186 0.57376023]] 0.9816164200791068
step 53 params [[0.3        0.64712941 0.34539186 0.5       ]
 [0.62627949 0.5        0.34539186 0.57376023]] loss 0.9816164200791068 predictions [ter
Iter:    54 | Cost: 0.9752651 | Accuracy: 1.0000000
step 54 [[0.3        0.6518014  0.34924886 0.5       ]
 [0.6315613  0.5        0.34924886 0.57873391]] 0.975265071677357
step 54 params [[0.3        0.6518014  0.34924886 0.5       ]
 [0.6315613  0.5        0.34924886 0.57873391]] loss 0.975265071677357 predictions [tens
Iter:    55 | Cost: 1.0053297 | Accuracy: 1.0000000
step 55 [[0.3        0.65521485 0.35288768 0.5       ]
 [0.63568992 0.5        0.35288768 0.58305868]] 0.4555864220313945
step 55 params [[0.3        0.65521485 0.35288768 0.5       ]
 [0.63568992 0.5        0.35288768 0.58305868]] loss 0.4555864220313945 predictions [ter
Iter:    56 | Cost: 0.9622637 | Accuracy: 1.0000000
step 56 [[0.3        0.65886329 0.35724794 0.5       ]
 [0.64034125 0.5        0.35724794 0.5881886 ]] 0.9622637280496275
step 56 params [[0.3        0.65886329 0.35724794 0.5       ]
 [0.64034125 0.5        0.35724794 0.5881886 ]] loss 0.9622637280496275 predictions [ter
Iter:    57 | Cost: 0.9954549 | Accuracy: 1.0000000
step 57 [[0.3        0.66271781 0.36222876 0.5       ]
 [0.64545364 0.5        0.36222876 0.5940169 ]] 0.955640357200405
step 57 params [[0.3        0.66271781 0.36222876 0.5       ]
 [0.64545364 0.5        0.36222876 0.5940169 ]] loss 0.955640357200405 predictions [tens
Iter:    58 | Cost: 0.9483878 | Accuracy: 1.0000000
step 58 [[0.3        0.6667507  0.36773832 0.5       ]
 [0.65097052 0.5        0.36773832 0.60044859]] 0.9483877737847256
step 58 params [[0.3        0.6667507  0.36773832 0.5       ]
 [0.65097052 0.5        0.36773832 0.60044859]] loss 0.9483877737847256 predictions [ter
Iter:    59 | Cost: 0.9406464 | Accuracy: 1.0000000
step 59 [[0.3        0.67007793 0.3720867  0.5       ]
 [0.6553967  0.5        0.3720867  0.60549359]] 0.494911919483788
step 59 params [[0.3        0.67007793 0.3720867  0.5       ]
 [0.6553967  0.5        0.3720867  0.60549359]] loss 0.494911919483788 predictions [tens
Iter:    60 | Cost: 0.9799005 | Accuracy: 1.0000000
step 60 [[0.3        0.6727641  0.37539103 0.5       ]
 [0.65883165 0.5        0.37539103 0.60928412]] 0.498801372795859
step 60 params [[0.3        0.6727641  0.37539103 0.5       ]
 [0.65883165 0.5        0.37539103 0.60928412]] loss 0.498801372795859 predictions [tens
Iter:    61 | Cost: 0.4726433 | Accuracy: 1.0000000
step 61 [[0.3        0.67435045 0.3786396  0.5       ]
 [0.66127877 0.5        0.3786396  0.6125692 ]] 0.4726432969719202
step 61 params [[0.3        0.67435045 0.3786396  0.5       ]
 [0.66127877 0.5        0.3786396  0.6125692 ]] loss 0.4726432969719202 predictions [ter
Iter:    62 | Cost: 0.9264132 | Accuracy: 1.0000000
step 62 [[0.3        0.67494325 0.3818512  0.5       ]
 [0.66282954 0.5        0.3818512  0.61539682]] 0.4739722002685836
step 62 params [[0.3        0.67494325 0.3818512  0.5       ]
 [0.66282954 0.5        0.3818512  0.61539682]] loss 0.4739722002685836 predictions [ter
Iter:    63 | Cost: 0.9236291 | Accuracy: 1.0000000
step 63 [[0.3        0.67463938 0.38504011 0.5       ]
 [0.66356765 0.5        0.38504011 0.61780976]] 0.47462423101503465
step 63 params [[0.3        0.67463938 0.38504011 0.5       ]
 [0.66356765 0.5        0.38504011 0.61780976]] loss 0.47462423101503465 predictions [te
Iter:    64 | Cost: 0.5073141 | Accuracy: 1.0000000
step 64 [[0.3        0.67573087 0.38747551 0.5       ]
 [0.66531866 0.5        0.38747551 0.62026163]] 0.9730999475576824
step 64 params [[0.3        0.67573087 0.38747551 0.5       ]
 [0.66531866 0.5        0.38747551 0.62026163]] loss 0.9730999475576824 predictions [ter
Iter:    65 | Cost: 0.5090252 | Accuracy: 1.0000000
step 65 [[0.3        0.67639443 0.38904525 0.5       ]
 [0.66632478 0.5        0.38904525 0.62168359]] 0.5090251822792382
step 65 params [[0.3        0.67639443 0.38904525 0.5       ]
 [0.66632478 0.5        0.38904525 0.62168359]] loss 0.5090251822792382 predictions [ter
Iter:    66 | Cost: 0.9707790 | Accuracy: 1.0000000
step 66 [[0.3        0.67833984 0.38998217 0.5       ]
 [0.66831306 0.5        0.38998217 0.62322849]] 0.970779029686789
step 66 params [[0.3        0.67833984 0.38998217 0.5       ]
 [0.66831306 0.5        0.38998217 0.62322849]] loss 0.970779029686789 predictions [tens
Iter:    67 | Cost: 0.4774014 | Accuracy: 1.0000000
step 67 [[0.3        0.67978086 0.39019573 0.5       ]
 [0.66953281 0.5        0.39019573 0.62382264]] 0.5114354653804272
step 67 params [[0.3        0.67978086 0.39019573 0.5       ]
 [0.66953281 0.5        0.39019573 0.62382264]] loss 0.5114354653804272 predictions [ter
```

```
Iter:    68 | Cost: 0.4783039 | Accuracy: 1.0000000
step 68 [[0.3        0.68157806 0.39133209 0.5       ]
 [0.67150479 0.5        0.39133209 0.62554707]] 0.9142790453001239
step 68 params [[0.3        0.68157806 0.39133209 0.5       ]
 [0.67150479 0.5        0.39133209 0.62554707]] loss 0.9142790453001239 predictions [ter
Iter:    69 | Cost: 0.9651079 | Accuracy: 1.0000000
step 69 [[0.3        0.68288019 0.39173257 0.5       ]
 [0.67270502 0.5        0.39173257 0.62630546]] 0.5136147553434671
step 69 params [[0.3        0.68288019 0.39173257 0.5       ]
 [0.67270502 0.5        0.39173257 0.62630546]] loss 0.5136147553434671 predictions [ter
Iter:    70 | Cost: 0.5143959 | Accuracy: 1.0000000
step 70 [[0.3        0.68455409 0.39302292 0.5       ]
 [0.67466088 0.5        0.39302292 0.62816778]] 0.9109591246302184
step 70 params [[0.3        0.68455409 0.39302292 0.5       ]
 [0.67466088 0.5        0.39302292 0.62816778]] loss 0.9109591246302184 predictions [ter
Iter:    71 | Cost: 0.5158758 | Accuracy: 1.0000000
step 71 [[0.3        0.68520662 0.39453445 0.5       ]
 [0.67575261 0.5        0.39453445 0.62971692]] 0.4815907323535994
step 71 params [[0.3        0.68520662 0.39453445 0.5       ]
 [0.67575261 0.5        0.39453445 0.62971692]] loss 0.4815907323535994 predictions [ter
Iter:    72 | Cost: 0.4821155 | Accuracy: 1.0000000
step 72 [[0.3        0.68546625 0.39527941 0.5       ]
 [0.67614916 0.5        0.39527941 0.63031268]] 0.5169216207795698
step 72 params [[0.3        0.68546625 0.39527941 0.5       ]
 [0.67614916 0.5        0.39527941 0.63031268]] loss 0.5169216207795698 predictions [ter
Iter:    73 | Cost: 0.9606691 | Accuracy: 1.0000000
step 73 [[0.3        0.68536692 0.39532792 0.5       ]
 [0.67591304 0.5        0.39532792 0.63004551]] 0.5173423369654132
step 73 params [[0.3        0.68536692 0.39532792 0.5       ]
 [0.67591304 0.5        0.39532792 0.63004551]] loss 0.5173423369654132 predictions [ter
Iter:    74 | Cost: 0.9068083 | Accuracy: 1.0000000
step 74 [[0.3        0.68493875 0.39474114 0.5       ]
 [0.67510053 0.5        0.39474114 0.62899668]] 0.5171975765172598
step 74 params [[0.3        0.68493875 0.39474114 0.5       ]
 [0.67510053 0.5        0.39474114 0.62899668]] loss 0.5171975765172598 predictions [ter
Iter:    75 | Cost: 0.9615354 | Accuracy: 1.0000000
step 75 [[0.3        0.68367212 0.3945614  0.5       ]
 [0.67367325 0.5        0.3945614  0.62790024]] 0.48176121421132456
step 75 params [[0.3        0.68367212 0.3945614  0.5       ]
 [0.67367325 0.5        0.3945614  0.62790024]] loss 0.48176121421132456 predictions [te
Iter:    76 | Cost: 0.9630437 | Accuracy: 1.0000000
step 76 [[0.3        0.68164826 0.39474796 0.5       ]
 [0.67168727 0.5        0.39474796 0.62675777]] 0.480828798945051
step 76 params [[0.3        0.68164826 0.39474796 0.5       ]
 [0.67168727 0.5        0.39474796 0.62675777]] loss 0.480828798945051 predictions [tens
Iter:    77 | Cost: 0.9653208 | Accuracy: 1.0000000
step 77 [[0.3        0.67894156 0.39526194 0.5       ]
 [0.66919428 0.5        0.39526194 0.62557014]] 0.479425500182578
step 77 params [[0.3        0.67894156 0.39526194 0.5       ]
 [0.66919428 0.5        0.39526194 0.62557014]] loss 0.479425500182578 predictions [tens
Iter:    78 | Cost: 0.9682937 | Accuracy: 1.0000000
step 78 [[0.3        0.67562007 0.39606629 0.5       ]
 [0.66624192 0.5        0.39606629 0.62433768]] 0.477601061943045
step 78 params [[0.3        0.67562007 0.39606629 0.5       ]
 [0.66624192 0.5        0.39606629 0.62433768]] loss 0.477601061943045 predictions [tens
Iter:    79 | Cost: 0.5117633 | Accuracy: 1.0000000
step 79 [[0.3        0.67174603 0.39712572 0.5       ]
 [0.66287407 0.5        0.39712572 0.62306022]] 0.475402608051898
step 79 params [[0.3        0.67174603 0.39712572 0.5       ]
 [0.66287407 0.5        0.39712572 0.62306022]] loss 0.475402608051898 predictions [tens
Iter:    80 | Cost: 0.5101344 | Accuracy: 1.0000000
step 80 [[0.3        0.66972448 0.39755761 0.5       ]
 [0.66099714 0.5        0.39755761 0.62217051]] 0.9760627976271151
step 80 params [[0.3        0.66972448 0.39755761 0.5       ]
 [0.66099714 0.5        0.39755761 0.62217051]] loss 0.9760627976271151 predictions [ter
Iter:    81 | Cost: 0.9783072 | Accuracy: 1.0000000
step 81 [[0.3        0.66703578 0.39826388 0.5       ]
 [0.65860607 0.5        0.39826388 0.62119643]] 0.4715199466716703
step 81 params [[0.3        0.66703578 0.39826388 0.5       ]
 [0.65860607 0.5        0.39826388 0.62119643]] loss 0.4715199466716703 predictions [ter
Iter:    82 | Cost: 0.5080050 | Accuracy: 1.0000000
step 82 [[0.3        0.66374647 0.39921243 0.5       ]
 [0.65574957 0.5        0.39921243 0.62014199]] 0.46975912581915624
step 82 params [[0.3        0.66374647 0.39921243 0.5       ]
 [0.65574957 0.5        0.39921243 0.62014199]] loss 0.46975912581915624 predictions [te
Iter:    83 | Cost: 0.4676377 | Accuracy: 1.0000000
step 83 [[0.3        0.66041751 0.39930944 0.5       ]
 [0.65255181 0.5        0.39930944 0.61832775]] 0.5066507012728911
step 83 params [[0.3        0.66041751 0.39930944 0.5       ]
 [0.65255181 0.5        0.39930944 0.61832775]] loss 0.5066507012728911 predictions [ter
Iter:    84 | Cost: 0.5048487 | Accuracy: 1.0000000
step 84 [[0.3        0.6579328  0.40056981 0.5       ]
 [0.65060323 0.5        0.40056981 0.61801049]] 0.9253236511049406
step 84 params [[0.3        0.6579328  0.40056981 0.5       ]
```

```
 [0.65060323 0.5        0.40056981 0.61801049]] loss 0.9253236511049406 predictions [ter
Iter:    85 | Cost: 0.9264432 | Accuracy: 1.0000000
step 85 [[0.3        0.65532901 0.4009263  0.5       ]
 [0.64822568 0.5        0.4009263  0.61686646]] 0.5041141026534404
step 85 params [[0.3        0.65532901 0.4009263  0.5       ]
 [0.64822568 0.5        0.4009263  0.61686646]] loss 0.5041141026534404 predictions [ter
Iter:    86 | Cost: 0.4622021 | Accuracy: 1.0000000
step 86 [[0.3        0.65351102 0.40245254 0.5       ]
 [0.64702334 0.5        0.40245254 0.61714696]] 0.9283493424377195
step 86 params [[0.3        0.65351102 0.40245254 0.5       ]
 [0.64702334 0.5        0.40245254 0.61714696]] loss 0.9283493424377195 predictions [ter
Iter:    87 | Cost: 0.9286929 | Accuracy: 1.0000000
step 87 [[0.3        0.65099854 0.40410306 0.5       ]
 [0.64524367 0.5        0.40410306 0.61722032]] 0.4611549647572211
step 87 params [[0.3        0.65099854 0.40410306 0.5       ]
 [0.64524367 0.5        0.40410306 0.61722032]] loss 0.4611549647572211 predictions [ter
Iter:    88 | Cost: 0.9294657 | Accuracy: 1.0000000
step 88 [[0.3        0.64926587 0.4068157  0.5       ]
 [0.64458139 0.5        0.4068157  0.61859007]] 0.9294656557827844
step 88 params [[0.3        0.64926587 0.4068157  0.5       ]
 [0.64458139 0.5        0.4068157  0.61859007]] loss 0.9294656557827844 predictions [ter
Iter:    89 | Cost: 0.5025874 | Accuracy: 1.0000000
step 89 [[0.3        0.6468324  0.40953749 0.5       ]
 [0.64329414 0.5        0.40953749 0.61965088]] 0.45884074601224045
step 89 params [[0.3        0.6468324  0.40953749 0.5       ]
 [0.64329414 0.5        0.40953749 0.61965088]] loss 0.45884074601224045 predictions [te
Iter:    90 | Cost: 1.0019155 | Accuracy: 1.0000000
step 90 [[0.3        0.64377054 0.41226684 0.5       ]
 [0.64144369 0.5        0.41226684 0.62042978]] 0.4575620546838315
step 90 params [[0.3        0.64377054 0.41226684 0.5       ]
 [0.64144369 0.5        0.41226684 0.62042978]] loss 0.4575620546838315 predictions [ter
Iter:    91 | Cost: 0.5025076 | Accuracy: 1.0000000
step 91 [[0.3        0.64153655 0.41598495 0.5       ]
 [0.64071301 0.5        0.41598495 0.62243057]] 0.9288989510186488
step 91 params [[0.3        0.64153655 0.41598495 0.5       ]
 [0.64071301 0.5        0.41598495 0.62243057]] loss 0.9288989510186488 predictions [ter
Iter:    92 | Cost: 0.9277743 | Accuracy: 1.0000000
step 92 [[0.3        0.64004992 0.4205862  0.5       ]
 [0.64098469 0.5        0.4205862  0.62550987]] 0.927774264528005
step 92 params [[0.3        0.64004992 0.4205862  0.5       ]
 [0.64098469 0.5        0.4205862  0.62550987]] loss 0.927774264528005 predictions [tens
Iter:    93 | Cost: 0.9254067 | Accuracy: 1.0000000
step 93 [[0.3        0.63835286 0.42390314 0.5       ]
 [0.64062348 0.5        0.42390314 0.62744783]] 0.5047941694077852
step 93 params [[0.3        0.63835286 0.42390314 0.5       ]
 [0.64062348 0.5        0.42390314 0.62744783]] loss 0.5047941694077852 predictions [ter
Iter:    94 | Cost: 0.9241655 | Accuracy: 1.0000000
step 94 [[0.3        0.63832089 0.42644888 0.5       ]
 [0.64146142 0.5        0.42644888 0.62951603]] 1.008662102596171
step 94 params [[0.3        0.63832089 0.42644888 0.5       ]
 [0.64146142 0.5        0.42644888 0.62951603]] loss 1.008662102596171 predictions [tens
Iter:    95 | Cost: 0.4538676 | Accuracy: 1.0000000
step 95 [[0.3        0.63975267 0.42828181 0.5       ]
 [0.64336662 0.5        0.42828181 0.63170003]] 1.008315147857118
step 95 params [[0.3        0.63975267 0.42828181 0.5       ]
 [0.64336662 0.5        0.42828181 0.63170003]] loss 1.008315147857118 predictions [tens
Iter:    96 | Cost: 0.9198864 | Accuracy: 1.0000000
step 96 [[0.3        0.64071169 0.42909671 0.5       ]
 [0.64449082 0.5        0.42909671 0.63282382]] 0.5084356970610406
step 96 params [[0.3        0.64071169 0.42909671 0.5       ]
 [0.64449082 0.5        0.42909671 0.63282382]] loss 0.5084356970610406 predictions [ter
Iter:    97 | Cost: 0.4555151 | Accuracy: 1.0000000
step 97 [[0.3        0.6430165  0.42933104 0.5       ]
 [0.64664836 0.5        0.42933104 0.63414386]] 1.0054532498287407
step 97 params [[0.3        0.6430165  0.42933104 0.5       ]
 [0.64664836 0.5        0.42933104 0.63414386]] loss 1.0054532498287407 predictions [ter
Iter:    98 | Cost: 0.5105367 | Accuracy: 1.0000000
step 98 [[0.3        0.64476859 0.42871126 0.5       ]
 [0.64799964 0.5        0.42871126 0.63448119]] 0.5105367155985059
step 98 params [[0.3        0.64476859 0.42871126 0.5       ]
 [0.64799964 0.5        0.42871126 0.63448119]] loss 0.5105367155985059 predictions [ter
Iter:    99 | Cost: 1.0012917 | Accuracy: 1.0000000
step 99 [[0.3        0.64686342 0.4293873  0.5       ]
 [0.65012041 0.5        0.4293873  0.63605705]] 0.9158978239444003
step 99 params [[0.3        0.64686342 0.4293873  0.5       ]
 [0.65012041 0.5        0.4293873  0.63605705]] loss 0.9158978239444003 predictions [ter
Iter:   100 | Cost: 0.5124395 | Accuracy: 1.0000000
'state_preparation(x)\n\nlosses = []\nfor i in range(400):\n  params, loss = optimizer.
step_and_cost(circuit, params)\n  losses.append(loss)\n  print(loss)'
```

```
import matplotlib.pyplot as plt

plt.plot(losses, label='Loss function')
plt.xlabel('Iteration')
plt.ylabel('Loss')
plt.legend()
```

```
<matplotlib.legend.Legend at 0x78d61c90ea70>
```