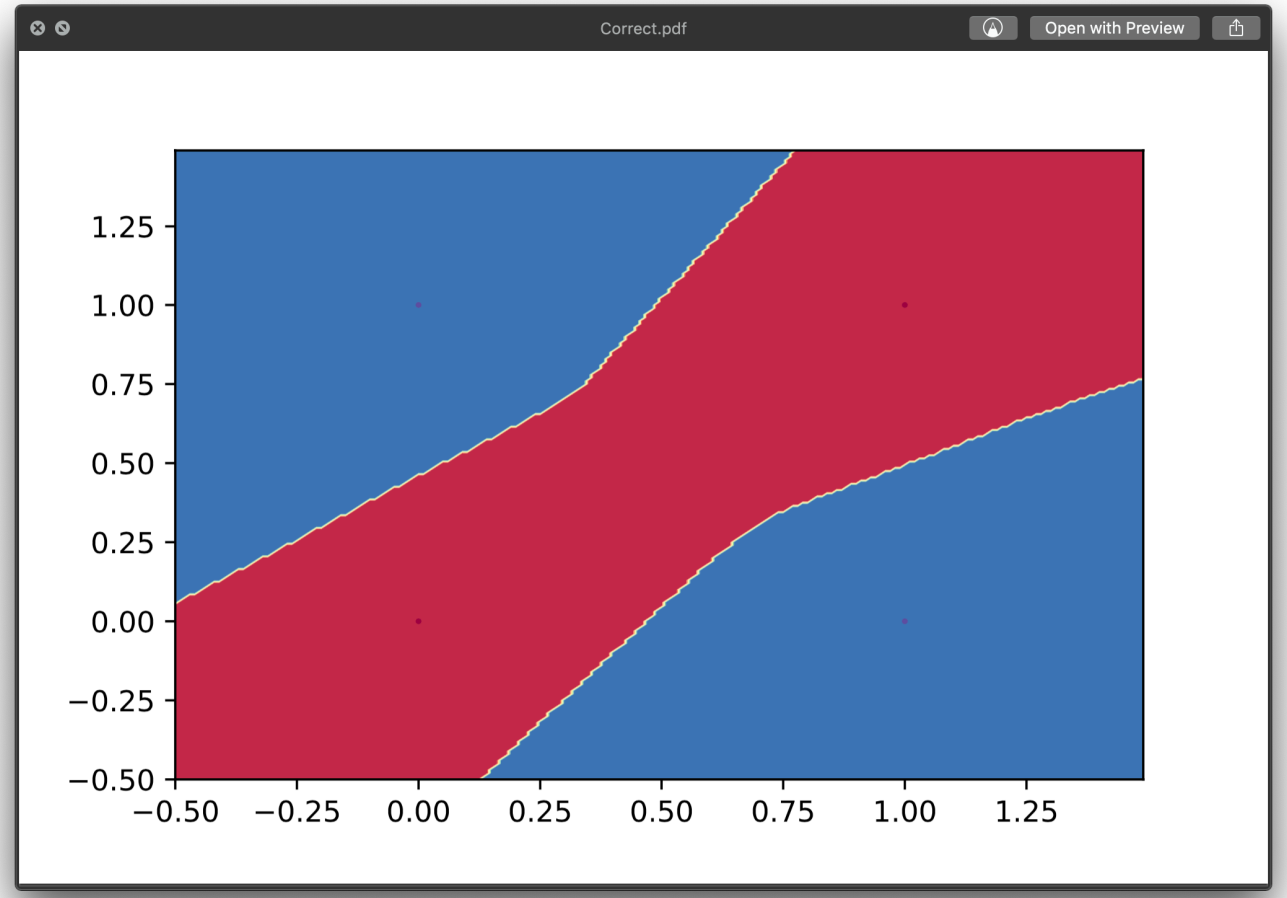
In [1]:
```
#Question 1
```

In [2]:
```
import torch
import torch.nn as nn
import torch.nn.functional as F
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

In [3]:
```
class Net(nn.Module):

    def __init__(self):
        super(Net, self).__init__()
        self.fc1 = nn.Linear(2, 20)
        self.fc2 = nn.Linear(20, 20)
        self.fc3 = nn.Linear(20, 20)
        self.fc4 = nn.Linear(20, 20)
        self.fc5 = nn.Linear(20, 20)
        self.fc6 = nn.Linear(20, 20)
        self.fc7 = nn.Linear(20, 2)

    def forward(self, x):
        x = F.relu(self.fc1(x))
        x = F.relu(self.fc2(x))
        x = F.relu(self.fc3(x))
        x = F.relu(self.fc4(x))
        x = F.relu(self.fc5(x))
        x = F.relu(self.fc6(x))
        x = self.fc7(x)
        return F.log_softmax(x)
        #return F.softmax(x)
```


Correct.pdf
Open with Preview
1.25
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00
1.25

In [1]:
```
#Question 1
```

In [2]:
```
import torch
import torch.nn as nn
import torch.nn.functional as F
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

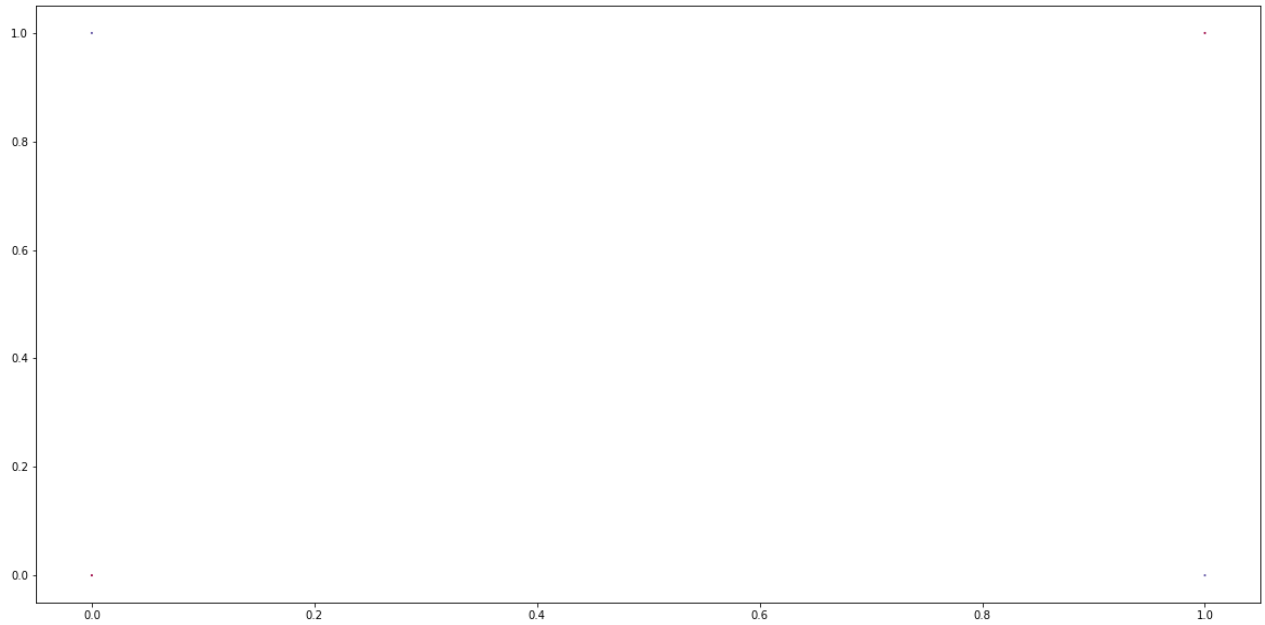
In [3]:
```
class Net(nn.Module):

    def __init__(self):
        super(Net, self).__init__()
        self.fc1 = nn.Linear(2, 20)
        self.fc2 = nn.Linear(20, 20)
        self.fc3 = nn.Linear(20, 20)
        self.fc4 = nn.Linear(20, 20)
        self.fc5 = nn.Linear(20, 20)
        self.fc6 = nn.Linear(20, 20)
        self.fc7 = nn.Linear(20, 2)

    def forward(self, x):
        x = F.relu(self.fc1(x))
        x = F.relu(self.fc2(x))
        x = F.relu(self.fc3(x))
        x = F.relu(self.fc4(x))
        x = F.relu(self.fc5(x))
        x = F.relu(self.fc6(x))
        x = self.fc7(x)
        return F.log_softmax(x)
        #return F.softmax(x)
```

In [4]:

```
#%% plot function

def plot_data(X, y, filename):
    plt.scatter(X[:, 0], X[:, 1], c=y, cmap=plt.cm.Spectral, s = 1)
    plt.savefig(filename)
    plt.close()

def plot_decision_boundary(clf, X, y, filename):
    # Set min and max values and give it some padding
    #x_min, x_max = X[:, 0].min() - .1, X[:, 0].max() + .1
    #y_min, y_max = X[:, 1].min() - .1, X[:, 1].max() + .1
    x_min, x_max = -0.5, 1.5
    y_min, y_max = -0.5, 1.5
    h = 0.01
    # Generate a grid of points with distance h between them
    xx, yy = np.meshgrid(np.arange(x_min, x_max, h), np.arange(y_min, y_ma
    # Predict the function value for the whole gid
    #Z = clf.predict(np.c_[xx.ravel(), yy.ravel()])
    X_out = net(torch.tensor(np.c_[xx.ravel(), yy.ravel()], dtype = torch.
    Z = X_out.data.max(1)[1]
    # Z.shape
    Z = Z.reshape(xx.shape)
    # Plot the contour and training examples
    plt.contourf(xx, yy, Z, cmap=plt.cm.Spectral)
    plt.scatter(X[:, 0], X[:, 1], c=y, cmap=plt.cm.Spectral, s = 1)
    plt.savefig(filename)
    plt.close()
```

In [5]:

```
#%% read data

data = pd.read_csv("XOR.csv")
X = data.values[:, 0:2]  # Take only the first two features.
X = torch.tensor(X, dtype = torch.float)
y = data.values[:, 2]
y = torch.tensor(y, dtype = torch.long)

plot_data(X,y,'data.pdf')
plt.figure(figsize=(20,10))
plt.scatter(X[:, 0], X[:, 1], c=y, cmap=plt.cm.Spectral, s = 1)
plt.show()
```



In [6]:
```
data
```

Out[6]:

	Input A	Input B	Output
0	0	0	0
1	0	1	1
2	1	0	1
3	1	1	0

In [7]:
```
#%% train
net = Net()

# create a stochastic gradient descent optimizer
learning_rate = .01
optimizer = torch.optim.SGD(net.parameters(), lr=learning_rate, momentum=0

# create a loss function
#criterion = nn.CrossEntropyLoss()
criterion = nn.NLLLoss()
```

In [8]:

```
# nepochs = 10000
nepochs = 3000 #10000
data, target = X, y
# run the main training loop
for epoch in range(nepochs):
#    adjust learning rate if desired
#    if epoch % 3000 == 0 and epoch <= 24000:
#        for g in optimizer.param_groups:
#            g['lr'] = g['lr']/2
    optimizer.zero_grad()
    # forward propagate
    net_out = net(data)
    # compute loss
    loss = criterion(net_out, target)

    # backpropagate
    loss.backward()
    # update parameters
    optimizer.step()
    # print out report

    if epoch % 10 == 0:
        print('Epoch ', epoch, 'Loss ', loss.item())
        net_out = net(data)
        pred = net_out.data.max(1)[1]  # get the index of the max log-prob
        correctidx = pred.eq(target.data)
        ncorrect = correctidx.sum()
        accuracy = ncorrect.item()/len(data)
        if accuracy== 1:
            break
        print('Training accuracy is ', accuracy)
```

```
<ipython-input-3-b68d425386b7>:21: UserWarning: Implicit dimension choice f
or log_softmax has been deprecated. Change the call to include dim=X as an
argument.
  return F.log_softmax(x)
Epoch  0 Loss  0.6941758394241333
Training accuracy is  0.5
Epoch  10 Loss  0.6936168670654297
Training accuracy is  0.5
Epoch  20 Loss  0.6931634545326233
Training accuracy is  0.5
Epoch  30 Loss  0.6930881142616272
Training accuracy is  0.5
Epoch  40 Loss  0.6930862665176392
Training accuracy is  0.5
Epoch  50 Loss  0.6930666565895081
Training accuracy is  0.5
Epoch  60 Loss  0.6930466890335083
Training accuracy is  0.5
Epoch  70 Loss  0.6930317282676697
Training accuracy is  0.75
Epoch  80 Loss  0.6930174827575684
Training accuracy is  0.5
Epoch  90 Loss  0.6930090188980103
Training accuracy is  0.5
Epoch  100 Loss  0.6930021047592163
```

```
Training accuracy is  0.75
Epoch  110 Loss  0.6929948925971985
Training accuracy is  0.75
Epoch  120 Loss  0.6929873824119568
Training accuracy is  0.75
Epoch  130 Loss  0.6929792761802673
Training accuracy is  0.75
Epoch  140 Loss  0.6929664015769958
Training accuracy is  0.75
Epoch  150 Loss  0.6929500102996826
Training accuracy is  0.75
Epoch  160 Loss  0.6929318904876709
Training accuracy is  0.75
Epoch  170 Loss  0.692912757396698
Training accuracy is  0.75
Epoch  180 Loss  0.6928925514221191
Training accuracy is  0.75
Epoch  190 Loss  0.6928713321685791
Training accuracy is  0.75
Epoch  200 Loss  0.6928490996360779
Training accuracy is  0.75
Epoch  210 Loss  0.692834734916687
Training accuracy is  0.75
Epoch  220 Loss  0.69282066822052
Training accuracy is  0.75
Epoch  230 Loss  0.692805826663971
Training accuracy is  0.75
Epoch  240 Loss  0.692790150642395
Training accuracy is  0.75
Epoch  250 Loss  0.6927738189697266
Training accuracy is  0.75
Epoch  260 Loss  0.6927566528320312
Training accuracy is  0.75
Epoch  270 Loss  0.6927385330200195
Training accuracy is  0.75
Epoch  280 Loss  0.6927194595336914
Training accuracy is  0.75
Epoch  290 Loss  0.6926991939544678
Training accuracy is  0.75
Epoch  300 Loss  0.6926789283752441
Training accuracy is  0.75
Epoch  310 Loss  0.6926573514938354
Training accuracy is  0.75
Epoch  320 Loss  0.6926344633102417
Training accuracy is  0.75
Epoch  330 Loss  0.6926104426383972
Training accuracy is  0.75
Epoch  340 Loss  0.6925852298736572
Training accuracy is  0.75
Epoch  350 Loss  0.6925604343414307
Training accuracy is  0.75
Epoch  360 Loss  0.692534327507019
Training accuracy is  0.75
Epoch  370 Loss  0.6925068497657776
Training accuracy is  0.75
Epoch  380 Loss  0.6924779415130615
Training accuracy is  0.75
Epoch  390 Loss  0.692447304725647
Training accuracy is  0.75
Epoch  400 Loss  0.6924149990081787
Training accuracy is  0.75
```

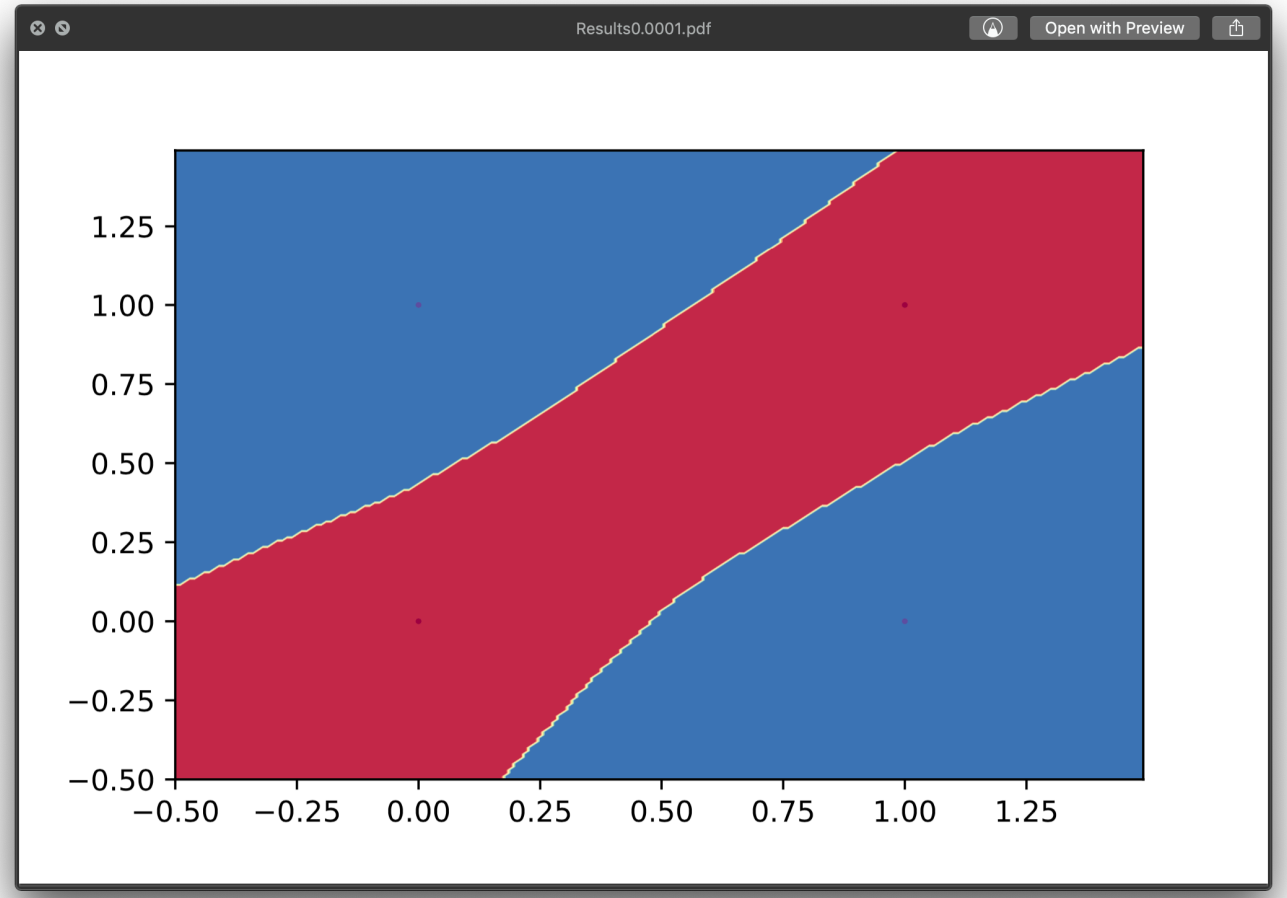
```
Epoch  410 Loss  0.6923807263374329
Training accuracy is  0.75
Epoch  420 Loss  0.6923444271087646
Training accuracy is  0.75
Epoch  430 Loss  0.6923058032989502
Training accuracy is  0.75
Epoch  440 Loss  0.6922647356987
Training accuracy is  0.75
Epoch  450 Loss  0.6922210454940796
Training accuracy is  0.75
Epoch  460 Loss  0.6921743154525757
Training accuracy is  0.75
Epoch  470 Loss  0.6921243667602539
Training accuracy is  0.75
Epoch  480 Loss  0.6920709609985352
Training accuracy is  0.75
Epoch  490 Loss  0.6920138001441956
Training accuracy is  0.75
Epoch  500 Loss  0.6919522881507874
Training accuracy is  0.75
Epoch  510 Loss  0.6918848752975464
Training accuracy is  0.75
Epoch  520 Loss  0.6918103098869324
Training accuracy is  0.75
Epoch  530 Loss  0.6917284727096558
Training accuracy is  0.75
Epoch  540 Loss  0.6916390657424927
Training accuracy is  0.75
Epoch  550 Loss  0.691541314125061
Training accuracy is  0.75
Epoch  560 Loss  0.6914339065551758
Training accuracy is  0.5
Epoch  570 Loss  0.691317081451416
Training accuracy is  0.5
Epoch  580 Loss  0.6911896467208862
Training accuracy is  0.5
Epoch  590 Loss  0.6910485029220581
Training accuracy is  0.5
Epoch  600 Loss  0.6908772587776184
Training accuracy is  0.5
Epoch  610 Loss  0.6906797289848328
Training accuracy is  0.5
Epoch  620 Loss  0.690450131893158
Training accuracy is  0.5
Epoch  630 Loss  0.6902066469192505
Training accuracy is  0.5
Epoch  640 Loss  0.6899532675743103
Training accuracy is  0.5
Epoch  650 Loss  0.6896663308143616
Training accuracy is  0.5
Epoch  660 Loss  0.6893412470817566
Training accuracy is  0.5
Epoch  670 Loss  0.6889669895172119
Training accuracy is  0.75
Epoch  680 Loss  0.6885308027267456
Training accuracy is  0.5
Epoch  690 Loss  0.6880131959915161
Training accuracy is  0.75
Epoch  700 Loss  0.6873992681503296
Training accuracy is  0.75
Epoch  710 Loss  0.6866694688796997
```

```
Training accuracy is  0.75
Epoch  720 Loss  0.6856508851051331
Training accuracy is  0.75
Epoch  730 Loss  0.6842676401138306
Training accuracy is  0.75
Epoch  740 Loss  0.6827037930488586
Training accuracy is  0.75
Epoch  750 Loss  0.6807572841644287
Training accuracy is  0.75
Epoch  760 Loss  0.6783814430236816
Training accuracy is  0.75
Epoch  770 Loss  0.6752572059631348
Training accuracy is  0.75
Epoch  780 Loss  0.6711351871490479
Training accuracy is  0.75
Epoch  790 Loss  0.6654714345932007
Training accuracy is  0.75
Epoch  800 Loss  0.6570992469787598
Training accuracy is  0.75
Epoch  810 Loss  0.6444829702377319
Training accuracy is  0.75
Epoch  820 Loss  0.6250220537185669
```

In [9]:

```
#%% compute accuracy on training data

net_out = net(data)
pred = net_out.data.max(1)[1]  # get the index of the max log-probability
correctidx = pred.eq(target.data)
ncorrect = correctidx.sum()
accuracy = ncorrect.item()/len(data)
print('Training accuracy is ', accuracy)
plt.scatter(X[:, 0], X[:, 1], c=pred, cmap=plt.cm.Spectral, s = 1)

#%% if need to train further

# for epoch in range(nepochs):
#     # resize data from (batch_size, 1, 28, 28) to (batch_size, 28*28)
#     optimizer.zero_grad()
#     net_out = net(data)
#     loss = criterion(net_out, target)
#     loss.backward()
#     optimizer.step()
#     if epoch % 100 == 0:
#         print('Epoch ', epoch, 'Loss ', loss.item())


#%%  plot outputs
plot_decision_boundary(net, X, y, 'Results0.0001.pdf')
#plot_decision_boundary(net, X[correctidx,:], y[correctidx], 'Correct.pdf'
#plot_decision_boundary(net, X[~correctidx,:], y[~correctidx], 'Inorrect.p

#%% save model
```

```
Training accuracy is  1.0
<ipython-input-3-b68d425386b7>:21: UserWarning: Implicit dimension choice f
or log_softmax has been deprecated. Change the call to include dim=X as an
argument.
  return F.log_softmax(x)
```

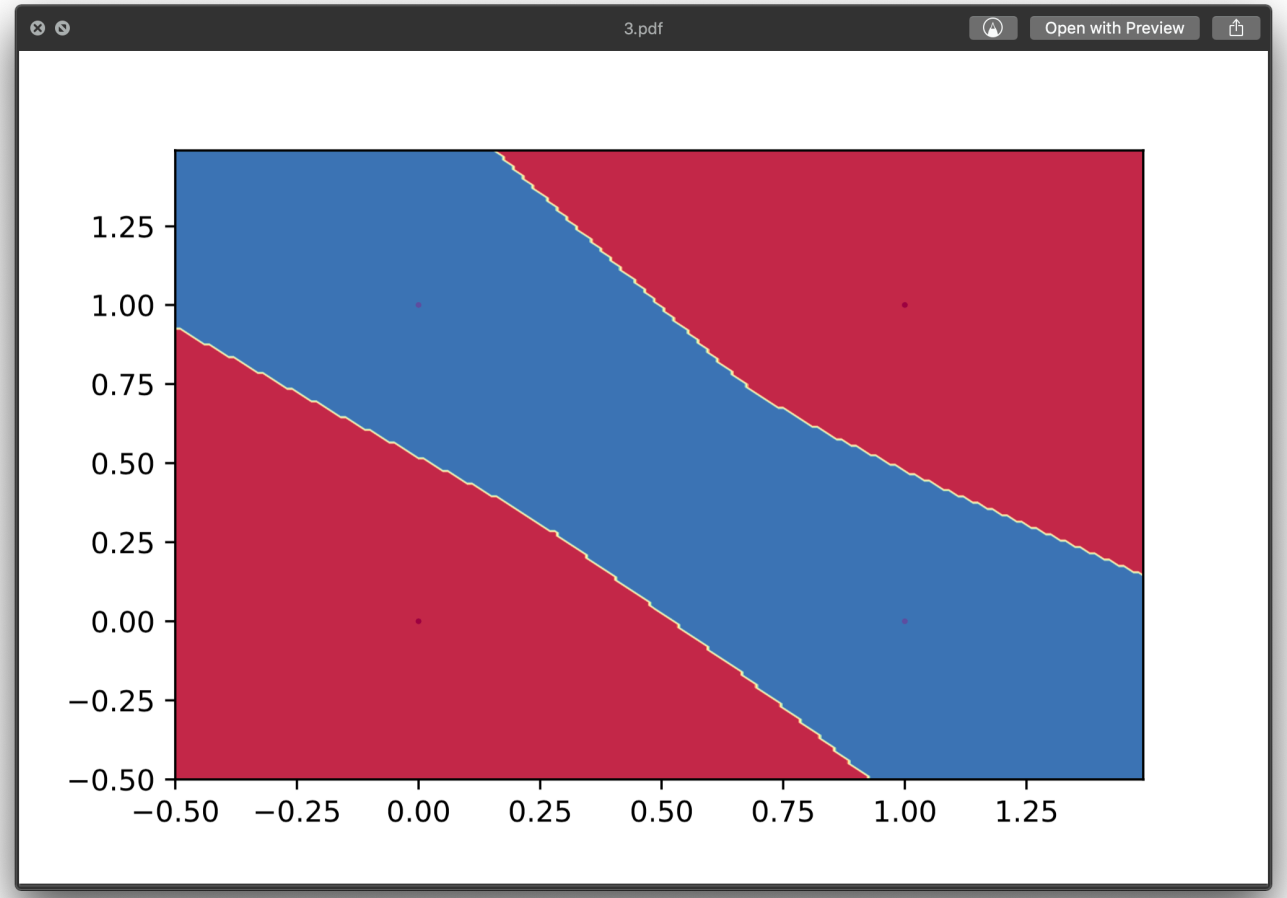

Results0.0001.pdf
Open with Preview
1.25
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00
1.25

In [10]:
```
#Question 2:
```

In [11]:
```
# nepochs = 10000
nepochs = 3000 #10000
data, target = X, y
# run the main training loop
for epoch in range(nepochs):
#    adjust learning rate if desired
#    if epoch % 3000 == 0 and epoch <= 24000:
#        for g in optimizer.param_groups:
#            g['lr'] = g['lr']/2
    optimizer.zero_grad()
    # forward propagate
    net_out = net(data)
    # compute loss
    loss = criterion(net_out, target)

    # backpropagate
    loss.backward()
    # update parameters
    optimizer.step()
    # print out report

    if epoch % 10 == 0:
        print('Epoch ', epoch, 'Loss ', loss.item())
        net_out = net(data)
        pred = net_out.data.max(1)[1]  # get the index of the max log-prob
        correctidx = pred.eq(target.data)
        ncorrect = correctidx.sum()
        accuracy = ncorrect.item()/len(data)
        if epoch< 0.001:
            break
        print('Training accuracy is ', accuracy)
```

```
Epoch  0 Loss  0.6225759387016296
<ipython-input-3-b68d425386b7>:21: UserWarning: Implicit dimension choice f
or log_softmax has been deprecated. Change the call to include dim=X as an
argument.
  return F.log_softmax(x)
```

In [12]:
```
#Question 3
```


3.pdf
Open with Preview
1.25
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00
1.25

In [13]:

```
class Net(nn.Module):

    def __init__(self):
        super(Net, self).__init__()
        self.fc1 = nn.Linear(2, 20)
        self.fc2 = nn.Linear(20, 20)
        self.fc3 = nn.Linear(20, 20)
        #self.fc4 = nn.Linear(20, 20)
        #self.fc5 = nn.Linear(20, 20)
        #self.fc6 = nn.Linear(20, 20)
        #self.fc7 = nn.Linear(20, 2)

    def forward(self, x):
        x = F.relu(self.fc1(x))
        x = F.relu(self.fc2(x))
        #x = F.relu(self.fc3(x))
        #x = F.relu(self.fc4(x))
        #x = F.relu(self.fc5(x))
        #x = F.relu(self.fc6(x))
        x = self.fc3(x)
        return F.log_softmax(x)
        #return F.softmax(x)
```

In [14]:

```
# nepochs = 10000
nepochs = 3000 #10000
data, target = X, y
# run the main training loop
for epoch in range(nepochs):
#    adjust learning rate if desired
#    if epoch % 3000 == 0 and epoch <= 24000:
#        for g in optimizer.param_groups:
#            g['lr'] = g['lr']/2
    optimizer.zero_grad()
    # forward propagate
    net_out = net(data)
    # compute loss
    loss = criterion(net_out, target)

    # backpropagate
    loss.backward()
    # update parameters
    optimizer.step()
    # print out report

    if epoch % 10 == 0:
        print('Epoch ', epoch, 'Loss ', loss.item())
        net_out = net(data)
        pred = net_out.data.max(1)[1]  # get the index of the max log-prob
        correctidx = pred.eq(target.data)
        ncorrect = correctidx.sum()
        accuracy = ncorrect.item()/len(data)
        #if epochs< 0.001:
         #   break
        print('Training accuracy is ', accuracy)
```

```
<ipython-input-3-b68d425386b7>:21: UserWarning: Implicit dimension choice f
or log_softmax has been deprecated. Change the call to include dim=X as an
argument.
  return F.log_softmax(x)
Epoch  0 Loss  0.6200512647628784
Training accuracy is  1.0
Epoch  10 Loss  0.5861670970916748
Training accuracy is  1.0
Epoch  20 Loss  0.5298529863357544
Training accuracy is  1.0
Epoch  30 Loss  0.43002861738204956
Training accuracy is  1.0
Epoch  40 Loss  0.27282190322875977
Training accuracy is  1.0
Epoch  50 Loss  0.13154783844947815
Training accuracy is  1.0
Epoch  60 Loss  0.05327668413519859
Training accuracy is  1.0
Epoch  70 Loss  0.023709189146757126
Training accuracy is  1.0
Epoch  80 Loss  0.013243664987385273
Training accuracy is  1.0
Epoch  90 Loss  0.008865930140018463
Training accuracy is  1.0
Epoch  100 Loss  0.006658598780632019
```

```
Training accuracy is  1.0
Epoch  110 Loss  0.005379956215620041
Training accuracy is  1.0
Epoch  120 Loss  0.0045394496992230415
Training accuracy is  1.0
Epoch  130 Loss  0.003928876481950283
Training accuracy is  1.0
Epoch  140 Loss  0.00346049340441823
Training accuracy is  1.0
Epoch  150 Loss  0.0030872030183672905
Training accuracy is  1.0
Epoch  160 Loss  0.0027822465635836124
Training accuracy is  1.0
Epoch  170 Loss  0.0025281780399382114
Training accuracy is  1.0
Epoch  180 Loss  0.0023130846675485373
Training accuracy is  1.0
Epoch  190 Loss  0.0021286052651703358
Training accuracy is  1.0
Epoch  200 Loss  0.001969016157090664
Training accuracy is  1.0
Epoch  210 Loss  0.0018297492060810328
Training accuracy is  1.0
Epoch  220 Loss  0.001707063871435821
Training accuracy is  1.0
Epoch  230 Loss  0.0015983477933332324
Training accuracy is  1.0
Epoch  240 Loss  0.0015014930395409465
Training accuracy is  1.0
Epoch  250 Loss  0.0014145683962851763
Training accuracy is  1.0
Epoch  260 Loss  0.0013362973695620894
Training accuracy is  1.0
Epoch  270 Loss  0.0012653418816626072
Training accuracy is  1.0
Epoch  280 Loss  0.001200811006128788
Training accuracy is  1.0
Epoch  290 Loss  0.0011419614311307669
Training accuracy is  1.0
Epoch  300 Loss  0.001088079996407032
Training accuracy is  1.0
Epoch  310 Loss  0.0010385422501713037
Training accuracy is  1.0
Epoch  320 Loss  0.000992931891232729
Training accuracy is  1.0
Epoch  330 Loss  0.0009508328512310982
Training accuracy is  1.0
Epoch  340 Loss  0.0009117689915001392
Training accuracy is  1.0
Epoch  350 Loss  0.0008755025337450206
Training accuracy is  1.0
Epoch  360 Loss  0.0008417657809332013
Training accuracy is  1.0
Epoch  370 Loss  0.000810201745480299
Training accuracy is  1.0
Epoch  380 Loss  0.0007807806832715869
Training accuracy is  1.0
Epoch  390 Loss  0.0007531456649303436
Training accuracy is  1.0
Epoch  400 Loss  0.0007272965158335865
Training accuracy is  1.0
```

```
Epoch  410 Loss  0.0007029358530417085
Training accuracy is  1.0
Epoch  420 Loss  0.0006800339324399829
Training accuracy is  1.0
Epoch  430 Loss  0.0006584418006241322
Training accuracy is  1.0
Epoch  440 Loss  0.0006379810511134565
Training accuracy is  1.0
Epoch  450 Loss  0.0006187112303450704
Training accuracy is  1.0
Epoch  460 Loss  0.000600394036155194
Training accuracy is  1.0
Epoch  470 Loss  0.0005830595036968589
Training accuracy is  1.0
Epoch  480 Loss  0.0005665883654728532
Training accuracy is  1.0
Epoch  490 Loss  0.0005509211914613843
Training accuracy is  1.0
Epoch  500 Loss  0.0005360282375477254
Training accuracy is  1.0
Epoch  510 Loss  0.000521879643201828
Training accuracy is  1.0
Epoch  520 Loss  0.0005083564901724458
Training accuracy is  1.0
Epoch  530 Loss  0.0004953990573994815
Training accuracy is  1.0
Epoch  540 Loss  0.0004830968100577593
Training accuracy is  1.0
Epoch  550 Loss  0.000471300765639171
Training accuracy is  1.0
Epoch  560 Loss  0.00045998120913282037
Training accuracy is  1.0
Epoch  570 Loss  0.00044919774518348277
Training accuracy is  1.0
Epoch  580 Loss  0.0004388609668239951
Training accuracy is  1.0
Epoch  590 Loss  0.00042888158350251615
Training accuracy is  1.0
Epoch  600 Loss  0.00041931914165616035
Training accuracy is  1.0
Epoch  610 Loss  0.0004101736412849277
Training accuracy is  1.0
Epoch  620 Loss  0.00040135576273314655
Training accuracy is  1.0
Epoch  630 Loss  0.0003928953083232045
Training accuracy is  1.0
Epoch  640 Loss  0.000384702900191769
Training accuracy is  1.0
Epoch  650 Loss  0.0003768083406612277
Training accuracy is  1.0
Epoch  660 Loss  0.00036927120527252555
Training accuracy is  1.0
Epoch  670 Loss  0.0003619723138399422
Training accuracy is  1.0
Epoch  680 Loss  0.0003549117245711386
Training accuracy is  1.0
Epoch  690 Loss  0.000348119210684672
Training accuracy is  1.0
Epoch  700 Loss  0.0003415053943172097
Training accuracy is  1.0
Epoch  710 Loss  0.0003351596824359149
```

```
Training accuracy is  1.0
Epoch  720 Loss  0.00032902247039601207
Training accuracy is  1.0
Epoch  730 Loss  0.0003230639558751136
Training accuracy is  1.0
Epoch  740 Loss  0.00031734377262182534
Training accuracy is  1.0
Epoch  750 Loss  0.0003117427113465965
Training accuracy is  1.0
Epoch  760 Loss  0.0003063501790165901
Training accuracy is  1.0
Epoch  770 Loss  0.00030113637330941856
Training accuracy is  1.0
Epoch  780 Loss  0.00029604171868413687
Training accuracy is  1.0
Epoch  790 Loss  0.00029112579068169
Training accuracy is  1.0
Epoch  800 Loss  0.00028635881608352065
Training accuracy is  1.0
Epoch  810 Loss  0.00028171109925672412
Training accuracy is  1.0
Epoch  820 Loss  0.00027718232013285 16
Training accuracy is  1.0
Epoch  830 Loss  0.00027280260110273957
Training accuracy is  1.0
Epoch  840 Loss  0.00026852718354769051
Training accuracy is  1.0
Epoch  850 Loss  0.00026443041861057 28
Training accuracy is  1.0
Epoch  860 Loss  0.00026040815282613 04
Training accuracy is  1.0
Epoch  870 Loss  0.00025653489865362644
Training accuracy is  1.0
Epoch  880 Loss  0.00025272119091823697
Training accuracy is  1.0
Epoch  890 Loss  0.00024899683194234967
Training accuracy is  1.0
Epoch  900 Loss  0.00024533207761123776
Training accuracy is  1.0
Epoch  910 Loss  0.00024184602079913 02
Training accuracy is  1.0
Epoch  920 Loss  0.00023844940005801618
Training accuracy is  1.0
Epoch  930 Loss  0.00023511234030593187
Training accuracy is  1.0
Epoch  940 Loss  0.00023183487064670771
Training accuracy is  1.0
Epoch  950 Loss  0.00022867656662128866
Training accuracy is  1.0
Epoch  960 Loss  0.00022555480212625116
Training accuracy is  1.0
Epoch  970 Loss  0.00022250888287089765
Training accuracy is  1.0
Epoch  980 Loss  0.00021955910770779 11
Training accuracy is  1.0
Epoch  990 Loss  0.00021669872512575 24
Training accuracy is  1.0
Epoch  1000 Loss  0.00021389791800174862
Training accuracy is  1.0
Epoch  1010 Loss  0.00021115668641868 98
Training accuracy is  1.0
```

```
Epoch  1020 Loss  0.00020847504492849112
Training accuracy is  1.0
Epoch  1030 Loss  0.0002058529935311526
Training accuracy is  1.0
Epoch  1040 Loss  0.00020326071535237134
Training accuracy is  1.0
Epoch  1050 Loss  0.00020075784414075315
Training accuracy is  1.0
Epoch  1060 Loss  0.0001982847461476922
Training accuracy is  1.0
Epoch  1070 Loss  0.00019590104056987911
Training accuracy is  1.0
Epoch  1080 Loss  0.0001935471227625385
Training accuracy is  1.0
Epoch  1090 Loss  0.00019125279504805803
Training accuracy is  1.0
Epoch  1100 Loss  0.0001890180428745225
Training accuracy is  1.0
Epoch  1110 Loss  0.00018678329070098698
Training accuracy is  1.0
Epoch  1120 Loss  0.000184667733265087
Training accuracy is  1.0
Epoch  1130 Loss  0.00018255521612772718
Training accuracy is  1.0
Epoch  1140 Loss  0.00018046637705992907
Training accuracy is  1.0
Epoch  1150 Loss  0.0001784401829354465
Training accuracy is  1.0
Epoch  1160 Loss  0.0001764735789038241
Training accuracy is  1.0
Epoch  1170 Loss  0.00017453677719458938
Training accuracy is  1.0
Epoch  1180 Loss  0.00017262976325582713
Training accuracy is  1.0
Epoch  1190 Loss  0.00017075255163945258
Training accuracy is  1.0
Epoch  1200 Loss  0.0001689051277935505
Training accuracy is  1.0
Epoch  1210 Loss  0.00016711727948859334
Training accuracy is  1.0
Epoch  1220 Loss  0.0001653592480579391
Training accuracy is  1.0
Epoch  1230 Loss  0.00016363100439775735
Training accuracy is  1.0
Epoch  1240 Loss  0.00016190274618566036
Training accuracy is  1.0
Epoch  1250 Loss  0.00016026388038881123
Training accuracy is  1.0
Epoch  1260 Loss  0.0001586250145919621
Training accuracy is  1.0
Epoch  1270 Loss  0.00015701595111750066
Training accuracy is  1.0
Epoch  1280 Loss  0.00015540688764303923
Training accuracy is  1.0
Epoch  1290 Loss  0.00015391700435429811
Training accuracy is  1.0
Epoch  1300 Loss  0.0001523675164207816
Training accuracy is  1.0
Epoch  1310 Loss  0.0001508776331320405
Training accuracy is  1.0
Epoch  1320 Loss  0.00014941753761377186
```

```
Training accuracy is  1.0
Epoch  1330 Loss  0.0001479872444178909
Training accuracy is  1.0
Epoch  1340 Loss  0.00014655693667009473
Training accuracy is  1.0
Epoch  1350 Loss  0.00014515644579660147
Training accuracy is  1.0
Epoch  1360 Loss  0.0001437857426935807
Training accuracy is  1.0
Epoch  1370 Loss  0.00014244482736103237
Training accuracy is  1.0
Epoch  1380 Loss  0.00014110391202848405
Training accuracy is  1.0
Epoch  1390 Loss  0.00013982260134071112
Training accuracy is  1.0
Epoch  1400 Loss  0.00013851147377863526
Training accuracy is  1.0
Epoch  1410 Loss  0.0001372897531837225
Training accuracy is  1.0
Epoch  1420 Loss  0.00013603823026642203
Training accuracy is  1.0
Epoch  1430 Loss  0.00013481650967150927
Training accuracy is  1.0
Epoch  1440 Loss  0.00013359477452468127
Training accuracy is  1.0
Epoch  1450 Loss  0.00013243264402262866
Training accuracy is  1.0
Epoch  1460 Loss  0.00013130030129104853
Training accuracy is  1.0
Epoch  1470 Loss  0.00013013817078899592
Training accuracy is  1.0
Epoch  1480 Loss  0.00012903561582788825
Training accuracy is  1.0
Epoch  1490 Loss  0.0001279330754186958
Training accuracy is  1.0
Epoch  1500 Loss  0.00012680073268711567
Training accuracy is  1.0
Epoch  1510 Loss  0.00012575779692269862
Training accuracy is  1.0
Epoch  1520 Loss  0.0001247446343768388
Training accuracy is  1.0
Epoch  1530 Loss  0.00012367189629003406
Training accuracy is  1.0
Epoch  1540 Loss  0.00012262894597370178
Training accuracy is  1.0
Epoch  1550 Loss  0.00012161578342784196
Training accuracy is  1.0
Epoch  1560 Loss  0.00012066222552675754
Training accuracy is  1.0
Epoch  1570 Loss  0.00011964907025685534
Training accuracy is  1.0
Epoch  1580 Loss  0.00011869551963172853
Training accuracy is  1.0
Epoch  1590 Loss  0.00011774194717872888
Training accuracy is  1.0
Epoch  1600 Loss  0.00011678839655360207
Training accuracy is  1.0
Epoch  1610 Loss  0.0001158944214694202
Training accuracy is  1.0
Epoch  1620 Loss  0.00011494086356833577
Training accuracy is  1.0
```

```
Epoch  1630 Loss  0.00011404690303606912
Training accuracy is  1.0
Epoch  1640 Loss  0.00011318273027427495
Training accuracy is  1.0
Epoch  1650 Loss  0.00011231856478843838
Training accuracy is  1.0
Epoch  1660 Loss  0.0001114245897042565
Training accuracy is  1.0
Epoch  1670 Loss  0.00011056041694246233
Training accuracy is  1.0
Epoch  1680 Loss  0.00010978565114783123
Training accuracy is  1.0
Epoch  1690 Loss  0.00010892147838603705
Training accuracy is  1.0
Epoch  1700 Loss  0.00010814670531544834
Training accuracy is  1.0
Epoch  1710 Loss  0.00010734212992247194
Training accuracy is  1.0
Epoch  1720 Loss  0.00010650775220710784
Training accuracy is  1.0
Epoch  1730 Loss  0.00010576278145890683
Training accuracy is  1.0
Epoch  1740 Loss  0.0001049581915140152
Training accuracy is  1.0
Epoch  1750 Loss  0.00010421322076581419
Training accuracy is  1.0
Epoch  1760 Loss  0.00010343844041926786
Training accuracy is  1.0
Epoch  1770 Loss  0.00010269346239510924
Training accuracy is  1.0
Epoch  1780 Loss  0.00010200807446381077
Training accuracy is  1.0
Epoch  1790 Loss  0.00010126309643965214
Training accuracy is  1.0
Epoch  1800 Loss  0.00010057777157843113
Training accuracy is  1.0
Epoch  1810 Loss  9.983273048419505e-05
Training accuracy is  1.0
Epoch  1820 Loss  9.91473498288542e-05
Training accuracy is  1.0
Epoch  1830 Loss  9.846196189755574e-05
Training accuracy is  1.0
Epoch  1840 Loss  9.777657396625727e-05
Training accuracy is  1.0
Epoch  1850 Loss  9.709119331091642e-05
Training accuracy is  1.0
Epoch  1860 Loss  9.643560770200565e-05
Training accuracy is  1.0
Epoch  1870 Loss  9.578001481713727e-05
Training accuracy is  1.0
Epoch  1880 Loss  9.51542315306142e-05
Training accuracy is  1.0
Epoch  1890 Loss  9.449863864574581e-05
Training accuracy is  1.0
Epoch  1900 Loss  9.390265040565282e-05
Training accuracy is  1.0
Epoch  1910 Loss  9.327685984317213e-05
Training accuracy is  1.0
Epoch  1920 Loss  9.265106928069144e-05
Training accuracy is  1.0
Epoch  1930 Loss  9.202527871821076e-05
```

```
Training accuracy is  1.0
Epoch  1940 Loss  9.142928320216015e-05
Training accuracy is  1.0
Epoch  1950 Loss  9.083329496206716e-05
Training accuracy is  1.0
Epoch  1960 Loss  9.023729944601655e-05
Training accuracy is  1.0
Epoch  1970 Loss  8.967111352831125e-05
Training accuracy is  1.0
Epoch  1980 Loss  8.910492761060596e-05
Training accuracy is  1.0
Epoch  1990 Loss  8.853872714098543e-05
Training accuracy is  1.0
Epoch  2000 Loss  8.797253394732252e-05
Training accuracy is  1.0
Epoch  2010 Loss  8.74063407536596e-05
Training accuracy is  1.0
Epoch  2020 Loss  8.686994260642678e-05
Training accuracy is  1.0
Epoch  2030 Loss  8.633354445919394e-05
Training accuracy is  1.0
Epoch  2040 Loss  8.582695591030642e-05
Training accuracy is  1.0
Epoch  2050 Loss  8.529056503903121e-05
Training accuracy is  1.0
Epoch  2060 Loss  8.475415961584076e-05
Training accuracy is  1.0
Epoch  2070 Loss  8.424757106695324e-05
Training accuracy is  1.0
Epoch  2080 Loss  8.37111656437628e-05
Training accuracy is  1.0
Epoch  2090 Loss  8.323437941726297e-05
Training accuracy is  1.0
Epoch  2100 Loss  8.269797399407253e-05
Training accuracy is  1.0
Epoch  2110 Loss  8.225097553804517e-05
Training accuracy is  1.0
Epoch  2120 Loss  8.171458466676995e-05
Training accuracy is  1.0
Epoch  2130 Loss  8.12377838883549e-05
Training accuracy is  1.0
Epoch  2140 Loss  8.079079270828515e-05
Training accuracy is  1.0
Epoch  2150 Loss  8.028418233152479e-05
Training accuracy is  1.0
Epoch  2160 Loss  7.983719115145504e-05
Training accuracy is  1.0
Epoch  2170 Loss  7.936039037303999e-05
Training accuracy is  1.0
Epoch  2180 Loss  7.891339191701263e-05
Training accuracy is  1.0
Epoch  2190 Loss  7.846639346098527e-05
Training accuracy is  1.0
Epoch  2200 Loss  7.798959268257022e-05
Training accuracy is  1.0
Epoch  2210 Loss  7.754259422654286e-05
Training accuracy is  1.0
Epoch  2220 Loss  7.70955957705155e-05
Training accuracy is  1.0
Epoch  2230 Loss  7.667839236091822e-05
Training accuracy is  1.0
```

```
Epoch  2240 Loss  7.626119622727856e-05
Training accuracy is  1.0
Epoch  2250 Loss  7.584399281768128e-05
Training accuracy is  1.0
Epoch  2260 Loss  7.539699436165392e-05
Training accuracy is  1.0
Epoch  2270 Loss  7.494999590562657e-05
Training accuracy is  1.0
Epoch  2280 Loss  7.456259481841698e-05
Training accuracy is  1.0
Epoch  2290 Loss  7.41751937312074e-05
Training accuracy is  1.0
Epoch  2300 Loss  7.375799032161012e-05
Training accuracy is  1.0
Epoch  2310 Loss  7.334078691201285e-05
Training accuracy is  1.0
Epoch  2320 Loss  7.295339310076088e-05
Training accuracy is  1.0
Epoch  2330 Loss  7.256598473759368e-05
Training accuracy is  1.0
Epoch  2340 Loss  7.220837869681418e-05
Training accuracy is  1.0
Epoch  2350 Loss  7.176138024078682e-05
Training accuracy is  1.0
Epoch  2360 Loss  7.143358379835263e-05
Training accuracy is  1.0
Epoch  2370 Loss  7.101637311279774e-05
Training accuracy is  1.0
Epoch  2380 Loss  7.068857667036355e-05
Training accuracy is  1.0
Epoch  2390 Loss  7.027137326076627e-05
Training accuracy is  1.0
Epoch  2400 Loss  6.994356954237446e-05
Training accuracy is  1.0
Epoch  2410 Loss  6.955616845516488e-05
Training accuracy is  1.0
Epoch  2420 Loss  6.916876009199768e-05
Training accuracy is  1.0
Epoch  2430 Loss  6.88111613271758e-05
Training accuracy is  1.0
Epoch  2440 Loss  6.848335760878399e-05
Training accuracy is  1.0
Epoch  2450 Loss  6.812575156800449e-05
Training accuracy is  1.0
Epoch  2460 Loss  6.77681528031826e-05
Training accuracy is  1.0
Epoch  2470 Loss  6.744035636074841e-05
Training accuracy is  1.0
Epoch  2480 Loss  6.70827430440113e-05
Training accuracy is  1.0
Epoch  2490 Loss  6.675493932561949e-05
Training accuracy is  1.0
Epoch  2500 Loss  6.642713560722768e-05
Training accuracy is  1.0
Epoch  2510 Loss  6.60695368424058e-05
Training accuracy is  1.0
Epoch  2520 Loss  6.574173312401399e-05
Training accuracy is  1.0
Epoch  2530 Loss  6.544372445205227e-05
Training accuracy is  1.0
Epoch  2540 Loss  6.511592800961807e-05
```

```
Training accuracy is  1.0
Epoch  2550 Loss  6.475832196883857e-05
Training accuracy is  1.0
Epoch  2560 Loss  6.446031329687685e-05
Training accuracy is  1.0
Epoch  2570 Loss  6.416231190087274e-05
Training accuracy is  1.0
Epoch  2580 Loss  6.383450818248093e-05
Training accuracy is  1.0
Epoch  2590 Loss  6.353650678647682e-05
Training accuracy is  1.0
Epoch  2600 Loss  6.323850539047271e-05
Training accuracy is  1.0
Epoch  2610 Loss  6.294049671851099e-05
Training accuracy is  1.0
Epoch  2620 Loss  6.264249532250687e-05
Training accuracy is  1.0
Epoch  2630 Loss  6.231469160411507e-05
Training accuracy is  1.0
Epoch  2640 Loss  6.204648525454104e-05
Training accuracy is  1.0
Epoch  2650 Loss  6.174847658257931e-05
Training accuracy is  1.0
Epoch  2660 Loss  6.14504751865752e-05
Training accuracy is  1.0
Epoch  2670 Loss  6.118226883700117e-05
Training accuracy is  1.0
Epoch  2680 Loss  6.088426744099706 4e-05
Training accuracy is  1.0
Epoch  2690 Loss  6.058625876903534e-05
Training accuracy is  1.0
Epoch  2700 Loss  6.031805241946131e-05
Training accuracy is  1.0
Epoch  2710 Loss  6.00200510234572e-05
Training accuracy is  1.0
Epoch  2720 Loss  5.9781646996270865e-05
Training accuracy is  1.0
Epoch  2730 Loss  5.945384327787906e-05
Training accuracy is  1.0
Epoch  2740 Loss  5.9215439250692725e-05
Training accuracy is  1.0
Epoch  2750 Loss  5.8917430578731e-05
Training accuracy is  1.0
Epoch  2760 Loss  5.867902291356586e-05
Training accuracy is  1.0
Epoch  2770 Loss  5.841082020197064e-05
Training accuracy is  1.0
Epoch  2780 Loss  5.8142617490375414e-05
Training accuracy is  1.0
Epoch  2790 Loss  5.790421346318908e-05
Training accuracy is  1.0
Epoch  2800 Loss  5.760620842920616e-05
Training accuracy is  1.0
Epoch  2810 Loss  5.736780076404102e-05
Training accuracy is  1.0
Epoch  2820 Loss  5.70995980524458e-05
Training accuracy is  1.0
Epoch  2830 Loss  5.6861194025259465e-05
Training accuracy is  1.0
Epoch  2840 Loss  5.6622786360094324e-05
Training accuracy is  1.0
```

```
Epoch  2850 Loss  5.638437869492918e-05
Training accuracy is  1.0
Epoch  2860 Loss  5.611617598333396e-05
Training accuracy is  1.0
Epoch  2870 Loss  5.5907570640556514e-05
Training accuracy is  1.0
Epoch  2880 Loss  5.563936792896129e-05
Training accuracy is  1.0
Epoch  2890 Loss  5.5371157941408455e-05
Training accuracy is  1.0
Epoch  2900 Loss  5.5132750276243314e-05
Training accuracy is  1.0
Epoch  2910 Loss  5.489434624905698e-05
Training accuracy is  1.0
Epoch  2920 Loss  5.4655942221870646e-05
Training accuracy is  1.0
Epoch  2930 Loss  5.44473368790932e-05
Training accuracy is  1.0
Epoch  2940 Loss  5.4208932851906866e-05
Training accuracy is  1.0
Epoch  2950 Loss  5.403012619353831e-05
Training accuracy is  1.0
Epoch  2960 Loss  5.3791722166351974e-05
Training accuracy is  1.0
Epoch  2970 Loss  5.358311682357453e-05
Training accuracy is  1.0
Epoch  2980 Loss  5.328511178959161e-05
Training accuracy is  1.0
Epoch  2990 Loss  5.3076506446814165e-05
Training accuracy is  1.0
```

In [ ]: