Batch RNN

June 28, 2022

[470]:
```
import numpy as np
import matplotlib.pyplot as plt
import torch
from torchvision import datasets
from torchvision.transforms import ToTensor
from torch.utils.data import TensorDataset, DataLoader
import sys
import copy
```

[2]:
```
def tanh(x):
    return np.tanh(x)

def d_tanh(x):
    return 1 - np.square(tanh(x))

def sigmoid(y):
    y[y < -700] = -700
    return 1.0/(1.0+np.exp(-y))

def d_sigmoid(x):
    z = sigmoid(x)*(1-sigmoid(x))
    return z


def softmax(x):
    m = nn.Softmax(dim=0)
    return m(torch.tensor(x)).numpy()

def d_softmax(z):
    '''
    return the jacobian of the softmax
    '''
    z = softmax(z)
    return np.diag(z) - np.outer(z,z)

def MSE(a,b):
    return np.sum(np.square(a-b), axis = 1)
```

```python
def d_MSE(output_activations, y):
    return (output_activations - y)
```

[228]:
```python
class RNN:

    def __init__(self,input_dim,hidden_dim,output_dim,lr):
        self.input_dim = input_dim
        self.hidden_dim = hidden_dim
        self.output_dim = output_dim
        self.w_in = np.random.uniform(-1,1, (hidden_dim, input_dim))
        self.w_hidden = np.random.uniform(-1,1, (hidden_dim, hidden_dim))
        self.b_hidden = np.random.uniform(-1,1, (hidden_dim, 1))
        self.w_out = np.random.uniform(-1,1, (output_dim, hidden_dim))
        self.b_out = np.random.uniform(-1,1, (output_dim, 1))
        self.loss = []
        self.lr = lr
        self.epsilon = 0.000001

    def forward(self, x):
        hidden_state = np.zeros((self.hidden_dim,1))
        T = x.shape[-1]
        prediction = np.zeros((T,1))

        for i in range(T):
            x_in = self.w_in @ x[:,i].reshape(self.input_dim,1)
            x_hid = self.w_hidden @ hidden_state + self.b_hidden
            hid_out = sigmoid(x_in + x_hid)
            hidden_state = hid_out
            x_out = self.w_out @ hid_out + self.b_out
            prediction[i] = x_out
        return prediction

    def update(self, x, y):

        dldb_hidden,dldb_out,dldw_in,dldw_hidden,dldw_out,loss = self.
↪backprop(x, y)
        self.w_in -=  self.lr * dldw_in
        self.w_hidden -=  self.lr * dldw_hidden
        self.b_hidden -=  self.lr * dldb_hidden
        self.w_out -=  self.lr * dldw_out
        self.b_out -=  self.lr * dldb_out
        self.loss.append(loss)

    def gradient_approximation(self, x, y):
        nable_w_in = np.zeros(self.w_in.shape)
        nable_w_hidden = np.zeros(self.w_hidden.shape)
        nable_w_out = np.zeros(self.w_out.shape)
```

```python
        nable_b_hidden = np.zeros(self.b_hidden.shape)
        nable_b_out = np.zeros(self.b_out.shape)

        aprox1_w_in = np.zeros(self.w_in.shape)
        aprox2_w_in = np.zeros(self.w_in.shape)
        aprox1_w_hidden = np.zeros(self.w_hidden.shape)
        aprox2_w_hidden = np.zeros(self.w_hidden.shape)
        aprox1_w_out = np.zeros(self.w_out.shape)
        aprox2_w_out = np.zeros(self.w_out.shape)
        aprox1_b_hidden = np.zeros(self.b_hidden.shape)
        aprox2_b_hidden = np.zeros(self.b_hidden.shape)
        aprox1_b_out = np.zeros(self.b_out.shape)
        aprox2_b_out = np.zeros(self.b_out.shape)

        for k in range(len(self.b_hidden)):
            aprox1_b_hidden = copy.deepcopy(self.b_hidden)
            aprox2_b_hidden = copy.deepcopy(self.b_hidden)
            aprox1_b_hidden[k] += self.epsilon
            aprox2_b_hidden[k] -= self.epsilon

            hidden_state_1 = np.zeros((batch_size,self.hidden_dim,1))
            loss1 = np.zeros((batch_size,1))
            T = x.shape[-1]
            for i in range(T):
                x_in = self.w_in @ np.expand_dims(x[:,:,i],axis = 2)
                x_hid = self.w_hidden @ hidden_state_1 + aprox1_b_hidden
                hid_out = sigmoid(x_in + x_hid)
                hidden_state_1 = hid_out
                x_out = self.w_out @ hid_out + self.b_out
                loss1 += MSE(x_out,np.expand_dims(y[:,:,i],axis = 2))

            hidden_state_2 = np.zeros((batch_size,self.hidden_dim,1))
            loss2 = np.zeros((batch_size,1))
            T = x.shape[-1]

            for i in range(T):
                x_in = self.w_in @ np.expand_dims(x[:,:,i],axis = 2)
                x_hid = self.w_hidden @ hidden_state_2 + aprox2_b_hidden
                hid_out = sigmoid(x_in + x_hid)
                hidden_state_2 = hid_out
                x_out = self.w_out @ hid_out + self.b_out
                loss2 += MSE(x_out,np.expand_dims(y[:,:,i],axis = 2))

            nable_b_hidden[k] = (np.sum((loss1 - loss2) / (2*self.epsilon))) /␣
↪batch_size
```

```
        for k in range(len(self.w_in)):
            for j in range(len(self.w_in[k])):
                aprox1_w_in = copy.deepcopy(self.w_in)
                aprox2_w_in = copy.deepcopy(self.w_in)
                aprox1_w_in[k][j] += self.epsilon
                aprox2_w_in[k][j] -= self.epsilon

                hidden_state_1 = np.zeros((batch_size,self.hidden_dim,1))
                loss1 = np.zeros((batch_size,1))
                T = x.shape[-1]
                for i in range(T):
                    x_in = aprox1_w_in @ np.expand_dims(x[:,:,i],axis = 2)
                    x_hid = self.w_hidden @ hidden_state_1 + self.b_hidden
                    hid_out = sigmoid(x_in + x_hid)
                    hidden_state_1 = hid_out
                    x_out = self.w_out @ hid_out + self.b_out
                    loss1 += MSE(x_out,np.expand_dims(y[:,:,i],axis = 2))

                hidden_state_2 = np.zeros((batch_size,self.hidden_dim,1))
                loss2 = np.zeros((batch_size,1))
                T = x.shape[-1]

                for i in range(T):
                    x_in = aprox2_w_in @ np.expand_dims(x[:,:,i],axis = 2)
                    x_hid = self.w_hidden @ hidden_state_2 + self.b_hidden
                    hid_out = sigmoid(x_in + x_hid)
                    hidden_state_2 = hid_out
                    x_out = self.w_out @ hid_out + + self.b_out
                    loss2 += MSE(x_out,np.expand_dims(y[:,:,i],axis = 2))

                nable_w_in[k][j] = (np.sum((loss1 - loss2) / (2*self.epsilon)))␣
↪/ batch_size


        for k in range(len(self.w_hidden)):
            for j in range(len(self.w_hidden[k])):
                aprox1_w_hidden = copy.deepcopy(self.w_hidden)
                aprox2_w_hidden = copy.deepcopy(self.w_hidden)
                aprox1_w_hidden[k][j] += self.epsilon
                aprox2_w_hidden[k][j] -= self.epsilon

                hidden_state_1 = np.zeros((batch_size,self.hidden_dim,1))
                loss1 = np.zeros((batch_size,1))
                T = x.shape[-1]
                for i in range(T):
                    x_in = self.w_in @ np.expand_dims(x[:,:,i],axis = 2)
                    x_hid = aprox1_w_hidden @ hidden_state_1 + self.b_hidden
```

```
                    hid_out = sigmoid(x_in + x_hid)
                    hidden_state_1 = hid_out
                    x_out = self.w_out @ hid_out + self.b_out
                    loss1 += MSE(x_out,np.expand_dims(y[:,:,i],axis = 2))

                hidden_state_2 = np.zeros((batch_size,self.hidden_dim,1))
                loss2 = np.zeros((batch_size,1))
                T = x.shape[-1]

                for i in range(T):
                    x_in = self.w_in @ np.expand_dims(x[:,:,i],axis = 2)
                    x_hid = aprox2_w_hidden @ hidden_state_2 + self.b_hidden
                    hid_out = sigmoid(x_in + x_hid)
                    hidden_state_2 = hid_out
                    x_out = self.w_out @ hid_out + + self.b_out
                    loss2 += MSE(x_out,np.expand_dims(y[:,:,i],axis = 2))

                nable_w_hidden[k][j] = (np.sum((loss1 - loss2) / (2*self.
↪epsilon))) / batch_size

        for k in range(len(self.w_out)):
            for j in range(len(self.w_out[k])):
                aprox1_w_out = copy.deepcopy(self.w_out)
                aprox2_w_out = copy.deepcopy(self.w_out)
                aprox1_w_out[k][j] += self.epsilon
                aprox2_w_out[k][j] -= self.epsilon

                hidden_state_1 = np.zeros((batch_size,self.hidden_dim,1))
                loss1 = np.zeros((batch_size,1))
                T = x.shape[-1]
                for i in range(T):
                    x_in = self.w_in @ np.expand_dims(x[:,:,i],axis = 2)
                    x_hid = self.w_hidden @ hidden_state_1 + self.b_hidden
                    hid_out = sigmoid(x_in + x_hid)
                    hidden_state_1 = hid_out
                    x_out = aprox1_w_out @ hid_out + self.b_out
                    loss1 += MSE(x_out,np.expand_dims(y[:,:,i],axis = 2))

                hidden_state_2 = np.zeros((batch_size,self.hidden_dim,1))
                loss2 = np.zeros((batch_size,1))
                T = x.shape[-1]

                for i in range(T):
                    x_in = self.w_in @ np.expand_dims(x[:,:,i],axis = 2)
                    x_hid = self.w_hidden @ hidden_state_2 + self.b_hidden
                    hid_out = sigmoid(x_in + x_hid)
                    hidden_state_2 = hid_out
```

```
                    x_out = aprox2_w_out @ hid_out + + self.b_out
                    loss2 += MSE(x_out,np.expand_dims(y[:,:,i],axis = 2))

                nable_w_out[k][j] = (np.sum((loss1 - loss2) / (2*self.
↪epsilon))) / batch_size

        for k in range(len(self.b_out)):
            aprox1_b_out = copy.deepcopy(self.b_out)
            aprox2_b_out = copy.deepcopy(self.b_out)
            aprox1_b_out[k] += self.epsilon
            aprox2_b_out[k] -= self.epsilon

            hidden_state_1 = np.zeros((batch_size,self.hidden_dim,1))
            loss1 = np.zeros((batch_size,1))
            T = x.shape[-1]
            for i in range(T):
                x_in = self.w_in @ np.expand_dims(x[:,:,i],axis = 2)
                x_hid = self.w_hidden @ hidden_state_1 + self.b_hidden
                hid_out = sigmoid(x_in + x_hid)
                hidden_state_1 = hid_out
                x_out = self.w_out @ hid_out + aprox1_b_out
                loss1 += MSE(x_out,np.expand_dims(y[:,:,i],axis = 2))


            hidden_state_2 = np.zeros((batch_size,self.hidden_dim,1))
            loss2 = np.zeros((batch_size,1))
            T = x.shape[-1]

            for i in range(T):
                x_in = self.w_in @ np.expand_dims(x[:,:,i],axis = 2)
                x_hid = self.w_hidden @ hidden_state_2 + self.b_hidden
                hid_out = sigmoid(x_in + x_hid)
                hidden_state_2 = hid_out
                x_out = self.w_out @ hid_out + aprox2_b_out
                loss2 += MSE(x_out,np.expand_dims(y[:,:,i],axis = 2))

            nable_b_out[k] = (np.sum((loss1 - loss2) / (2*self.epsilon))) /␣
↪batch_size

        return nable_b_hidden,nable_b_out,nable_w_in,nable_w_hidden,nable_w_out

    def backprop(self, x, y):
        hidden_state = np.zeros((batch_size,self.hidden_dim,1))
        hidden_out = []
        hidden_out.append(hidden_state)
        hidden_z = []
        y_out = []
```

```
        dldw_in = np.zeros((batch_size,) + self.w_in.shape)
        dldw_hidden = np.zeros((batch_size,) + self.w_hidden.shape)
        dldb_hidden = np.zeros((batch_size,) + self.b_hidden.shape)
        dldw_out = np.zeros((batch_size,) + self.w_out.shape)
        dldb_out = np.zeros((batch_size,) + self.b_out.shape)
        loss = 0
        T = x.shape[-1]

        for i in range(T):
            x_in = self.w_in @ np.expand_dims(x[:,:,i],axis = 2)
            x_hid = self.w_hidden @ hidden_state + self.b_hidden
            hidden_z.append(x_in+x_hid)
            # hid_out = tanh(x_in + x_hid)
            hid_out = sigmoid(x_in + x_hid)
            hidden_out.append(hid_out)
            hidden_state = hid_out
            x_out = self.w_out @ hid_out + self.b_out
            y_out.append(x_out)
            loss += MSE(x_out,np.expand_dims(y[:,:,i],axis = 2))

        '''
        dldw_out
        '''


        for i in range(1,T+1):

            dldb_out += d_MSE(y_out[-i],np.expand_dims(y[:,:,-i],axis = 2))
            dldw_out += d_MSE(y_out[-i],np.expand_dims(y[:,:,-i],axis = 2)) @␣
↪np.transpose(hidden_out[-i],(0, 2, 1))

        '''
        dldw_hidden
        dldw_input
        '''

        for i in range(1,T+1):

            delta = d_MSE(y_out[-i],np.expand_dims(y[:,:,-i],axis = 2))
            delta = np.transpose(self.w_out) @ delta * d_sigmoid(hidden_z[-i])
            dldb_hidden += delta
            dldw_hidden += delta @ np.transpose(hidden_out[-i-1],(0, 2, 1))
            dldw_in += delta @ np.transpose(np.expand_dims(x[:,:,-i],axis =␣
↪2),(0, 2, 1))

            for k in range(i+1,T+1):
```

```
                delta = np.transpose(self.w_hidden) @ delta *␣
↪d_sigmoid(hidden_z[-k])
                dldb_hidden += delta
                dldw_hidden += delta @ np.transpose(hidden_out[-k-1],(0, 2, 1))
                dldw_in += delta @ np.transpose(np.expand_dims(x[:,:,-k],axis =␣
↪2),(0, 2, 1))

        dldb_hidden = 2 * dldb_hidden.sum(axis = 0) / batch_size
        dldb_out = 2 * dldb_out.sum(axis = 0) / batch_size
        dldw_in = 2 * dldw_in.sum(axis = 0) / batch_size
        dldw_hidden = 2 * dldw_hidden.sum(axis = 0) / batch_size
        dldw_out = 2 * dldw_out.sum(axis = 0) / batch_size
        loss = loss.sum(axis = 0) / batch_size

        return (dldb_hidden,dldb_out,dldw_in,dldw_hidden,dldw_out,loss)
```

[479]:
```
input = []
target = []
T = 50
Num = 50


for i in range(Num):

    #parameters = np.random.randint(1,5,(3,1))
    p = np.random.uniform(1,5)
    fix = np.array([0.0,1.0])
    parameters = np.insert(fix,0,p).reshape(3,1)
    x = np.tile(parameters,(T))
    time = np.linspace(0, 5, T)
    frequency = x[0][0]
    theta = x[1][0]
    amplitude = x[2][0]
    y = amplitude * np.sin(frequency * time + theta)
    input.append(x)
    target.append(y)

batch_size = 5
input = np.array(input)
target = np.array(target).reshape(Num,T)
batch_input = np.split(input, Num/batch_size)
batch_target = np.split(target, Num/batch_size)
```

[480]:
```
%%time
sinrnn = RNN(3,100,1,0.0001)
Loss = []
```

```
for i in range(500):
    idx = 0
    for input,target in zip(batch_input,batch_target):
        #nable_b_hidden,nable_b_out,nable_w_in,nable_w_hidden,nable_w_out =␣
 ↪sinrnn.gradient_approximation(input,np.expand_dims(target,axis = 1))
        #dldb_hidden,dldb_out,dldw_in,dldw_hidden,dldw_out,loss = sinrnn.
 ↪backprop(input,np.expand_dims(target,axis = 1))
        #print('approximation',(nable_w_hidden / dldw_hidden).sum())
        sinrnn.update(input,np.expand_dims(target,axis = 1))
        Loss.append(np.sum(sinrnn.loss))
    print('epoch',i,': ',np.sum(sinrnn.loss))
    sinrnn.loss.clear()
```

```
epoch 0 :  384.96272671316905
epoch 1 :  317.3553270089092
epoch 2 :  297.8930150586428
epoch 3 :  284.2806773588231
epoch 4 :  274.323033908126
epoch 5 :  266.78721260860516
epoch 6 :  260.98492895343463
epoch 7 :  256.4408307366314
epoch 8 :  252.79029919712232
epoch 9 :  249.77860928144597
epoch 10 :  247.24154617060128
epoch 11 :  245.07318332293607
epoch 12 :  243.20071353003752
epoch 13 :  241.56971989930193
epoch 14 :  240.13696631860316
epoch 15 :  238.86722971692166
epoch 16 :  237.73177371657448
epoch 17 :  236.70730289292837
epoch 18 :  235.77503224337556
epoch 19 :  234.91983346922345
epoch 20 :  234.12949061726087
epoch 21 :  233.3940839561098
epoch 22 :  232.7054982038124
epoch 23 :  232.05703840300407
epoch 24 :  231.44313307432913
epoch 25 :  230.8591055536442
epoch 26 :  230.3009975431707
epoch 27 :  229.7654322755983
epoch 28 :  229.24950765118027
epoch 29 :  228.75071208877748
epoch 30 :  228.26685766198142
epoch 31 :  227.79602646553258
epoch 32 :  227.33652717696705
```

```
epoch 33 :  226.88685953163002
epoch 34 :  226.44568498536253
epoch 35 :  226.01180225078173
epoch 36 :  225.58412669891223
epoch 37 :  225.16167284627122
epoch 38 :  224.74353931889536
epoch 39 :  224.32889581409353
epoch 40 :  223.91697167871067
epoch 41 :  223.50704579722614
epoch 42 :  223.09843753980613
epoch 43 :  222.6904985636521
epoch 44 :  222.28260529370215
epoch 45 :  221.87415193319325
epoch 46 :  221.46454387243182
epoch 47 :  221.05319137659356
epoch 48 :  220.63950344141452
epoch 49 :  220.22288171003808
epoch 50 :  219.80271434578083
epoch 51 :  219.3783697550583
epoch 52 :  218.94919005336817
epoch 53 :  218.51448416700026
epoch 54 :  218.0735204671442
epoch 55 :  217.62551884651882
epoch 56 :  217.16964218015588
epoch 57 :  216.70498717564166
epoch 58 :  216.2305747367672
epoch 59 :  215.74534017452268
epoch 60 :  215.24812395744607
epoch 61 :  214.73766428567427
epoch 62 :  214.2125937248688
epoch 63 :  213.6714436168122
epoch 64 :  213.11266219208636
epoch 65 :  212.5346554092338
epoch 66 :  211.93586347518033
epoch 67 :  211.31489007609636
epoch 68 :  210.67070351888853
epoch 69 :  210.00292488003817
epoch 70 :  209.31220077538597
epoch 71 :  208.6006199970453
epoch 72 :  207.8720744737025
epoch 73 :  207.13240843639625
epoch 74 :  206.3891964607837
epoch 75 :  205.65109132172108
epoch 76 :  204.92686720747037
epoch 77 :  204.22443382864157
epoch 78 :  203.5500873394855
epoch 79 :  202.90809923457567
epoch 80 :  202.30054599624916
```

```
epoch 81 :  201.72714576605443
epoch 82 :  201.18478523459024
epoch 83 :  200.66631597160656
epoch 84 :  200.15799632832352
epoch 85 :  199.63468451912033
epoch 86 :  199.0525187775265
epoch 87 :  198.34886123741248
epoch 88 :  197.54609662846735
epoch 89 :  197.5855381928708
epoch 90 :  201.23771233742588
epoch 91 :  202.49627483650525
epoch 92 :  201.60125097463
epoch 93 :  201.14363840751147
epoch 94 :  200.59735452245062
epoch 95 :  200.07152924006377
epoch 96 :  199.55553279400783
epoch 97 :  199.0411855870473
epoch 98 :  198.5222516602989
epoch 99 :  197.99436633617267
epoch 100 :  197.45530523071133
epoch 101 :  196.90488402740394
epoch 102 :  196.3447167873849
epoch 103 :  195.77793182028256
epoch 104 :  195.20887462522793
epoch 105 :  194.6427956060561
epoch 106 :  194.08550765835588
epoch 107 :  193.54299657372954
epoch 108 :  193.02097098417593
epoch 109 :  192.52434754390853
epoch 110 :  192.0566846441998
epoch 111 :  191.61961121383052
epoch 112 :  191.21235207203156
epoch 113 :  190.83151980829902
epoch 114 :  190.47138428644107
epoch 115 :  190.12476924840908
epoch 116 :  189.78450442130486
epoch 117 :  189.44504129896555
epoch 118 :  189.10362998685315
epoch 119 :  188.76056179901684
epoch 120 :  188.41839306145766
epoch 121 :  188.0805177197531
epoch 122 :  187.7496769405347
epoch 123 :  187.42692059994573
epoch 124 :  187.11129248970803
epoch 125 :  186.80023916414308
epoch 126 :  186.4905101944382
epoch 127 :  186.17916423727888
epoch 128 :  185.8642789100091
```

```
epoch 129 :  185.54511758514744
epoch 130 :  185.2217703939583
epoch 131 :  184.89451638104754
epoch 132 :  184.5632303667304
epoch 133 :  184.22706825233695
epoch 134 :  183.88448489737593
epoch 135 :  183.5334705774882
epoch 136 :  183.17181077949158
epoch 137 :  182.79720528313044
epoch 138 :  182.4071938309334
epoch 139 :  181.99895272892294
epoch 140 :  181.56907793433362
epoch 141 :  181.11343576972865
epoch 142 :  180.6270879931638
epoch 143 :  180.1042586894007
epoch 144 :  179.53834806669855
epoch 145 :  178.92208172205235
epoch 146 :  178.24793831200304
epoch 147 :  177.50898199177732
epoch 148 :  176.7001551977624
epoch 149 :  175.81998710999778
epoch 150 :  174.8726692308666
epoch 151 :  173.87058877946976
epoch 152 :  172.8375830834478
epoch 153 :  171.8149254020076
epoch 154 :  170.872640856566
epoch 155 :  170.21088361607775
epoch 156 :  170.35758921801977
epoch 157 :  180.56840310511728
epoch 158 :  255.90678632798503
epoch 159 :  164.8127647970242
epoch 160 :  159.75190261102387
epoch 161 :  166.84433866219376
epoch 162 :  167.0318176706391
epoch 163 :  163.5924329465957
epoch 164 :  161.36845522147306
epoch 165 :  161.3798584733346
epoch 166 :  161.64197413870804
epoch 167 :  160.9732069211817
epoch 168 :  158.97058515943
epoch 169 :  156.37599166485123
epoch 170 :  154.36767208061286
epoch 171 :  153.1916489512289
epoch 172 :  152.30839939744902
epoch 173 :  151.31586174950507
epoch 174 :  150.15039587432972
epoch 175 :  148.9160482647877
epoch 176 :  147.7113370834704
```

```
epoch 177 :  146.5624743638016
epoch 178 :  145.44954588761811
epoch 179 :  144.35038562653764
epoch 180 :  143.2575610146989
epoch 181 :  142.17346345655037
epoch 182 :  141.1017920405459
epoch 183 :  140.0440447684763
epoch 184 :  138.99995503933678
epoch 185 :  137.96856963924674
epoch 186 :  136.94859866156213
epoch 187 :  135.938312664722
epoch 188 :  134.9354485029285
epoch 189 :  133.93722797386238
epoch 190 :  132.94042036251057
epoch 191 :  131.94140614541115
epoch 192 :  130.93624478305014
epoch 193 :  129.92075848415155
epoch 194 :  128.89063877561878
epoch 195 :  127.84157889644717
epoch 196 :  126.76943160286558
epoch 197 :  125.67038691217745
epoch 198 :  124.5411581889286
epoch 199 :  123.37915923635545
epoch 200 :  122.18265122405693
epoch 201 :  120.95083760751342
epoch 202 :  119.68388807587843
epoch 203 :  118.38287816373294
epoch 204 :  117.04963765096893
epoch 205 :  115.68650644226213
epoch 206 :  114.29600100714359
epoch 207 :  112.88040041139102
epoch 208 :  111.4412745232414
epoch 209 :  109.97900507059491
epoch 210 :  108.49239389258985
epoch 211 :  106.97849677365716
epoch 212 :  105.43282854758665
epoch 213 :  103.8500116320573
epoch 214 :  102.22477793994193
epoch 215 :  100.55305279512301
epoch 216 :  98.8327732529315
epoch 217 :  97.06419917123534
epoch 218 :  95.24970712634548
epoch 219 :  93.39326277133493
epoch 220 :  91.49982982465548
epoch 221 :  89.57489734638813
epoch 222 :  87.62418218420737
epoch 223 :  85.65347564263008
epoch 224 :  83.6685765932506
```

```
epoch 225 :  81.6752634659772
epoch 226 :  79.67927446120267
epoch 227 :  77.68627580752987
epoch 228 :  75.70180398096292
epoch 229 :  73.7311751609145
epoch 230 :  71.77936583954893
epoch 231 :  69.85087999765061
epoch 232 :  67.94962634251237
epoch 233 :  66.07883053507368
epoch 234 :  64.24100151171713
epoch 235 :  62.43795998354627
epoch 236 :  60.67092456228182
epoch 237 :  58.94064033084561
epoch 238 :  57.24752847814199
epoch 239 :  55.59183458620551
epoch 240 :  53.97375652694728
epoch 241 :  52.393539052718864
epoch 242 :  50.851529230408175
epoch 243 :  49.34819338181032
epoch 244 :  47.884101209980706
epoch 245 :  46.459885905313186
epoch 246 :  45.07619024220909
epoch 247 :  43.733608277596126
epoch 248 :  42.43263068371758
epoch 249 :  41.17359949144496
epoch 250 :  39.9566755695442
epoch 251 :  38.78181991469982
epoch 252 :  37.64878803810613
epoch 253 :  36.55713552422624
epoch 254 :  35.50623219910692
epoch 255 :  34.495282188017434
epoch 256 :  33.52334733316755
epoch 257 :  32.589371845232534
epoch 258 :  31.69220655929506
epoch 259 :  30.830631668121686
epoch 260 :  30.00337725550069
epoch 261 :  29.20914131860141
epoch 262 :  28.446605240431072
epoch 263 :  27.71444685508305
epoch 264 :  27.011351351883942
epoch 265 :  26.336020306379993
epoch 266 :  25.687179124003272
epoch 267 :  25.063583152709672
epoch 268 :  24.464022677848448
epoch 269 :  23.887326966510628
epoch 270 :  23.332367486891748
epoch 271 :  22.798060394725653
epoch 272 :  22.28336835507066
```

```
epoch 273 :  21.7873017532046
epoch 274 :  21.308919341500545
epoch 275 :  20.847328367759406
epoch 276 :  20.4016842324294
epoch 277 :  19.97118972546804
epoch 278 :  19.555093896893887
epoch 279 :  19.15269061732309
epoch 280 :  18.763316885483572
epoch 281 :  18.38635093868188
epoch 282 :  18.021210219585356
epoch 283 :  17.667349248737953
epoch 284 :  17.324257447307534
epoch 285 :  16.991456949030187
epoch 286 :  16.66850043451646
epoch 287 :  16.354969015287026
epoch 288 :  16.05047018936092
epoch 289 :  15.754635885067787
epoch 290 :  15.467120605107281
epoch 291 :  15.187599678807189
epoch 292 :  14.915767627030473
epoch 293 :  14.651336641262192
epoch 294 :  14.394035176027883
epoch 295 :  14.143606651914256
epoch 296 :  13.899808265032998
epoch 297 :  13.66240989772444
epoch 298 :  13.431193124594245
epoch 299 :  13.205950307549088
epoch 300 :  12.986483773306283
epoch 301 :  12.77260506683703
epoch 302 :  12.564134274342504
epoch 303 :  12.36089940959305
epoch 304 :  12.16273585778423
epoch 305 :  11.96948587142529
epoch 306 :  11.780998113170968
epoch 307 :  11.597127240917857
epoch 308 :  11.417733530888675
epoch 309 :  11.242682534828269
epoch 310 :  11.071844767808813
epoch 311 :  10.905095423495425
epoch 312 :  10.742314114057306
epoch 313 :  10.58338463220467
epoch 314 :  10.428194733105185
epoch 315 :  10.276635934183577
epoch 316 :  10.128603331023834
epoch 317 :  9.983995427791497
epoch 318 :  9.842713980766176
epoch 319 :  9.704663853726183
epoch 320 :  9.569752884062765
```

```
epoch 321 :  9.437891758619518
epoch 322 :  9.30899389835623
epoch 323 :  9.182975351028903
epoch 324 :  9.059754691155259
epoch 325 :  8.939252926608281
epoch 326 :  8.821393411242822
epoch 327 :  8.706101763015024
epoch 328 :  8.593305787104468
epoch 329 :  8.482935403591961
epoch 330 :  8.37492257929039
epoch 331 :  8.269201263354415
epoch 332 :  8.165707326335738
epoch 333 :  8.064378502373732
epoch 334 :  7.965154334244282
epoch 335 :  7.867976121008526
epoch 336 :  7.7727868680321235
epoch 337 :  7.679531239163559
epoch 338 :  7.588155510879241
epoch 339 :  7.498607528223324
epoch 340 :  7.410836662385606
epoch 341 :  7.324793769775825
epoch 342 :  7.240431152468694
epoch 343 :  7.157702519904704
epoch 344 :  7.076562951745239
epoch 345 :  6.996968861792214
epoch 346 :  6.918877962892302
epoch 347 :  6.842249232752263
epoch 348 :  6.767042880605437
epoch 349 :  6.693220314673309
epoch 350 :  6.620744110374081
epoch 351 :  6.5495779792377276
epoch 352 :  6.479686738489243
epoch 353 :  6.411036281273249
epoch 354 :  6.343593547488622
epoch 355 :  6.277326495214982
epoch 356 :  6.212204072710146
epoch 357 :  6.148196190961474
epoch 358 :  6.085273696781574
epoch 359 :  6.023408346434489
epoch 360 :  5.962572779783907
epoch 361 :  5.902740494957605
epoch 362 :  5.843885823520702
epoch 363 :  5.785983906153124
epoch 364 :  5.729010668827771
epoch 365 :  5.672942799487402
epoch 366 :  5.6177577252149025
epoch 367 :  5.563433589895883
epoch 368 :  5.509949232372644
```

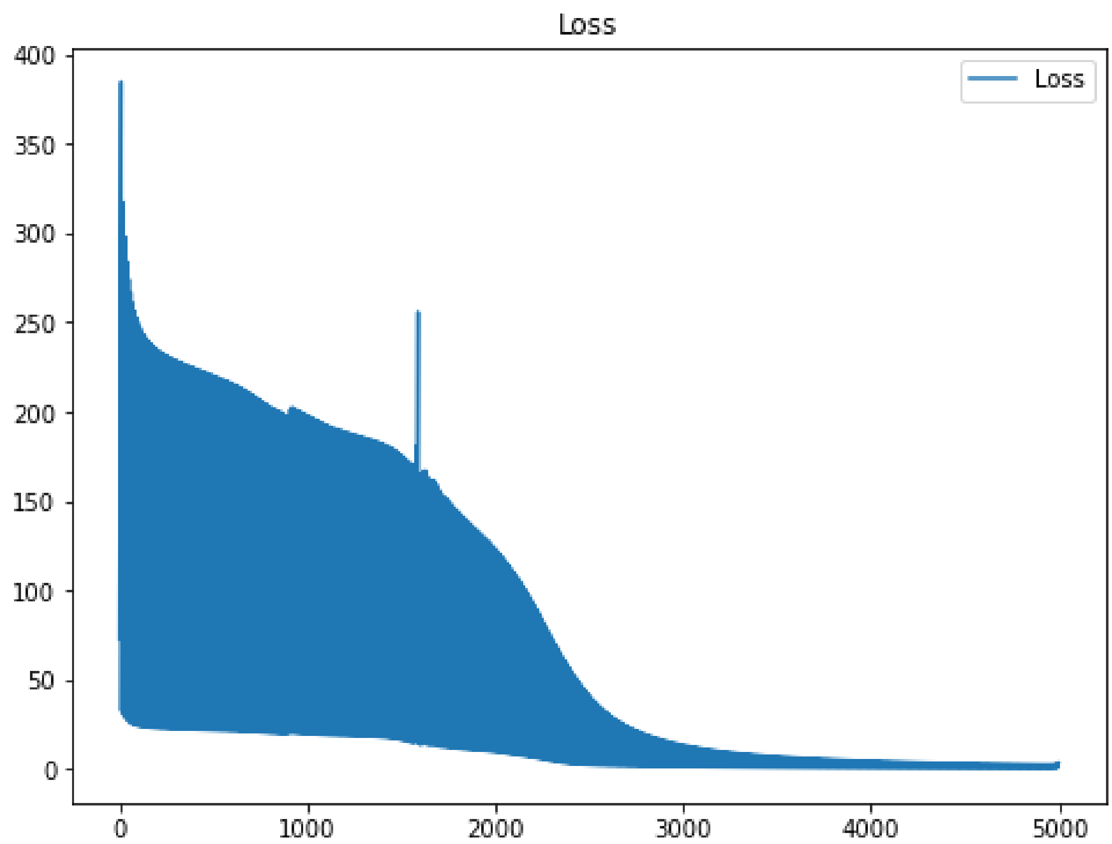
```
epoch 369 :  5.457284165085695
epoch 370 :  5.405418553201362
epoch 371 :  5.354333194224104
epoch 372 :  5.3040094980903545
epoch 373 :  5.254429467742388
epoch 374 :  5.205575680178496
epoch 375 :  5.157431267975819
epoch 376 :  5.10997990128569
epoch 377 :  5.063205770294023
epoch 378 :  5.017093568144808
epoch 379 :  4.971628474321492
epoch 380 :  4.926796138482419
epoch 381 :  4.882582664744194
epoch 382 :  4.838974596407274
epoch 383 :  4.795958901118874
epoch 384 :  4.753522956467072
epoch 385 :  4.71165453599899
epoch 386 :  4.670341795657347
epoch 387 :  4.629573260626069
epoch 388 :  4.589337812581075
epoch 389 :  4.549624677335939
epoch 390 :  4.510423412876894
epoch 391 :  4.4717238977779274
epoch 392 :  4.433516319988252
epoch 393 :  4.395791165985564
epoch 394 :  4.358539210284504
epoch 395 :  4.321751505293895
epoch 396 :  4.2854193715140045
epoch 397 :  4.249534388065295
epoch 398 :  4.21408838354027
epoch 399 :  4.179073427171103
epoch 400 :  4.14448182030191
epoch 401 :  4.110306088161286
epoch 402 :  4.076538971924753
epoch 403 :  4.0431734210574035
epoch 404 :  4.010202585932624
epoch 405 :  3.9776198107147698
epoch 406 :  3.945418626500779
epoch 407 :  3.91359274471213
epoch 408 :  3.88213605072799
epoch 409 :  3.851042597754939
epoch 410 :  3.820306600922389
epoch 411 :  3.789922431600907
epoch 412 :  3.7598846119315823
epoch 413 :  3.7301878095626715
epoch 414 :  3.7008268325860487
epoch 415 :  3.671796624665417
epoch 416 :  3.6430922603528804
```

```
epoch 417 :  3.6147089405839026
epoch 418 :  3.5866419883484837
epoch 419 :  3.5588868445287978
epoch 420 :  3.531439063900731
epoch 421 :  3.5042943112909795
epoch 422 :  3.4774483578880284
epoch 423 :  3.4508970776973467
epoch 424 :  3.424636444139388
epoch 425 :  3.3986625267839594
epoch 426 :  3.3729714882168547
epoch 427 :  3.347559581033457
epoch 428 :  3.322423144956405
epoch 429 :  3.297558604072317
epoch 430 :  3.272962464183729
epoch 431 :  3.248631310272127
epoch 432 :  3.2245618040706456
epoch 433 :  3.2007506817403724
epoch 434 :  3.177194751648417
epoch 435 :  3.1538908922459714
epoch 436 :  3.1308360500407497
epoch 437 :  3.1080272376645612
epoch 438 :  3.085461532031936
epoch 439 :  3.063136072588269
epoch 440 :  3.0410480596457528
epoch 441 :  3.019194752806197
epoch 442 :  2.997573469468785
epoch 443 :  2.9761815834227017
epoch 444 :  2.9550165235244394
epoch 445 :  2.934075772459481
epoch 446 :  2.9133568655889155
epoch 447 :  2.892857389882643
epoch 448 :  2.8725749829417095
epoch 449 :  2.8525073321115344
epoch 450 :  2.8326521736895875
epoch 451 :  2.8130072922351497
epoch 452 :  2.7935705199846628
epoch 453 :  2.7743397363840936
epoch 454 :  2.7553128677471648
epoch 455 :  2.7364878870551865
epoch 456 :  2.717862813915222
epoch 457 :  2.699435714701047
epoch 458 :  2.6812047029049206
epoch 459 :  2.663167939739246
epoch 460 :  2.6453236350372595
epoch 461 :  2.6276700485161033
epoch 462 :  2.6102054914884603
epoch 463 :  2.592928329132403
epoch 464 :  2.5758369834699755
```

```
epoch 465 :  2.5589299372541876
epoch 466 :  2.542205739036521
epoch 467 :  2.5256630097866335
epoch 468 :  2.5093004515795165
epoch 469 :  2.49311685906655
epoch 470 :  2.477111134740107
epoch 471 :  2.4612823094251945
epoch 472 :  2.445629570055381
epoch 473 :  2.430152297708571
epoch 474 :  2.4148501202528454
epoch 475 :  2.399722986018511
epoch 476 :  2.3847712680475404
epoch 477 :  2.36999591328064
epoch 478 :  2.3553986584686575
epoch 479 :  2.3409823461890777
epoch 480 :  2.326751392620409
epoch 481 :  2.312712487818457
epoch 482 :  2.2988756560962202
epoch 483 :  2.285255880543604
epoch 484 :  2.2718756222607284
epoch 485 :  2.2587687780966528
epoch 486 :  2.245986987855145
epoch 487 :  2.2336098516140996
epoch 488 :  2.221761807138025
epoch 489 :  2.2106406876846245
epoch 490 :  2.2005675352673926
epoch 491 :  2.192076908277077
epoch 492 :  2.1860886708752094
epoch 493 :  2.1842541055723355
epoch 494 :  2.1896994986282494
epoch 495 :  2.208732588644157
epoch 496 :  2.255005036600102
epoch 497 :  2.360162356860622
epoch 498 :  2.601639591278877
epoch 499 :  3.171622111761611
CPU times: user 12min 9s, sys: 1.6 s, total: 12min 11s
Wall time: 12min 11s
```

[481]:

```
plt.figure(figsize = (8, 6))
plt.plot(Loss, label = 'Loss', markersize = 3)
plt.title("Loss")
plt.legend()
plt.show()
```



[453]:
```
for i in range(500,1000):
    for input,target in zip(batch_input,batch_target):
        sinrnn.update(input,np.expand_dims(target,axis = 1))
        Loss.append(np.sum(sinrnn.loss))
    print('epoch',i,': ',np.sum(sinrnn.loss))
    sinrnn.loss.clear()
```

```
epoch 500 :  198.24944756470492
epoch 501 :  202.5444883848307
epoch 502 :  197.45436722823044
epoch 503 :  189.84922773637317
epoch 504 :  173.92292426693962
epoch 505 :  184.4960952117848
epoch 506 :  181.6608038112666
epoch 507 :  176.2926666643467
epoch 508 :  184.8534715946752
epoch 509 :  180.4805952762221
epoch 510 :  176.84409165695405
epoch 511 :  181.71584382087377
epoch 512 :  174.01212389699123
epoch 513 :  201.08760487224978
```

```
epoch 514 :  204.56618449056907
epoch 515 :  200.37877496468496
epoch 516 :  195.83047839800435
epoch 517 :  187.82127452574198
epoch 518 :  171.76495285539806
epoch 519 :  183.8068260791311
epoch 520 :  181.46353852913788
epoch 521 :  175.03745045641867
epoch 522 :  182.16293727215896
epoch 523 :  174.94470148982015
epoch 524 :  195.8491044659138
epoch 525 :  200.61273931693222
epoch 526 :  194.4432023958976
epoch 527 :  183.2766026936142
epoch 528 :  174.14490100345577
epoch 529 :  178.95078577325538
epoch 530 :  170.72901570681202
epoch 531 :  195.2472104052692
epoch 532 :  199.99600510064937
epoch 533 :  193.43276515517982
epoch 534 :  181.0683364357388
epoch 535 :  175.29763014850812
epoch 536 :  169.33497172294577
epoch 537 :  175.47832506660305
epoch 538 :  176.24702380532764
epoch 539 :  168.29740843139552
epoch 540 :  174.10512762802097
epoch 541 :  171.27223277379645
epoch 542 :  169.52829327404083
epoch 543 :  183.90735008346488
epoch 544 :  178.60397953638386
epoch 545 :  176.72492288247003
epoch 546 :  167.70911395335517
epoch 547 :  176.43872217796675
epoch 548 :  169.04740058495315
epoch 549 :  169.0886068866996
epoch 550 :  168.70257653326462
epoch 551 :  174.43481708539767
epoch 552 :  180.14106418269301
epoch 553 :  176.38138332177317
epoch 554 :  197.75603433370344
epoch 555 :  202.45918032594003
epoch 556 :  196.23859431011766
epoch 557 :  186.5965134981833
epoch 558 :  167.9752625322221
epoch 559 :  183.32585478050504
epoch 560 :  184.19404644186034
epoch 561 :  175.8827440123618
```

```
epoch 562 :  167.40344985822344
epoch 563 :  193.20202696999078
epoch 564 :  198.07641062082675
epoch 565 :  188.81564245734327
epoch 566 :  170.4653649300191
epoch 567 :  185.54429933266772
epoch 568 :  188.14160175300586
epoch 569 :  168.38014366917847
epoch 570 :  188.2316379856375
epoch 571 :  192.98477059621817
epoch 572 :  177.77224149029576
epoch 573 :  176.18715447964436
epoch 574 :  167.83445259399238
epoch 575 :  189.9982050996111
epoch 576 :  195.41831014289696
epoch 577 :  183.9798219040802
epoch 578 :  165.71554285368546
epoch 579 :  185.57036032225938
epoch 580 :  188.92634926376797
epoch 581 :  169.91016033617257
epoch 582 :  184.4591722771166
epoch 583 :  186.9021545737937
epoch 584 :  167.1127045128086
epoch 585 :  185.36817306802558
epoch 586 :  188.79760423147707
epoch 587 :  169.7864413102939
epoch 588 :  182.92408557095206
epoch 589 :  184.24752996773327
epoch 590 :  164.847795846819
epoch 591 :  187.38818854700287
epoch 592 :  191.62295779832039
epoch 593 :  175.92598722036107
epoch 594 :  174.91083610676614
epoch 595 :  166.81460693811192
epoch 596 :  186.49677459156402
epoch 597 :  190.63898115006486
epoch 598 :  173.72105549381132
epoch 599 :  177.15318609572267
epoch 600 :  171.23371679665968
epoch 601 :  183.14393095410298
epoch 602 :  184.79804138755856
epoch 603 :  164.84538531461126
epoch 604 :  182.91567269905036
epoch 605 :  184.87736770024674
epoch 606 :  164.72813132654682
epoch 607 :  182.77560397188935
epoch 608 :  184.7066998003348
epoch 609 :  164.48831090434518
```

```
epoch 610 :  182.42572425574437
epoch 611 :  184.1404888840592
epoch 612 :  163.93608524051956
epoch 613 :  182.71990758694696
epoch 614 :  184.71320256747794
epoch 615 :  164.32903466215944
epoch 616 :  180.94280317545605
epoch 617 :  181.49150970183283
epoch 618 :  162.8212539402514
epoch 619 :  181.15531960544803
epoch 620 :  180.15076706310697
epoch 621 :  166.80941821164575
epoch 622 :  170.80729735695883
epoch 623 :  162.32855723246823
epoch 624 :  180.07962071511085
epoch 625 :  176.52124535576183
epoch 626 :  168.37624734925475
epoch 627 :  175.69587002713445
epoch 628 :  167.3153235858682
epoch 629 :  181.05645734820217
epoch 630 :  182.3390571874286
epoch 631 :  161.4569987870986
epoch 632 :  183.63560400986017
epoch 633 :  186.52210372690502
epoch 634 :  166.2628558899888
epoch 635 :  174.48058017552086
epoch 636 :  168.85519842115355
epoch 637 :  181.35341209196983
epoch 638 :  182.4415780781769
epoch 639 :  161.5591508520815
epoch 640 :  177.41427808645753
epoch 641 :  175.20456035602757
epoch 642 :  168.5745539450775
epoch 643 :  160.61479983719377
epoch 644 :  180.86154587654949
epoch 645 :  179.9082031147004
epoch 646 :  160.1098816820844
epoch 647 :  182.11140650635772
epoch 648 :  183.21233121625386
epoch 649 :  161.87127221942768
epoch 650 :  175.52241446388607
epoch 651 :  171.67856624532217
epoch 652 :  174.00805607586574
epoch 653 :  166.7632181822241
epoch 654 :  177.4543594457402
epoch 655 :  175.73890521022702
epoch 656 :  168.42411858660788
epoch 657 :  159.1917738743095
```

```
epoch 658 :  183.1166219598456
epoch 659 :  185.74628450797536
epoch 660 :  165.83207833534428
epoch 661 :  169.34913299292094
epoch 662 :  161.30635133674338
epoch 663 :  176.94662733739096
epoch 664 :  175.51766202537755
epoch 665 :  166.8288440664002
epoch 666 :  157.76944491066834
epoch 667 :  183.63022946685302
epoch 668 :  186.261988599711
epoch 669 :  167.46692970753884
epoch 670 :  168.322467821885
epoch 671 :  159.75083122810062
epoch 672 :  174.13703376740864
epoch 673 :  170.33376497838935
epoch 674 :  169.34919100905012
epoch 675 :  159.69884682895312
epoch 676 :  174.2503194218022
epoch 677 :  170.80837213175275
epoch 678 :  167.12490011852498
epoch 679 :  157.02207323720378
epoch 680 :  177.2976384302179
epoch 681 :  176.84121166242568
epoch 682 :  155.2717228657098
epoch 683 :  174.61563350175584
epoch 684 :  171.8190789807155
epoch 685 :  165.72719952965548
epoch 686 :  155.5963102106594
epoch 687 :  175.58565089275749
epoch 688 :  173.96801089050112
epoch 689 :  155.35534178155416
epoch 690 :  175.63808683101982
epoch 691 :  172.42672692460422
epoch 692 :  156.54768587633774
epoch 693 :  165.89139983594498
epoch 694 :  154.84353142553175
epoch 695 :  170.427185609472
epoch 696 :  164.6925400185089
epoch 697 :  174.58431335632068
epoch 698 :  171.07546403698373
epoch 699 :  153.7133514685843
epoch 700 :  171.7418094762678
epoch 701 :  165.39627594354494
epoch 702 :  165.43857495776672
epoch 703 :  154.96342952608742
epoch 704 :  159.8230701020177
epoch 705 :  150.6000382221719
```

```
epoch 706 :  181.93742507137426
epoch 707 :  182.41730663437897
epoch 708 :  163.96935625200985
epoch 709 :  156.59686810777893
epoch 710 :  147.88151647425747
epoch 711 :  175.5872597666983
epoch 712 :  173.54111575187468
epoch 713 :  150.5397001008146
epoch 714 :  151.28278592060533
epoch 715 :  158.39745056289368
epoch 716 :  145.97909203803522
epoch 717 :  162.39422001049533
epoch 718 :  152.60706251310907
epoch 719 :  171.8079386571859
epoch 720 :  167.62090827791837
epoch 721 :  142.72373762270277
epoch 722 :  146.34042546741858
epoch 723 :  146.69400928470156
epoch 724 :  154.38933180354363
epoch 725 :  139.11752693909733
epoch 726 :  131.03274160260813
epoch 727 :  189.0189730595796
epoch 728 :  178.53091186511762
epoch 729 :  163.1476672373317
epoch 730 :  142.64297544600737
epoch 731 :  128.11716185749466
epoch 732 :  136.83137645083772
epoch 733 :  121.41924192417783
epoch 734 :  122.4388979638049
epoch 735 :  169.85563873177958
epoch 736 :  151.76063674293923
epoch 737 :  129.52792486582064
epoch 738 :  108.70872656763629
epoch 739 :  102.7525614397106
epoch 740 :  138.2697728137293
epoch 741 :  111.91631119531486
epoch 742 :  94.52835736905982
epoch 743 :  164.09790162070541
epoch 744 :  127.70558724374699
epoch 745 :  101.00350429652215
epoch 746 :  90.10970480247838
epoch 747 :  96.93143971155395
epoch 748 :  90.13637814078223
epoch 749 :  143.32456867872463
epoch 750 :  84.51492085036955
epoch 751 :  148.44882426891465
epoch 752 :  88.8342577996097
epoch 753 :  71.3522864505098
```

```
epoch 754 :  103.51414040559322
epoch 755 :  161.93477290786197
epoch 756 :  91.66505430555681
epoch 757 :  76.83855334047722
epoch 758 :  154.63450286102469
epoch 759 :  92.49427305177014
epoch 760 :  76.67900781351172
epoch 761 :  92.20215625176728
epoch 762 :  59.69493440391055
epoch 763 :  75.52805040994704
epoch 764 :  163.52066366740308
epoch 765 :  75.17587154096012
epoch 766 :  176.61438757350925
epoch 767 :  92.85180559050062
epoch 768 :  64.10451236791923
epoch 769 :  126.48681177709969
epoch 770 :  66.10536851221059
epoch 771 :  60.173437299732925
epoch 772 :  58.71204387432381
epoch 773 :  193.79603231659752
epoch 774 :  96.72198832607884
epoch 775 :  77.40874438142156
epoch 776 :  48.37631819452589
epoch 777 :  122.38807527005088
epoch 778 :  56.045528001436764
epoch 779 :  133.35268464632873
epoch 780 :  59.760472175719066
epoch 781 :  30.836265500012342
epoch 782 :  47.759342283963214
epoch 783 :  46.54761769263891
epoch 784 :  67.69296862391147
epoch 785 :  103.67331795785023
epoch 786 :  36.08745701943293
epoch 787 :  130.05046988854005
epoch 788 :  51.774415563647906
epoch 789 :  29.271750371818094
epoch 790 :  47.352054838674526
epoch 791 :  21.4783427590489
epoch 792 :  80.12992897281212
epoch 793 :  28.47490745891056
epoch 794 :  29.026854495421905
epoch 795 :  42.54173313293042
epoch 796 :  24.00400765607066
epoch 797 :  39.03306624919102
epoch 798 :  28.38324670581524
epoch 799 :  31.146181999902783
epoch 800 :  28.76069057659184
epoch 801 :  27.638057970939727
```

```
epoch 802 :  26.798066796245003
epoch 803 :  26.12963309328024
epoch 804 :  25.565660097836314
epoch 805 :  25.072215369139883
epoch 806 :  24.63150241376434
epoch 807 :  24.232802199131196
epoch 808 :  23.868747540355617
epoch 809 :  23.533817010108773
epoch 810 :  23.223675471641524
epoch 811 :  22.93483432758819
epoch 812 :  22.66444335643515
epoch 813 :  22.41014748495363
epoch 814 :  22.16998235387007
epoch 815 :  21.94229600579118
epoch 816 :  21.725689131223596
epoch 817 :  21.518968809248936
epoch 818 :  21.321112208619997
epoch 819 :  21.131237746814065
epoch 820 :  20.948581917251094
epoch 821 :  20.772480488715146
epoch 822 :  20.602353124528026
epoch 823 :  20.437690710168006
epoch 824 :  20.27804484963332
epoch 825 :  20.1230191148157
epoch 826 :  19.972261723193494
epoch 827 :  19.825459387078755
epoch 828 :  19.68233212910502
epoch 829 :  19.542628898284406
epoch 830 :  19.40612385180753
epoch 831 :  19.272613192235955
epoch 832 :  19.141912469240282
epoch 833 :  19.01385427081259
epoch 834 :  18.888286241722252
epoch 835 :  18.765069377497117
epoch 836 :  18.6440765508638
epoch 837 :  18.525191234738692
epoch 838 :  18.4083063917778
epoch 839 :  18.293323505413685
epoch 840 :  18.180151731389255
epoch 841 :  18.068707152205704
epoch 842 :  17.95891211969956
epoch 843 :  17.850694673368103
epoch 844 :  17.743988023963116
epoch 845 :  17.638730093558774
epoch 846 :  17.534863104635047
epoch 847 :  17.432333211851972
epoch 848 :  17.331090171165883
epoch 849 :  17.231087041700018
```

epoch 850 : 17.132279916474488
epoch 851 : 17.034627678635946
epoch 852 : 16.938091780314377
epoch 853 : 16.84263604161031
epoch 854 : 16.748226467562834
epoch 855 : 16.654831081216578
epoch 856 : 16.562419771149088
epoch 857 : 16.47096415203105
epoch 858 : 16.380437436937388
epoch 859 : 16.290814320312258
epoch 860 : 16.202070870574083
epoch 861 : 16.114184431504817
epoch 862 : 16.027133531604477
epoch 863 : 15.940897800737845
epoch 864 : 15.855457893410428
epoch 865 : 15.770795418118434
epoch 866 : 15.686892872252113
epoch 867 : 15.603733582079405
epoch 868 : 15.521301647394088
epoch 869 : 15.439581890430881
epoch 870 : 15.358559808705664
epoch 871 : 15.278221531453765
epoch 872 : 15.198553779376159
epoch 873 : 15.119543827417795
epoch 874 : 15.041179470349972
epoch 875 : 14.963448990906002
epoch 876 : 14.88634113029167
epoch 877 : 14.80984506086105
epoch 878 : 14.733950360789656
epoch 879 : 14.658646990594631
epoch 880 : 14.583925271342144
epoch 881 : 14.509775864417234
epoch 882 : 14.436189752728833
epoch 883 : 14.363158223239287
epoch 884 : 14.290672850709006
epoch 885 : 14.218725482569523
epoch 886 : 14.147308224828569
epoch 887 : 14.076413428929332
epoch 888 : 14.006033679496587
epoch 889 : 13.93616178288425
epoch 890 : 13.866790756488303
epoch 891 : 13.797913818736207
epoch 892 : 13.72952437972918
epoch 893 : 13.661616032467606
epoch 894 : 13.59418254463064
epoch 895 : 13.527217850848192
epoch 896 : 13.460716045464185
epoch 897 : 13.394671375703311

```
epoch 898 :  13.329078235269302
epoch 899 :  13.263931158292898
epoch 900 :  13.199224813638944
epoch 901 :  13.134953999528456
epoch 902 :  13.071113638459769
epoch 903 :  13.007698772407709
epoch 904 :  12.944704558280419
epoch 905 :  12.88212626361604
epoch 906 :  12.819959262501257
epoch 907 :  12.758199031704066
epoch 908 :  12.696841146994105
epoch 909 :  12.635881279651187
epoch 910 :  12.575315193138696
epoch 911 :  12.51513873994406
epoch 912 :  12.455347858554417
epoch 913 :  12.395938570593268
epoch 914 :  12.33690697806313
epoch 915 :  12.27824926073595
epoch 916 :  12.219961673638196
epoch 917 :  12.162040544664903
epoch 918 :  12.104482272274709
epoch 919 :  12.047283323307115
epoch 920 :  11.990440230863587
epoch 921 :  11.933949592300744
epoch 922 :  11.87780806728462
epoch 923 :  11.822012375937565
epoch 924 :  11.766559297045953
epoch 925 :  11.711445666349528
epoch 926 :  11.656668374881448
epoch 927 :  11.602224367389718
epoch 928 :  11.548110640798166
epoch 929 :  11.49432424273988
epoch 930 :  11.440862270136025
epoch 931 :  11.387721867826066
epoch 932 :  11.33490022725368
epoch 933 :  11.282394585188579
epoch 934 :  11.230202222504614
epoch 935 :  11.178320462986997
epoch 936 :  11.126746672199129
epoch 937 :  11.07547825636216
epoch 938 :  11.024512661304795
epoch 939 :  10.97384737141067
epoch 940 :  10.923479908641028
epoch 941 :  10.873407831550168
epoch 942 :  10.82362873436825
epoch 943 :  10.774140246085215
epoch 944 :  10.724940029586095
epoch 945 :  10.676025780797543
```

```
epoch 946 :  10.627395227875798
epoch 947 :  10.57904613040562
epoch 948 :  10.53097627864062
epoch 949 :  10.483183492750806
epoch 950 :  10.43566562211013
epoch 951 :  10.388420544588
epoch 952 :  10.341446165885426
epoch 953 :  10.294740418862673
epoch 954 :  10.248301262920867
epoch 955 :  10.202126683367696
epoch 956 :  10.156214690837725
epoch 957 :  10.110563320693094
epoch 958 :  10.065170632480621
epoch 959 :  10.020034709368485
epoch 960 :  9.975153657634369
epoch 961 :  9.930525606134157
epoch 962 :  9.886148705821581
epoch 963 :  9.842021129249895
epoch 964 :  9.798141070114795
epoch 965 :  9.754506742790351
epoch 966 :  9.711116381893568
epoch 967 :  9.667968241851549
epoch 968 :  9.625060596488186
epoch 969 :  9.582391738618094
epoch 970 :  9.53995997965621
epoch 971 :  9.497763649234097
epoch 972 :  9.45580109483434
epoch 973 :  9.414070681427189
epoch 974 :  9.372570791124874
epoch 975 :  9.331299822841892
epoch 976 :  9.290256191965314
epoch 977 :  9.249438330036622
epoch 978 :  9.208844684441008
epoch 979 :  9.168473718104483
epoch 980 :  9.12832390920383
epoch 981 :  9.088393750878744
epoch 982 :  9.048681750958165
epoch 983 :  9.009186431689944
epoch 984 :  8.969906329479397
epoch 985 :  8.930839994638285
epoch 986 :  8.891985991133435
epoch 987 :  8.853342896354162
epoch 988 :  8.814909300869118
epoch 989 :  8.776683808211263
epoch 990 :  8.738665034648177
epoch 991 :  8.700851608973675
epoch 992 :  8.66324217229667
epoch 993 :  8.625835377840684
```

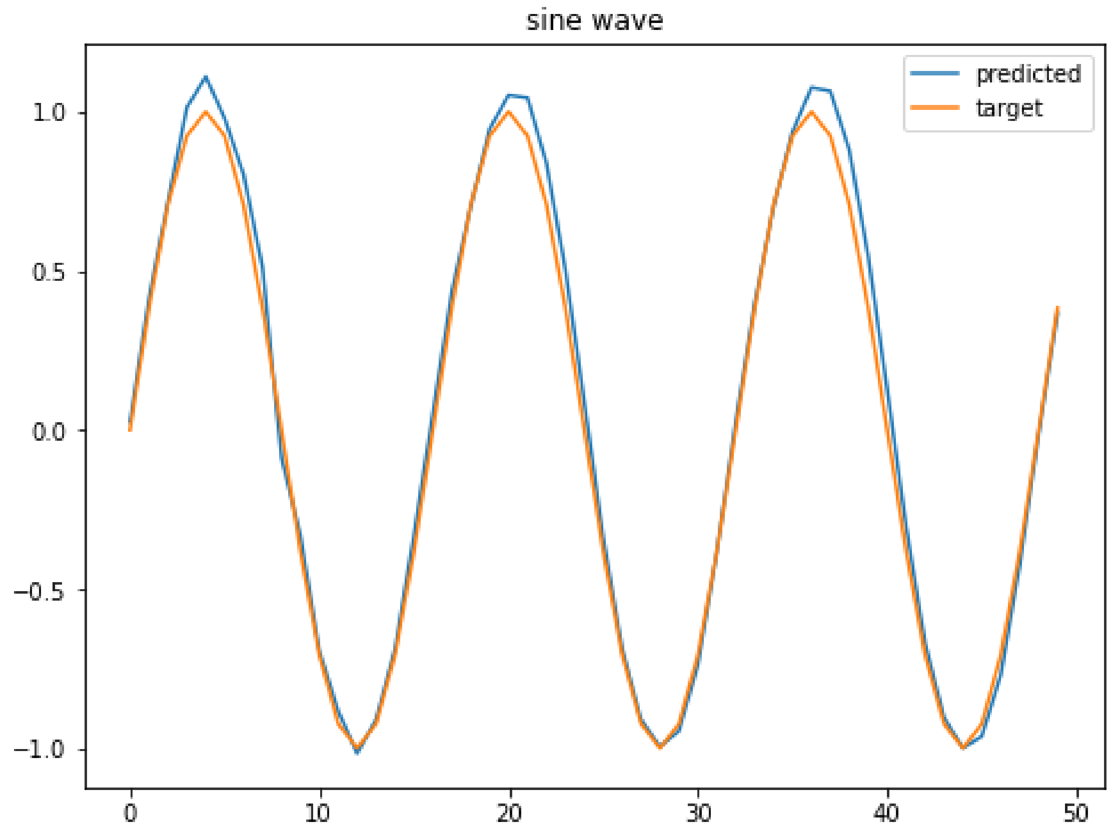
```
epoch 994 :  8.588629890744192
epoch 995 :  8.55162438787303
epoch 996 :  8.514817557629488
epoch 997 :  8.478208099775838
epoch 998 :  8.44179472525303
epoch 999 :  8.40557615601341
```

[493]:

```
#parameters = np.ones((3,1))
T = 50
p = np.random.uniform(1,5)
fix = np.array([0.0,1.0])
parameters = np.insert(fix,0,p).reshape(3,1)
x = np.tile(parameters,(T))
time = np.linspace(0, 5, T)
frequency = x[0][0]
theta = x[1][0]
amplitude = x[2][0]
y = amplitude * np.sin(frequency * time + theta)
print('frequency: ',p)


predicted = sinrnn.forward(x)
target = y
plt.figure(figsize = (8, 6))
plt.plot(predicted, label = 'predicted', markersize = 3)
plt.plot(target, label = 'target', markersize = 3)
plt.title("sine wave")
plt.legend()
plt.show()
```

```
frequency:  3.848427718978435
50
```

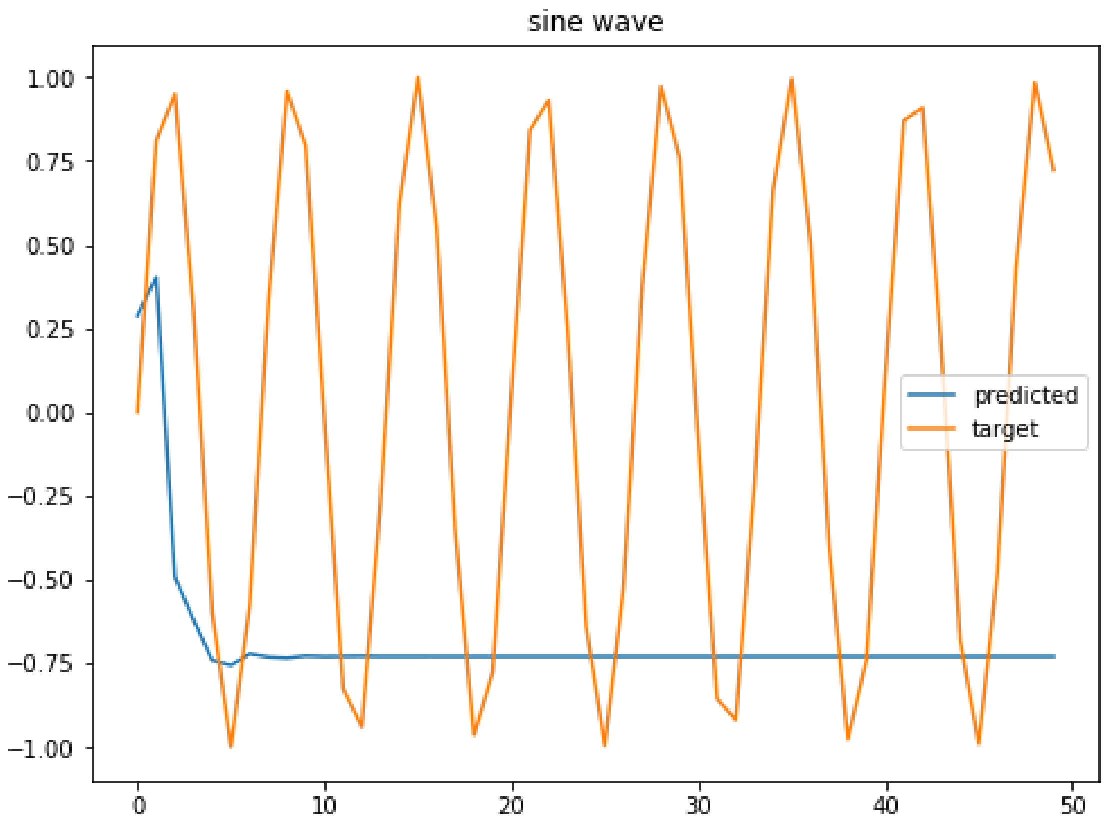


[495]:
```
T = 50
p = np.random.uniform(1,10)
fix = np.array([0.0,1.0])
parameters = np.insert(fix,0,p).reshape(3,1)
x = np.tile(parameters,(T))
time = np.linspace(0, 5, T)
frequency = x[0][0]
theta = x[1][0]
amplitude = x[2][0]
y = amplitude * np.sin(frequency * time + theta)
print('frequency: ',p)


predicted = sinrnn.forward(x)
target = y
plt.figure(figsize = (8, 6))
plt.plot(predicted, label = 'predicted', markersize = 3)
plt.plot(target, label = 'target', markersize = 3)
plt.title("sine wave")
plt.legend()
```

```
plt.show()
```

```
frequency:  9.263371130938035
50
```




```
[ ]:
```