Batch RNN

June 27, 2022

[1]:
```
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)
```

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

[228]:
```
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))

        print(T)
        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)
        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)
```

[414]:
```
input = []
target = []
T = 50
Num = 50


for i in range(Num):

    #parameters = np.random.randint(1,5,(3,1))
    p = np.random.randint(1,10)
    fix = np.array([0,1])
    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)
```

[415]:
```
%%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 :  3211.3155652208734
epoch 1 :  313.28849089258784
epoch 2 :  264.2473507681716
epoch 3 :  257.7493614826533
epoch 4 :  254.52409817533555
epoch 5 :  251.92466516350962
epoch 6 :  249.75500788702288
epoch 7 :  247.93005058443288
epoch 8 :  246.37484609120514
epoch 9 :  245.0308517791565
epoch 10 :  243.8545005319324
epoch 11 :  242.81334103297007
epoch 12 :  241.88280336008842
epoch 13 :  241.04393198639644
epoch 14 :  240.28184939483617
epoch 15 :  239.5847055448772
epoch 16 :  238.94294364129254
epoch 17 :  238.34877431155869
epoch 18 :  237.79579032318668
epoch 19 :  237.27867855828958
epoch 20 :  236.79300099263935
epoch 21 :  236.3350257474792
epoch 22 :  235.90159519384477
epoch 23 :  235.49002193514173
epoch 24 :  235.09800606194597
epoch 25 :  234.72356883145375
epoch 26 :  234.36499915623253
epoch 27 :  234.0208101685033
epoch 28 :  233.68970376851905
epoch 29 :  233.3705415410413
epoch 30 :  233.06232078061979
epoch 31 :  232.7641546370988
```

```
epoch 32 :  232.4752556002722
epoch 33 :  232.19492170300455
epoch 34 :  231.92252494703584
epoch 35 :  231.65750155357264
epoch 36 :  231.399343717926
epoch 37 :  231.14759260858503
epoch 38 :  230.90183239977884
epoch 39 :  230.6616851654793
epoch 40 :  230.4268064940328
epoch 41 :  230.19688170777243
epoch 42 :  229.97162259232138
epoch 43 :  229.75076455682094
epoch 44 :  229.53406415977543
epoch 45 :  229.3212969462053
epoch 46 :  229.112255550817
epoch 47 :  228.90674802931125
epoch 48 :  228.7045963860703
epoch 49 :  228.50563527151888
epoch 50 :  228.30971082664897
epoch 51 :  228.11667965568185
epoch 52 :  227.9264079107412
epoch 53 :  227.73877047483106
epoch 54 :  227.55365023143065
epoch 55 :  227.37093741070737
epoch 56 :  227.19052900375638
epoch 57 :  227.01232823746028
epoch 58 :  226.83624410354003
epoch 59 :  226.662190936199
epoch 60 :  226.49008803344256
epoch 61 :  226.31985931773224
epoch 62 :  226.151433032108
epoch 63 :  225.98474146830986
epoch 64 :  225.8197207237624
epoch 65 :  225.65631048455919
epoch 66 :  225.49445383181714
epoch 67 :  225.3340970689645
epoch 68 :  225.17518956769064
epoch 69 :  225.0176836304279
epoch 70 :  224.86153436735958
epoch 71 :  224.70669958605637
epoch 72 :  224.55313969194603
epoch 73 :  224.40081759791244
epoch 74 :  224.24969864141156
epoch 75 :  224.0997505075789
epoch 76 :  223.95094315689266
epoch 77 :  223.8032487560474
epoch 78 :  223.6566416107875
epoch 79 :  223.51109809954832
```

```
epoch 80 :  223.36659660685768
epoch 81 :  223.22311745555925
epoch 82 :  223.08064283703732
epoch 83 :  222.93915673874315
epoch 84 :  222.79864486845167
epoch 85 :  222.65909457481257
epoch 86 :  222.52049476389584
epoch 87 :  222.38283581157648
epoch 88 :  222.24610947174526
epoch 89 :  222.11030878047913
epoch 90 :  221.97542795644603
epoch 91 :  221.84146229796096
epoch 92 :  221.70840807724073
epoch 93 :  221.57626243253003
epoch 94 :  221.44502325888243
epoch 95 :  221.3146890984762
epoch 96 :  221.1852590314222
epoch 97 :  221.0567325680799
epoch 98 :  220.92910954392812
epoch 99 :  220.80239001804887
epoch 100 :  220.6765741762618
epoch 101 :  220.55166223990267
epoch 102 :  220.42765438116808
epoch 103 :  220.30455064585055
epoch 104 :  220.18235088416967
epoch 105 :  220.06105469026414
epoch 106 :  219.9406613507557
epoch 107 :  219.82116980262782
epoch 108 :  219.70257860049045
epoch 109 :  219.58488589312896
epoch 110 :  219.46808940906257
epoch 111 :  219.35218645068267
epoch 112 :  219.23717389638924
epoch 113 :  219.12304821001675
epoch 114 :  219.00980545673286
epoch 115 :  218.89744132450537
epoch 116 :  218.7859511501715
epoch 117 :  218.6753299491062
epoch 118 :  218.56557244747387
epoch 119 :  218.45667311605646
epoch 120 :  218.34862620468445
epoch 121 :  218.24142577634663
epoch 122 :  218.13506574012672
epoch 123 :  218.02953988219338
epoch 124 :  217.92484189416962
epoch 125 :  217.82096539830994
epoch 126 :  217.7179039690224
epoch 127 :  217.61565115039042
```

```
epoch 128 :  217.51420046946046
epoch 129 :  217.4135454451772
epoch 130 :  217.31367959295858
epoch 131 :  217.21459642500383
epoch 132 :  217.11628944652873
epoch 133 :  217.01875214820487
epoch 134 :  216.92197799515841
epoch 135 :  216.82596041294633
epoch 136 :  216.7306927709799
epoch 137 :  216.6361683638989
epoch 138 :  216.54238039142695
epoch 139 :  216.44932193724188
epoch 140 :  216.35698594739287
epoch 141 :  216.2653652087759
epoch 142 :  216.17445232814885
epoch 143 :  216.08423971212721
epoch 144 :  215.9947195485518
epoch 145 :  215.9058837895632
epoch 146 :  215.81772413665686
epoch 147 :  215.73023202793044
epoch 148 :  215.64339862766755
epoch 149 :  215.55721481834536
epoch 150 :  215.4716711950854
epoch 151 :  215.38675806252138
epoch 152 :  215.30246543399937
epoch 153 :  215.2187830329904
epoch 154 :  215.1357002965547
epoch 155 :  215.05320638067187
epoch 156 :  214.97129016723
epoch 157 :  214.8899402724499
epoch 158 :  214.80914505651947
epoch 159 :  214.72889263420606
epoch 160 :  214.6491708862217
epoch 161 :  214.56996747112493
epoch 162 :  214.4912698375508
epoch 163 :  214.41306523658005
epoch 164 :  214.33534073407037
epoch 165 :  214.258083222792
epoch 166 :  214.1812794342272
epoch 167 :  214.1049159499109
epoch 168 :  214.028979212207
epoch 169 :  213.95345553443144
epoch 170 :  213.87833111024875
epoch 171 :  213.80359202228234
epoch 172 :  213.7292242498929
epoch 173 :  213.65521367608946
epoch 174 :  213.58154609354878
epoch 175 :  213.5082072097273
```

```
epoch 176 :  213.43518265105698
epoch 177 :  213.36245796622322
epoch 178 :  213.29001862852826
epoch 179 :  213.21785003734706
epoch 180 :  213.14593751868637
epoch 181 :  213.07426632486093
epoch 182 :  213.00282163330004
epoch 183 :  212.93158854450266
epoch 184 :  212.86055207915572
epoch 185 :  212.78969717443334
epoch 186 :  212.71900867949358
epoch 187 :  212.64847135018692
epoch 188 :  212.57806984299373
epoch 189 :  212.50778870820153
epoch 190 :  212.43761238233654
epoch 191 :  212.36752517985866
epoch 192 :  212.29751128412965
epoch 193 :  212.22755473766134
epoch 194 :  212.15763943164893
epoch 195 :  212.08774909479388
epoch 196 :  212.01786728141596
epoch 197 :  211.9479773588565
epoch 198 :  211.87806249416764
epoch 199 :  211.80810564008618
epoch 200 :  211.73808952028395
epoch 201 :  211.6679966138876
epoch 202 :  211.59780913925883
epoch 203 :  211.52750903702338
epoch 204 :  211.4570779523367
epoch 205 :  211.38649721637353
epoch 206 :  211.31574782702543
epoch 207 :  211.2448104287923
epoch 208 :  211.17366529185216
epoch 209 :  211.10229229029113
epoch 210 :  211.0306708794817
epoch 211 :  210.95878007258986
epoch 212 :  210.88659841619992
epoch 213 :  210.81410396504342
epoch 214 :  210.74127425582077
epoch 215 :  210.66808628010898
epoch 216 :  210.59451645635136
epoch 217 :  210.52054060092746
epoch 218 :  210.446133898311
epoch 219 :  210.3712708703251
epoch 220 :  210.29592534451464
epoch 221 :  210.2200704216631
epoch 222 :  210.1436784424894
epoch 223 :  210.06672095357447
```

```
epoch 224 :  209.98916867257708
epoch 225 :  209.91099145281285
epoch 226 :  209.83215824728708
epoch 227 :  209.75263707228785
epoch 228 :  209.67239497066294
epoch 229 :  209.5913979749272
epoch 230 :  209.50961107036233
epoch 231 :  209.4269981582969
epoch 232 :  209.34352201977154
epoch 233 :  209.25914427981806
epoch 234 :  209.1738253725984
epoch 235 :  209.08752450766568
epoch 236 :  209.00019963762486
epoch 237 :  208.911807427475
epoch 238 :  208.82230322591647
epoch 239 :  208.73164103889556
epoch 240 :  208.63977350563695
epoch 241 :  208.54665187737646
epoch 242 :  208.4522259989491
epoch 243 :  208.3564442933089
epoch 244 :  208.25925374895257
epoch 245 :  208.16059991008515
epoch 246 :  208.06042686920063
epoch 247 :  207.9586772615498
epoch 248 :  207.85529226073413
epoch 249 :  207.75021157439244
epoch 250 :  207.64337343864702
epoch 251 :  207.5347146096443
epoch 252 :  207.42417035017826
epoch 253 :  207.31167440902752
epoch 254 :  207.19715899028918
epoch 255 :  207.08055470967167
epoch 256 :  206.96179053443427
epoch 257 :  206.8407937034623
epoch 258 :  206.71748962386505
epoch 259 :  206.59180174050078
epoch 260 :  206.4636513750029
epoch 261 :  206.33295753119688
epoch 262 :  206.19963666429084
epoch 263 :  206.06360241186962
epoch 264 :  205.92476528551876
epoch 265 :  205.78303232281039
epoch 266 :  205.6383067003681
epoch 267 :  205.49048730971995
epoch 268 :  205.33946829860065
epoch 269 :  205.18513858120218
epoch 270 :  205.02738132153135
epoch 271 :  204.86607339446252
```

```
epoch 272 :  204.70108482922973
epoch 273 :  204.53227823995394
epoch 274 :  204.35950824735755
epoch 275 :  204.18262089507672
epoch 276 :  204.00145306299504
epoch 277 :  203.815831878823
epoch 278 :  203.6255741278099
epoch 279 :  203.43048565904883
epoch 280 :  203.2303607853914
epoch 281 :  203.02498167258372
epoch 282 :  202.81411771190656
epoch 283 :  202.59752486940295
epoch 284 :  202.374945003715
epoch 285 :  202.14610514365413
epoch 286 :  201.91071671590333
epoch 287 :  201.66847471268846
epoch 288 :  201.41905678887593
epoch 289 :  201.16212227773346
epoch 290 :  200.89731111455507
epoch 291 :  200.6242426574829
epoch 292 :  200.34251439518837
epoch 293 :  200.0517005316059
epoch 294 :  199.75135043868121
epoch 295 :  199.440986969133
epoch 296 :  199.1201046225903
epoch 297 :  198.78816756021965
epoch 298 :  198.44460746519545
epoch 299 :  198.0888212492103
epoch 300 :  197.72016860882832
epoch 301 :  197.33796944005962
epoch 302 :  196.94150112533126
epoch 303 :  196.52999571438312
epoch 304 :  196.10263702993427
epoch 305 :  195.65855774074333
epoch 306 :  195.1968364595737
epoch 307 :  194.71649494231167
epoch 308 :  194.2164954880441
epoch 309 :  193.69573866941823
epoch 310 :  193.153061559552
epoch 311 :  192.5872366679476
epoch 312 :  191.9969718556485
epoch 313 :  191.38091157239873
epoch 314 :  190.73763985004913
epoch 315 :  190.0656856027986
epoch 316 :  189.36353093430833
epoch 317 :  188.62962334583818
epoch 318 :  187.8623929942393
epoch 319 :  187.06027648519486
```

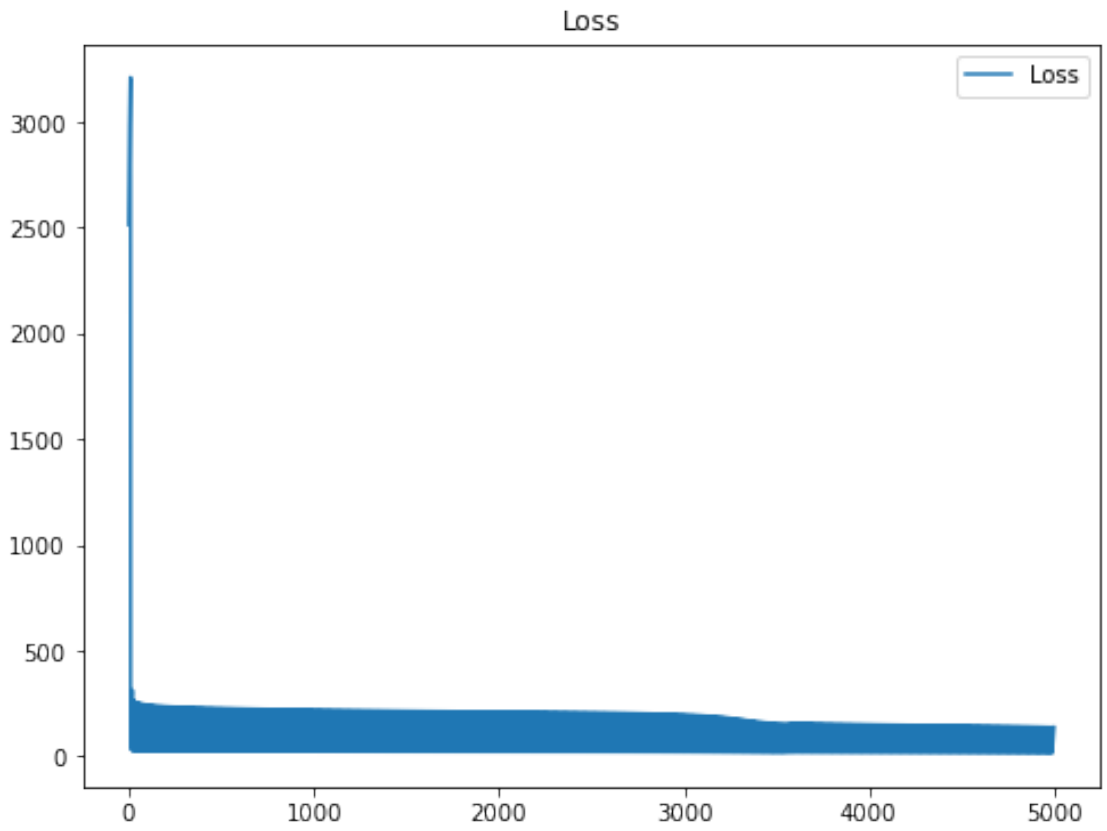
```
epoch 320 :  186.2217491327094
epoch 321 :  185.34536820272373
epoch 322 :  184.42983041963117
epoch 323 :  183.47404796860803
epoch 324 :  182.47724834016734
epoch 325 :  181.439104447639
epoch 326 :  180.35990193035667
epoch 327 :  179.24074905319435
epoch 328 :  178.08382840411213
epoch 329 :  176.89267458745218
epoch 330 :  175.6724348624142
epoch 331 :  174.4300331230247
epoch 332 :  173.17412896161784
epoch 333 :  171.91477449107356
epoch 334 :  170.66274738439208
epoch 335 :  169.42866211629104
epoch 336 :  168.22206375684232
epoch 337 :  167.05071969024814
epoch 338 :  165.9202405179698
epoch 339 :  164.83404683471463
epoch 340 :  163.79361980727325
epoch 341 :  162.79894461578002
epoch 342 :  161.84904859696982
epoch 343 :  160.9425242478909
epoch 344 :  160.07791779453777
epoch 345 :  159.25389596210914
epoch 346 :  158.469223817121
epoch 347 :  157.72281021138866
epoch 348 :  157.01434874073746
epoch 349 :  156.34626961023972
epoch 350 :  155.72768314208284
epoch 351 :  155.18078454210345
epoch 352 :  154.7501994451639
epoch 353 :  154.51483208917776
epoch 354 :  154.5869981487333
epoch 355 :  155.03235356606075
epoch 356 :  155.69303498746632
epoch 357 :  156.28095599937478
epoch 358 :  156.70662844966824
epoch 359 :  156.97592778827809
epoch 360 :  157.10323967442588
epoch 361 :  157.1188863612652
epoch 362 :  157.05726615174282
epoch 363 :  156.94721393870873
epoch 364 :  156.8089354074807
epoch 365 :  156.65512473843853
epoch 366 :  156.4932258588455
epoch 367 :  156.32743681370744
```

```
epoch 368 :  156.16010123069975
epoch 369 :  155.99255118852423
epoch 370 :  155.8255711941719
epoch 371 :  155.65963769307183
epoch 372 :  155.4950405079537
epoch 373 :  155.3319477581375
epoch 374 :  155.1704449828431
epoch 375 :  155.01056183483286
epoch 376 :  154.8522915444376
epoch 377 :  154.69560519321908
epoch 378 :  154.54046187765613
epoch 379 :  154.38681562288002
epoch 380 :  154.23461982872107
epoch 381 :  154.08382992920983
epoch 382 :  153.9344048153128
epoch 383 :  153.7863074368072
epoch 384 :  153.63950488271425
epoch 385 :  153.49396814829615
epoch 386 :  153.34967172896455
epoch 387 :  153.20659313342722
epoch 388 :  153.06471237513426
epoch 389 :  152.92401147829736
epoch 390 :  152.78447401955546
epoch 391 :  152.64608471614642
epoch 392 :  152.50882906493342
epoch 393 :  152.37269303240194
epoch 394 :  152.23766279327901
epoch 395 :  152.10372451395523
epoch 396 :  151.97086417625223
epoch 397 :  151.83906743681348
epoch 398 :  151.7083195175073
epoch 399 :  151.5786051224772
epoch 400 :  151.44990837780162
epoch 401 :  151.32221279015562
epoch 402 :  151.19550122124036
epoch 403 :  151.06975587512892
epoch 404 :  150.94495829608087
epoch 405 :  150.82108937467237
epoch 406 :  150.69812936040955
epoch 407 :  150.5760578792683
epoch 408 :  150.45485395483652
epoch 409 :  150.3344960319293
epoch 410 :  150.21496200176583
epoch 411 :  150.0962292279116
epoch 412 :  149.97827457236653
epoch 413 :  149.8610744212512
epoch 414 :  149.74460470968617
epoch 415 :  149.62884094552635
```

```
epoch 416 :  149.5137582316568
epoch 417 :  149.39933128666797
epoch 418 :  149.28553446373644
epoch 419 :  149.17234176758495
epoch 420 :  149.0597268694627
epoch 421 :  148.9476631200399
epoch 422 :  148.83612356023653
epoch 423 :  148.72508092990674
epoch 424 :  148.61450767442034
epoch 425 :  148.50437594909698
epoch 426 :  148.3946576215418
epoch 427 :  148.28532427185368
epoch 428 :  148.17634719074923
epoch 429 :  148.06769737559642
epoch 430 :  147.95934552437038
epoch 431 :  147.85126202754708
epoch 432 :  147.74341695792145
epoch 433 :  147.63578005837223
epoch 434 :  147.52832072752847
epoch 435 :  147.42100800336573
epoch 436 :  147.31381054467303
epoch 437 :  147.20669661038073
epoch 438 :  147.09963403669826
epoch 439 :  146.99259021202707
epoch 440 :  146.88553204956244
epoch 441 :  146.77842595755547
epoch 442 :  146.67123780710472
epoch 443 :  146.56393289743403
epoch 444 :  146.45647591850036
epoch 445 :  146.3488309108582
epoch 446 :  146.24096122260175
epoch 447 :  146.132829463273
epoch 448 :  146.024397454523
epoch 449 :  145.91562617738794
epoch 450 :  145.80647571591237
epoch 451 :  145.69690519696036
epoch 452 :  145.58687272587056
epoch 453 :  145.4763353177818
epoch 454 :  145.36524882419752
epoch 455 :  145.25356785457043
epoch 456 :  145.14124569241827
epoch 457 :  145.02823420565824
epoch 458 :  144.91448375061168
epoch 459 :  144.7999430692669
epoch 460 :  144.68455917915753
epoch 461 :  144.5682772553394
epoch 462 :  144.45104050371242
epoch 463 :  144.3327900250143
```

```
epoch 464 :  144.2134646686264
epoch 465 :  144.09300087531494
epoch 466 :  143.97133250790603
epoch 467 :  143.8483906687931
epoch 468 :  143.72410350306419
epoch 469 :  143.59839598589596
epoch 470 :  143.4711896927231
epoch 471 :  143.3424025505296
epoch 472 :  143.2119485684077
epoch 473 :  143.07973754536079
epoch 474 :  142.94567475304342
epoch 475 :  142.8096605909399
epoch 476 :  142.67159021112687
epoch 477 :  142.53135310948386
epoch 478 :  142.38883267984323
epoch 479 :  142.2439057271397
epoch 480 :  142.09644193517624
epoch 481 :  141.94630328411404
epoch 482 :  141.7933434121685
epoch 483 :  141.6374069153996
epoch 484 :  141.47832857870745
epoch 485 :  141.31593253032838
epoch 486 :  141.15003131124692
epoch 487 :  140.98042484987104
epoch 488 :  140.80689933122852
epoch 489 :  140.6292259486671
epoch 490 :  140.44715952470455
epoch 491 :  140.2604369861494
epoch 492 :  140.0687756770673
epoch 493 :  139.8718714914385
epoch 494 :  139.66939680564633
epoch 495 :  139.46099818917503
epoch 496 :  139.2462938702597
epoch 497 :  139.0248709318162
epoch 498 :  138.7962822119798
epoch 499 :  138.56004288332645
CPU times: user 12min 16s, sys: 2.09 s, total: 12min 18s
Wall time: 1h 11min 57s
```

[417]:
```
plt.figure(figsize = (8, 6))
plt.plot(Loss, label = 'Loss', markersize = 3)
plt.title("Loss")
plt.legend()
plt.show()
```



[418]:
```
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 :  138.315626685726
epoch 501 :  138.06246179039587
epoch 502 :  137.79992627792706
epoch 503 :  137.5273432220322
epoch 504 :  137.24397538511045
epoch 505 :  136.94901955375062
epoch 506 :  136.64160057503977
epoch 507 :  136.32076520218627
epoch 508 :  135.9854759261154
epoch 509 :  135.63460506574825
epoch 510 :  135.26692952312484
epoch 511 :  134.88112679261596
epoch 512 :  134.47577306070582
epoch 513 :  134.04934456120225
```

```
epoch 514 :  133.6002237775327
epoch 515 :  133.12671262337722
epoch 516 :  132.62705538923734
epoch 517 :  132.0994749968147
epoch 518 :  131.54222689238438
epoch 519 :  130.9536755970317
epoch 520 :  130.33239926112972
epoch 521 :  129.6773271259552
epoch 522 :  128.98791296744315
epoch 523 :  128.2643436120253
epoch 524 :  127.50777469126913
epoch 525 :  126.72057552561449
epoch 526 :  125.90655198530239
epoch 527 :  125.07110274737077
epoch 528 :  124.22125524093914
epoch 529 :  123.36552923393691
epoch 530 :  122.51359443167469
epoch 531 :  121.67572508890102
epoch 532 :  120.86210264686471
epoch 533 :  120.08206145493205
epoch 534 :  119.34339460395584
epoch 535 :  118.65182572904294
epoch 536 :  118.01071199351912
epoch 537 :  117.42099093793938
epoch 538 :  116.88134125134057
epoch 539 :  116.38850819294709
epoch 540 :  115.93774601878553
epoch 541 :  115.52333693022177
epoch 542 :  115.13914237955044
epoch 543 :  114.77912559958769
epoch 544 :  114.4377709396647
epoch 545 :  114.11033899540547
epoch 546 :  113.7929416005738
epoch 547 :  113.48247482576078
epoch 548 :  113.17647933045924
epoch 549 :  112.87299236297336
epoch 550 :  112.5704271312289
epoch 551 :  112.26748602860589
epoch 552 :  111.96309847429059
epoch 553 :  111.65637217896527
epoch 554 :  111.34655128852299
epoch 555 :  111.03297949759262
epoch 556 :  110.7150682727609
epoch 557 :  110.39227029523924
epoch 558 :  110.06405752703805
epoch 559 :  109.7299028287622
epoch 560 :  109.38926399941386
epoch 561 :  109.04156929044379
```

```
epoch 562 :  108.68620366372048
epoch 563 :  108.32249521957884
epoch 564 :  107.94970131142072
epoch 565 :  107.56699391825914
epoch 566 :  107.17344389984147
epoch 567 :  106.7680038431484
epoch 568 :  106.34948936986395
epoch 569 :  105.91655909586646
epoch 570 :  105.46769407880888
epoch 571 :  105.00117887428341
epoch 572 :  104.51508885439434
epoch 573 :  104.00729339044224
epoch 574 :  103.47549403792394
epoch 575 :  102.9173347905563
epoch 576 :  102.33065340780524
epoch 577 :  101.71399286372215
epoch 578 :  101.06754452582577
epoch 579 :  100.3946511371346
epoch 580 :  99.70353480133576
epoch 581 :  99.00746580192398
epoch 582 :  98.31959122553192
epoch 583 :  97.64091097470258
epoch 584 :  96.95184557129522
epoch 585 :  96.2237871717735
epoch 586 :  95.44174300396648
epoch 587 :  94.60851262889804
epoch 588 :  93.73257307089709
epoch 589 :  92.82222936882614
epoch 590 :  91.88629958756094
epoch 591 :  90.93425305140174
epoch 592 :  89.97546514873503
epoch 593 :  89.01853504705015
epoch 594 :  88.0708881650843
epoch 595 :  87.1386641299597
epoch 596 :  86.22678225227861
epoch 597 :  85.33907023683173
epoch 598 :  84.47839048490903
epoch 599 :  83.64674736693857
epoch 600 :  82.84538333350453
epoch 601 :  82.07487453774718
epoch 602 :  81.33523025614502
epoch 603 :  80.6259941156654
epoch 604 :  79.94634225745033
epoch 605 :  79.29517365935742
epoch 606 :  78.67118939621356
epoch 607 :  78.07295942637381
epoch 608 :  77.49897694432511
epoch 609 :  76.94770126750488
```

```
epoch 610 :  76.41759069384473
epoch 611 :  75.90712691007721
epoch 612 :  75.41483247993244
epoch 613 :  74.93928279349853
epoch 614 :  74.47911367603129
epoch 615 :  74.03302566944542
epoch 616 :  73.59978582674209
epoch 617 :  73.17822770270665
epoch 618 :  72.76725008371565
epoch 619 :  72.36581487629542
epoch 620 :  71.97294447123942
epoch 621 :  71.58771882326693
epoch 622 :  71.20927244353383
epoch 623 :  70.83679150480064
epoch 624 :  70.46951132254242
epoch 625 :  70.10671462506532
epoch 626 :  69.7477313059304
epoch 627 :  69.3919408443601
epoch 628 :  69.0387794392062
epoch 629 :  68.6877554315781
epoch 630 :  68.33847938855311
epoch 631 :  67.99072049203816
epoch 632 :  67.64451109618176
epoch 633 :  67.30034160255981
epoch 634 :  66.95952865915241
epoch 635 :  66.62492153233066
epoch 636 :  66.3022667456407
epoch 637 :  66.00278954145651
epoch 638 :  65.747608684438
epoch 639 :  65.57281340563674
epoch 640 :  65.52475678802232
epoch 641 :  65.61632328530717
epoch 642 :  65.74788863386779
epoch 643 :  65.75165860110734
epoch 644 :  65.58669596610389
epoch 645 :  65.3300311237063
epoch 646 :  65.03981656224406
epoch 647 :  64.7376787975277
epoch 648 :  64.43111782056698
epoch 649 :  64.12314777963354
epoch 650 :  63.81505883763975
epoch 651 :  63.507352741231536
epoch 652 :  63.200160286109906
epoch 653 :  62.89345630589182
epoch 654 :  62.58717305221007
epoch 655 :  62.281257348505925
epoch 656 :  61.97569604464695
epoch 657 :  61.67052384043524
```

```
epoch 658 :  61.36582166227853
epoch 659 :  61.061710349916865
epoch 660 :  60.758342388969055
epoch 661 :  60.45589322738193
epoch 662 :  60.15455300456478
epoch 663 :  59.85451910490022
epoch 664 :  59.555989705636875
epoch 665 :  59.25915835265519
epoch 666 :  58.964209523528595
epoch 667 :  58.671315099802655
epoch 668 :  58.380631653769285
epoch 669 :  58.092298449947066
epoch 670 :  57.806436062490604
epoch 671 :  57.523145513780456
epoch 672 :  57.242507844804265
epoch 673 :  56.96458403375702
epoch 674 :  56.68941518510056
epoch 675 :  56.41702291698227
epoch 676 :  56.14740988021086
epoch 677 :  55.88056034695845
epoch 678 :  55.616440811844114
epoch 679 :  55.35500055208231
epoch 680 :  55.09617209691507
epoch 681 :  54.83987155969788
epoch 682 :  54.58599878905421
epoch 683 :  54.33443729904108
epoch 684 :  54.08505394354215
epoch 685 :  53.83769830934687
epoch 686 :  53.59220181994493
epoch 687 :  53.3483765757206
epoch 688 :  53.10601402038949
epoch 689 :  52.86488364447698
epoch 690 :  52.62473216299696
epoch 691 :  52.385284025977846
epoch 692 :  52.1462449036843
epoch 693 :  51.90731124167803
epoch 694 :  51.6681916700687
epoch 695 :  51.42865098782598
epoch 696 :  51.1885963280529
epoch 697 :  50.94824040363932
epoch 698 :  50.70840048279752
epoch 699 :  50.47101945172403
epoch 700 :  50.239994502118535
epoch 701 :  50.02225454943728
epoch 702 :  49.828474365018735
epoch 703 :  49.67164713731473
epoch 704 :  49.56112410920909
epoch 705 :  49.493659399505376
```

```
epoch 706 :  49.451302124799646
epoch 707 :  49.41257457539562
epoch 708 :  49.36485848151088
epoch 709 :  49.305473972701954
epoch 710 :  49.23624000821923
epoch 711 :  49.15955406324072
epoch 712 :  49.07722886810538
epoch 713 :  48.99050175176744
epoch 714 :  48.90023199304663
epoch 715 :  48.80704159037283
epoch 716 :  48.71139725206828
epoch 717 :  48.61365907719644
epoch 718 :  48.51411129020951
epoch 719 :  48.412982681489964
epoch 720 :  48.3104607171891
epoch 721 :  48.2067015340912
epoch 722 :  48.10183714859171
epoch 723 :  47.99598072499021
epoch 724 :  47.88923046505877
epoch 725 :  47.781672504753985
epoch 726 :  47.67338308910062
epoch 727 :  47.564430218632744
epoch 728 :  47.45487490697814
epoch 729 :  47.34477215114554
epoch 730 :  47.234171688855724
epoch 731 :  47.12311859760289
epoch 732 :  47.01165377575925
epoch 733 :  46.89981433554525
epoch 734 :  46.7876339299267
epoch 735 :  46.67514302976056
epoch 736 :  46.56236916327496
epoch 737 :  46.44933712678479
epoch 738 :  46.33606917321625
epoch 739 :  46.22258518324496
epoch 740 :  46.108902822568986
epoch 741 :  45.99503768786461
epoch 742 :  45.881003443279305
epoch 743 :  45.76681194877691
epoch 744 :  45.652473381293994
epoch 745 :  45.53799634937683
epoch 746 :  45.42338800179213
epoch 747 :  45.30865413045098
epoch 748 :  45.19379926791822
epoch 749 :  45.07882677969389
epoch 750 :  44.96373895143914
epoch 751 :  44.84853707127905
epoch 752 :  44.73322150731588
epoch 753 :  44.617791780477084
```

```
epoch 754 :  44.50224663282265
epoch 755 :  44.38658409144591
epoch 756 :  44.27080152809912
epoch 757 :  44.15489571468608
epoch 758 :  44.038862874764085
epoch 759 :  43.92269873120971
epoch 760 :  43.80639855019075
epoch 761 :  43.68995718160847
epoch 762 :  43.57336909615426
epoch 763 :  43.45662841913948
epoch 764 :  43.339728961256114
epoch 765 :  43.222664246397315
epoch 766 :  43.105427536712476
epoch 767 :  42.988011855011344
epoch 768 :  42.87041000467195
epoch 769 :  42.75261458717382
epoch 770 :  42.6346180173873
epoch 771 :  42.51641253673488
epoch 772 :  42.39799022433926
epoch 773 :  42.27934300626347
epoch 774 :  42.160462662942486
epoch 775 :  42.041340834902165
epoch 776 :  41.9219690268463
epoch 777 :  41.80233861019924
epoch 778 :  41.6824408241703
epoch 779 :  41.562266775410336
epoch 780 :  41.44180743632365
epoch 781 :  41.32105364208876
epoch 782 :  41.199996086433714
epoch 783 :  41.07862531621944
epoch 784 :  40.95693172486302
epoch 785 :  40.834905544636825
epoch 786 :  40.7125368378783
epoch 787 :  40.589815487129975
epoch 788 :  40.466731184237446
epoch 789 :  40.34327341841941
epoch 790 :  40.21943146332826
epoch 791 :  40.095194363116924
epoch 792 :  39.970550917512035
epoch 793 :  39.84548966591113
epoch 794 :  39.71999887050997
epoch 795 :  39.59406649845303
epoch 796 :  39.467680203027456
epoch 797 :  39.340827303889384
epoch 798 :  39.21349476633181
epoch 799 :  39.085669179593985
epoch 800 :  38.95733673421808
epoch 801 :  38.82848319844854
```

```
epoch 802 :  38.69909389369193
epoch 803 :  38.56915366902914
epoch 804 :  38.43864687480167
epoch 805 :  38.30755733527274
epoch 806 :  38.17586832038653
epoch 807 :  38.0435625166391
epoch 808 :  37.91062199708818
epoch 809 :  37.777028190528426
epoch 810 :  37.64276184987074
epoch 811 :  37.50780301976465
epoch 812 :  37.3721310035213
epoch 813 :  37.23572432939227
epoch 814 :  37.09856071628071
epoch 815 :  36.96061703896545
epoch 816 :  36.82186929294097
epoch 817 :  36.68229255898067
epoch 818 :  36.541860967563004
epoch 819 :  36.400547663303314
epoch 820 :  36.25832476957089
epoch 821 :  36.11516335348337
epoch 822 :  35.971033391508286
epoch 823 :  35.82590373591876
epoch 824 :  35.67974208239843
epoch 825 :  35.53251493911595
epoch 826 :  35.384187597628625
epoch 827 :  35.2347241060299
epoch 828 :  35.084087244786836
epoch 829 :  34.93223850577568
epoch 830 :  34.779138075075515
epoch 831 :  34.624744820128754
epoch 832 :  34.46901628195245
epoch 833 :  34.31190867313438
epoch 834 :  34.153376882417476
epoch 835 :  33.99337448674104
epoch 836 :  33.83185377166609
epoch 837 :  33.66876576117837
epoch 838 :  33.504060257908044
epoch 839 :  33.33768589485691
epoch 840 :  33.16959019974872
epoch 841 :  32.999719673130365
epoch 842 :  32.82801988134057
epoch 843 :  32.654435565410886
epoch 844 :  32.4789107668742
epoch 845 :  32.30138897131056
epoch 846 :  32.1218132702495
epoch 847 :  31.94012654174602
epoch 848 :  31.756271649550317
epoch 849 :  31.570191660269057
```

```
epoch 850 :  31.381830077244697
epoch 851 :  31.191131089033007
epoch 852 :  30.998039829316628
epoch 853 :  30.802502643827538
epoch 854 :  30.604467358336652
epoch 855 :  30.40388354003164
epoch 856 :  30.20070274267479
epoch 857 :  29.994878723961456
epoch 858 :  29.786367621821803
epoch 859 :  29.575128075711813
epoch 860 :  29.361121280707955
epoch 861 :  29.14431096942983
epoch 862 :  28.924663335716073
epoch 863 :  28.702146957593662
epoch 864 :  28.47673287338521
epoch 865 :  28.24839517543602
epoch 866 :  28.017112951309535
epoch 867 :  27.782875455502875
epoch 868 :  27.5456948635642
epoch 869 :  27.30563701768227
epoch 870 :  27.062896291765536
epoch 871 :  26.81798433581803
epoch 872 :  26.572233525014784
epoch 873 :  26.32924649139649
epoch 874 :  26.09947286036406
epoch 875 :  25.91598564383512
epoch 876 :  25.88809722027601
epoch 877 :  26.286098130299145
epoch 878 :  27.021621808223372
epoch 879 :  26.86807391716388
epoch 880 :  26.297155591101752
epoch 881 :  25.994485911351024
epoch 882 :  25.721744848614264
epoch 883 :  25.46128218822931
epoch 884 :  25.211238368566303
epoch 885 :  24.96932970295621
epoch 886 :  24.73368662506218
epoch 887 :  24.503074181647584
epoch 888 :  24.27666774332328
epoch 889 :  24.053899944195486
epoch 890 :  23.834382797689198
epoch 891 :  23.617858221204557
epoch 892 :  23.40416189536991
epoch 893 :  23.19319654189283
epoch 894 :  22.984912189916454
epoch 895 :  22.779291542509988
epoch 896 :  22.576339101243953
epoch 897 :  22.376073094331623
```

```
epoch 898 :  22.178519511165366
epoch 899 :  21.98370772737728
epoch 900 :  21.791667335973184
epoch 901 :  21.602425895977575
epoch 902 :  21.41600738037624
epoch 903 :  21.232431157012325
epoch 904 :  21.051711374713094
epoch 905 :  20.873856655829286
epoch 906 :  20.698870018273368
epoch 907 :  20.52674896687744
epoch 908 :  20.357485706851595
epoch 909 :  20.19106744222554
epoch 910 :  20.0274767301874
epoch 911 :  19.866691868580943
epoch 912 :  19.70868729897886
epoch 913 :  19.55343401186896
epoch 914 :  19.400899943868527
epoch 915 :  19.251050359590238
epoch 916 :  19.103848212994123
epoch 917 :  18.95925448482508
epoch 918 :  18.817228494150857
epoch 919 :  18.67772818312793
epoch 920 :  18.54071037498012
epoch 921 :  18.406131005827085
epoch 922 :  18.273945331481556
epoch 923 :  18.144108110655672
epoch 924 :  18.016573766249113
epoch 925 :  17.891296526496163
epoch 926 :  17.76823054780718
epoch 927 :  17.647330021129758
epoch 928 :  17.52854926358612
epoch 929 :  17.411842797077618
epoch 930 :  17.297165415419215
epoch 931 :  17.184472241460355
epoch 932 :  17.073718775509995
epoch 933 :  16.9648609362529
epoch 934 :  16.857855095224053
epoch 935 :  16.75265810577412
epoch 936 :  16.649227327339702
epoch 937 :  16.547520645740665
epoch 938 :  16.447496490102274
epoch 939 :  16.34911384692986
epoch 940 :  16.252332271771394
epoch 941 :  16.157111898839787
epoch 942 :  16.063413448898324
epoch 943 :  15.971198235661229
epoch 944 :  15.8804281709162
epoch 945 :  15.791065768537518
```

```
epoch 946 :  15.703074147518919
epoch 947 :  15.616417034138589
epoch 948 :  15.531058763336729
epoch 949 :  15.446964279375617
epoch 950 :  15.364099135831868
epoch 951 :  15.282429494963758
epoch 952 :  15.201922126488634
epoch 953 :  15.122544405792006
epoch 954 :  15.044264311597644
epoch 955 :  14.967050423111827
epoch 956 :  14.890871916666065
epoch 957 :  14.815698561868551
epoch 958 :  14.741500717289364
epoch 959 :  14.668249325690567
epoch 960 :  14.595915908822558
epoch 961 :  14.524472561805721
epoch 962 :  14.453891947120683
epoch 963 :  14.384147288223671
epoch 964 :  14.315212362816405
epoch 965 :  14.247061495790591
epoch 966 :  14.179669551876682
epoch 967 :  14.113011928020288
epoch 968 :  14.047064545515715
epoch 969 :  13.981803841925672
epoch 970 :  13.917206762814198
epoch 971 :  13.85325075332433
epoch 972 :  13.789913749628536
epoch 973 :  13.727174170284206
epoch 974 :  13.665010907522163
epoch 975 :  13.60340331849829
epoch 976 :  13.542331216539031
epoch 977 :  13.481774862409948
epoch 978 :  13.421714955635009
epoch 979 :  13.362132625896656
epoch 980 :  13.303009424542697
epoch 981 :  13.244327316229223
epoch 982 :  13.186068670723317
epoch 983 :  13.128216254894435
epoch 984 :  13.07075322491481
epoch 985 :  13.013663118695819
epoch 986 :  12.956929848582716
epoch 987 :  12.900537694327522
epoch 988 :  12.844471296360926
epoch 989 :  12.788715649385834
epoch 990 :  12.733256096305148
epoch 991 :  12.678078322504875
epoch 992 :  12.623168350507768
epoch 993 :  12.568512535009493
```

```
epoch 994 :  12.514097558311878
epoch 995 :  12.459910426164654
epoch 996 :  12.405938464024942
epoch 997 :  12.352169313741987
epoch 998 :  12.298590930676902
epoch 999 :  12.245191581257204
```

[427]:
```
for i in range(1000,1100):
    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 1000 :  12.19195984097361
epoch 1001 :  12.138884592820517
epoch 1002 :  12.08595502617218
epoch 1003 :  12.033160636103236
epoch 1004 :  11.980491223140051
epoch 1005 :  11.927936893438176
epoch 1006 :  11.875488059378165
epoch 1007 :  11.823135440567373
epoch 1008 :  11.7708700652328
epoch 1009 :  11.718683271989255
epoch 1010 :  11.666566711965269
epoch 1011 :  11.614512351263814
epoch 1012 :  11.5625124737337
epoch 1013 :  11.510559684026784
epoch 1014 :  11.458646910910474
epoch 1015 :  11.406767410804173
epoch 1016 :  11.354914771506364
epoch 1017 :  11.303082916074164
epoch 1018 :  11.251266106817534
epoch 1019 :  11.19945894936552
epoch 1020 :  11.147656396763667
epoch 1021 :  11.095853753552133
epoch 1022 :  11.044046679783266
epoch 1023 :  10.992231194923765
epoch 1024 :  10.940403681595207
epoch 1025 :  10.888560889099333
epoch 1026 :  10.836699936676782
epoch 1027 :  10.7848183164446
epoch 1028 :  10.732913895963556
epoch 1029 :  10.680984920377911
epoch 1030 :  10.62903001407905
epoch 1031 :  10.577048181840565
epoch 1032 :  10.52503880937743
epoch 1033 :  10.473001663278836
```

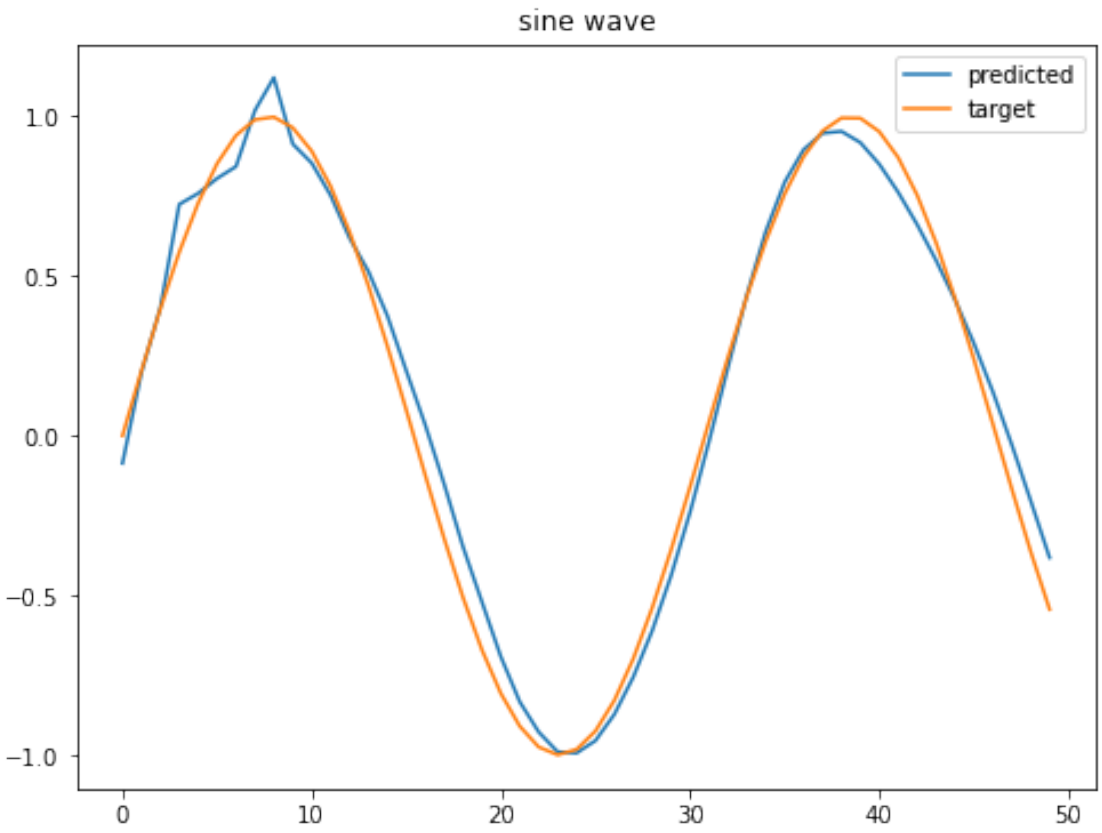
```
epoch 1034 :  10.420936890273092
epoch 1035 :  10.368845015781947
epoch 1036 :  10.316726941725198
epoch 1037 :  10.264583943540057
epoch 1038 :  10.212417666390179
epoch 1039 :  10.160230120532288
epoch 1040 :  10.10802367582348
epoch 1041 :  10.055801055355618
epoch 1042 :  10.003565328204024
epoch 1043 :  9.951319901289958
epoch 1044 :  9.899068510359154
epoch 1045 :  9.846815210085136
epoch 1046 :  9.794564363314358
epoch 1047 :  9.74232062947373
epoch 1048 :  9.690088952171019
epoch 1049 :  9.637874546020939
epoch 1050 :  9.585682882738254
epoch 1051 :  9.533519676546334
epoch 1052 :  9.481390868950411
epoch 1053 :  9.429302612933409
epoch 1054 :  9.377261256638171
epoch 1055 :  9.325273326597937
epoch 1056 :  9.273345510586795
epoch 1057 :  9.221484640161414
epoch 1058 :  9.169697672969066
epoch 1059 :  9.117991674895938
epoch 1060 :  9.066373802133969
epoch 1061 :  9.014851283243154
epoch 1062 :  8.963431401284698
epoch 1063 :  8.91212147610439
epoch 1064 :  8.860928846832627
epoch 1065 :  8.809860854678702
epoch 1066 :  8.758924826089087
epoch 1067 :  8.708128056325469
epoch 1068 :  8.657477793536337
epoch 1069 :  8.60698122336632
epoch 1070 :  8.556645454167255
epoch 1071 :  8.506477502846808
epoch 1072 :  8.456484281407445
epoch 1073 :  8.406672584200921
epoch 1074 :  8.357049075939297
epoch 1075 :  8.307620280482295
epoch 1076 :  8.258392570419831
epoch 1077 :  8.209372157468948
epoch 1078 :  8.160565083690283
epoch 1079 :  8.111977213527677
epoch 1080 :  8.063614226673057
epoch 1081 :  8.015481611746193
```

```
epoch 1082 :  7.967584660784167
epoch 1083 :  7.91992846452108
epoch 1084 :  7.872517908443099
epoch 1085 :  7.825357669597428
epoch 1086 :  7.778452214128967
epoch 1087 :  7.7318057955168715
epoch 1088 :  7.685422453486375
epoch 1089 :  7.6393060135573325
epoch 1090 :  7.593460087201837
epoch 1091 :  7.547888072576462
epoch 1092 :  7.502593155788504
epoch 1093 :  7.457578312670176
epoch 1094 :  7.412846311015064
epoch 1095 :  7.368399713247234
epoch 1096 :  7.324240879486027
epoch 1097 :  7.280371970969845
epoch 1098 :  7.236794953806129
epoch 1099 :  7.193511603013769
```

[450]:
```
#parameters = np.ones((3,1))
T = 50
parameters = np.array([2,0,1]).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)


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()
```

```
50
```



[390]:
```
parameters = np.random.randint(3,5)
fix = np.array([0,1])
x = np.insert(fix,0,parameters).reshape(3,1)
```

[390]: 3

[395]:
```
x = np.insert(fix,0,parameters).reshape(3,1)
```

[396]:
```
x.shape
```

[396]: (3, 1)

[ ]:
```

```