Artifical neural network - Homework3

Purusothaman Seenivasan CID: pursee, Personal number: 19970421-8396

October 2023

1 Chaotic time-series prediction 2023

1.1 Program

Listing 1: Python Code

```
import numpy as np
import csv
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D

input = []

with open('training-set.csv', 'r') as f:
  csv_reader = csv.reader(f)
  for row in csv_reader:
    input.append((row))
input = np.array([[float(item) for item in sublist] for sublist in input])
input = input.transpose()

noOfInput = len(input)  #number OF T
numberOfNeuron = 500
reservoirState = np.zeros((numberOfNeuron, 1))
localfield_reservoirstate = reservoirState.copy()
tempMatrix = reservoirState.copy()
input_reservoir_weight = np.random.normal(0,
                                          np.sqrt(0.002),
                                          size=(numberOfNeuron, 3))
reservoir_2_reservoir_weight = np.random.normal(0,
                                                np.sqrt(2 / 500),
                                                size=(numberOfNeuron,
                                                      numberOfNeuron))

allstates = np.zeros((noOfInput, numberOfNeuron))
```

```
allstates = np.zeros(
    (noOfInput, 500))  # Initialize allstates as (noOfInput, 500, 1) array

for i in range(noOfInput):

  x = input[i][:]
  allstates[i] = reservoirState.reshape(500, )
  # allstates.append(reservoirState)
  if i % 500 == 0:
    print(i, x)
  for j in range(numberOfNeuron):

    localfield_reservoirstate[j] = np.dot(
        reservoir_2_reservoir_weight[j], reservoirState) + np.dot(
            input_reservoir_weight[j], x.transpose())
    tempMatrix[j] = np.tanh(localfield_reservoirstate[j])
  reservoirState = tempMatrix.copy()

ridgeparameter = np.identity((500))
np.fill_diagonal(ridgeparameter, 0.01)
# ridgeparameter.fill(0.01)

allstates = allstates.transpose()
firstTerm = np.dot(input.transpose(), allstates.transpose())
secondTerm = np.linalg.inv(
    np.dot(allstates, allstates.transpose()) + ridgeparameter)
outputWeights = np.dot(firstTerm, secondTerm)

#---test data-----------------------------------------------------------

reservoirState = np.zeros((numberOfNeuron, 1))
testData = []
with open('test-set-7.csv', 'r') as f:
  csv_reader = csv.reader(f)
  for row in csv_reader:
    testData.append((row))
testData = np.array([[float(item) for item in sublist]
                     for sublist in testData])
testData = testData.transpose()

print('r2r-weight', reservoir_2_reservoir_weight.shape)
print('reservoir-state', reservoirState.shape)
print('input-reservoir-weight', input_reservoir_weight.shape)
print('testdata', testData.shape)
for i in range(len(testData)):  #it was len(testdata)-1
  x = testData[i][:].reshape(1, 3)
```

```
  reservoirState = np.tanh(
      (np.dot(reservoir_2_reservoir_weight, reservoirState)) +
      np.dot(input_reservoir_weight, x.transpose()))

output = np.dot(outputWeights, reservoirState)

print(x.shape)
print('reservoirstate', reservoirState.shape)
print('outputWeights', outputWeights.shape)
print('outputshape', output.shape)

prediction = np.zeros((500, 3))

for time in range(500):
  reservoirState = np.tanh(
      (np.dot(reservoir_2_reservoir_weight, reservoirState)) +
      np.dot(input_reservoir_weight, output))
  # print('final reservior state', reservoirState.shape)
  output = np.dot(outputWeights, reservoirState)
  prediction[time] = output.transpose()  #.reshape(3, )
print(prediction[0])
xcomponet = []
zcomponet = []
ycomponet = []
for i in range(len(prediction)):
  xcomponet.append(prediction[i][0])
  ycomponet.append(prediction[i][1])
  zcomponet.append(prediction[i][2])

with open('prediction.csv', 'w', newline='') as file:
  writer = csv.writer(file)
  writer.writerow(ycomponet)

with open('xyzPrediction.csv', 'w', newline='') as file:
  writer = csv.writer(file)
  writer.writerow(prediction)

iterations = range(0, 500)
plt.plot(iterations, ycomponet, marker='o', linestyle='-')
plt.show()

fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')
ax.plot(xcomponet, ycomponet, zcomponet, label='3d-curve')
plt.show()
```