FFR 135 HW 3

Chaotic time-series prediction 2023

In [30]:
```
import numpy as np
import matplotlib.pyplot as plt
```

In [31]:
```
# parameters of the code
k = 0.01
inputNeurons = 3
reservoirNeurons = 500
tMax = 500
```

In [32]:
```
# load the training and test set
trainingSet = np.genfromtxt("training_set.csv", delimiter=",")
testSet = np.genfromtxt("test_set.csv", delimiter=",")

# print the shapes of the training and test set
print("trainingSet.shape: ", trainingSet.shape)
print("testSet.shape: ", testSet.shape)

# init the weights
inputWeights = np.random.normal(loc=0.0, scale= np.sqrt(0.002), size=(reservoirNeurons, inputNeurons
reservoirWeights = np.random.normal(loc=0.0, scale= np.sqrt(2/reservoirNeurons), size=(reservoirNeu

# print the shapes
print("inputWeights.shape: ", inputWeights.shape)
print("reservoirWeights.shape: ", reservoirWeights.shape)

# init the reservoir
X = np.zeros((trainingSet.shape[1], reservoirNeurons))
print("X.shape: ", X.shape)

# loop over all training examples
for i in range(trainingSet.shape[1]):
    ri_t1 = np.zeros((reservoirNeurons, 1))
    tmp1 = np.matmul(reservoirWeights, ri_t1) # shape=(500,1)
    tmp2 = np.matmul(inputWeights, trainingSet[:,i]) # shape=(500,)
    # shapes
    #print("tmp1.shape: ", tmp1.shape)
    #print("tmp2.shape: ", tmp2.shape)
    tmp2 = np.reshape(tmp2, (reservoirNeurons, 1)) # shape=(500,1)
    ri_t1 = np.tanh(tmp1 + tmp2) # shape=(500,1)
    X[i,:] = ri_t1[:,0] # shape=(500,)
```

```
trainingSet.shape:  (3, 19900)
testSet.shape:  (3, 100)
inputWeights.shape:  (500, 3)
reservoirWeights.shape:  (500, 500)
X.shape:  (19900, 500)
```

In [33]:
```
# update the reservoir weights
X = X[0:-1,:] # shape=(19899, 500)
Y = trainingSet[:,1:]

ridgeMatrix = k * np.eye(reservoirNeurons) # shape=(500,500)
XTX = np.matmul(X.T, X) # shape=(500,500)
XTX_inv = np.linalg.inv(XTX + ridgeMatrix) # shape=(500,500)
XTY = np.matmul(X.T, Y.T) # shape=(500,3)
outputWeights = np.matmul(XTX_inv, XTY) # shape=(500,3)
outputWeights = outputWeights.T # shape=(3,500)
```

In [34]:
```
# Output testset
newSize = testSet.shape[1] + tMax
testX = np.zeros((newSize, reservoirNeurons)) # shape=(600,500)
outputX = np.zeros((inputNeurons, newSize)) # shape=(3,600)
print("testX.shape: ", testX.shape)
print("outputX.shape: ", outputX.shape)

# loop over all test examples
for i in range(newSize):
    ri_t1 = np.zeros((reservoirNeurons, 1))
    if i < testSet.shape[1]:
        tmp1 = np.matmul(reservoirWeights, ri_t1) # shape=(500,1)
        tmp2 = np.matmul(inputWeights, testSet[:,i]) # shape=(500,)
        tmp2 = np.reshape(tmp2, (reservoirNeurons, 1)) # shape=(500,1)
        ri_t1 = np.tanh(tmp1 + tmp2) # shape=(500,1)
        testX[i,:] = ri_t1[:,0] # shape=(500,)

        output = np.matmul(outputWeights, ri_t1) # shape=(3,1)
        outputX[:,i] = output[:,0] # shape=(3,)
    else:
        tmp1 = np.matmul(reservoirWeights, ri_t1) # shape=(500,1)
        tmp2 = np.matmul(inputWeights, output) # shape=(500,)
        tmp2 = np.reshape(tmp2, (reservoirNeurons, 1)) # shape=(500,1)
        ri_t1 = np.tanh(tmp1 + tmp2)
        testX[i,:] = ri_t1[:,0]

        output = np.matmul(outputWeights, ri_t1) # shape=(3,1)
        outputX[:,i] = output[:,0]

predictedOutput = outputX[:,100:]
timePred = predictedOutput[1,:]
prediction = np.reshape(timePred, (1, predictedOutput.shape[1]))
```

```
testX.shape:  (600, 500)
outputX.shape:  (3, 600)
```

In [36]:
```
np.savetxt("prediction_1.csv", prediction, delimiter=",")
```