Code for Chaotic time series prediction

```
% Chaotic time series prediction, written by Axel Qvarnström

clear all
close all
clc

N = 500;  % numbers of reservoirs
trainingData = load('training-set.csv');
testData = load('test-set-7.csv');
Ttrain = length(trainingData);
Ttest = length(testData);

% Initializing weights
inputWeights = normrnd(0, 0.002, [500, 3]);
reservoirWeights = normrnd(0, 2/500, [500, 500]);

reservois = zeros(N,1); % initialize reservoir time step zero
reservoirMatrix = zeros(N,Ttrain);

% Updating the dynamics of the reservoir
for t = 1:Ttrain
    reservoirMatrix(:,t) = reservois;
    reservoisUpdated = UpdateReservoir(reservoirWeights, reservois, inputWeights, trainingData(:
    reservois = reservoisUpdated;
end



identityMatrix = eye(N,N);
k = 0.01;

% Training the output weights with ridge regression
outputWeights = RidgeRegression(reservoirMatrix, k, trainingData, identityMatrix);

%% Feed the test data and make predictions

% Looping through the test data for timesteps = 100
for t = 1:Ttest
    reservois = UpdateReservoir(reservoirWeights, reservois, inputWeights, testData(:,t));
    outputNeurons = Output(outputWeights, reservois);
end

% Making the prediction for 500 timesteps
outputNeurons2Matrix = zeros(3,500);
for T = 1:500
    reservois = UpdateReservoir(reservoirWeights, reservois, inputWeights, outputNeurons);
    outputNeurons2Matrix(:,T) = Output(outputWeights,reservois);
end
```

```
StoredOutputNeurons = outputNeurons2Matrix(2,:);
csvwrite('prediction.csv',StoredOutputNeurons)

    function output = Output(outputWeigths, reservoirs)

    output = outputWeigths * reservoirs;

end

    function outputWeights = RidgeRegression(R, k, xTrain, identityMatrix)

    outputWeights = xTrain * R' *(R * R' + k*identityMatrix);

end

    function newReservoirs = UpdateReservoir(weights, reservoirs, inputWeights, inputNeurons)

    newReservoirs = tanh(weights * reservoirs + inputWeights * inputNeurons);

end
```