

```

trainingSet = readmatrix('training-set.csv');
testSet = readmatrix('test-set-5.csv');

meanValue = 0;
inputVariance = 0.002;
reservoirVariance = 0.004;

inputNeurons = zeros(3,1);
inputWeights = sqrt(inputVariance) .* randn(3,500) + meanValue;

reservoirNeurons = zeros(500,1);
reservoirWeights = sqrt(reservoirVariance) .* randn(500,500) + meanValue;

outputs = zeros(3,1);

X = zeros(19899,500);

for i = 1:19899
    inputNeurons = trainingSet(:,i);
    reservoirNeurons =
reservoirUpdateRule(inputNeurons,inputWeights,reservoirNeurons,reservoirWeights);
    X(i,:) = transpose(reservoirNeurons);
end

k = 0.01;
diagonalK = k * ones(1,500);
outputWeights = trainingSet(:,2:19900) * X * inv(transpose(X) * X +
diag(diagonalK));
outputWeights = transpose(outputWeights);

X = zeros(600,500);
reservoirNeurons = zeros(500,1);
timeSeriesPrediction = zeros(3,500);

for i = 1:100
    X(i,:) = reservoirNeurons;
    inputNeurons = testSet(:,i);
    reservoirNeurons =
reservoirUpdateRule(inputNeurons,inputWeights,reservoirNeurons,reservoirWeights);
end

for i = 1:500
    X(i,:) = reservoirNeurons;
    outputs = transpose(outputWeights) * transpose(X(i,:));
    timeSeriesPrediction(:,i) = outputs;
    reservoirNeurons =
reservoirUpdateRule(outputs,inputWeights,reservoirNeurons,reservoirWeights);
end

```

```
writematrix(timeSeriesPrediction(2,1:500), 'prediction.csv');
```

```
function newReservoirNeurons =  
reservoirUpdateRule(inputNeurons,inputWeights,reservoirNeurons,reservoirWeights)  
  
newReservoirNeurons = zeros(500,1);  
  
for i = 1:500  
    newReservoirNeurons(i) = tanh(sum(inputNeurons .* inputWeights(:,i)) +  
sum(reservoirNeurons .* reservoirWeights(:,i)));  
end  
  
end
```