

# Chaotic time series prediction

Erik Norlin

```
xTrain = readmatrix("training-set.csv");
xTest = readmatrix("test-set-4.csv");

nInputs = height(xTrain);
nOutputs = nInputs;
nReservoirs = 500;

w_in = randn(nReservoirs,nInputs)*sqrt(0.002);
w_ij= randn(nReservoirs,nReservoirs)*sqrt(0.004);
b_ij = zeros(nReservoirs,1);
b_ik = zeros(nReservoirs,1);

nTrainingInputs = width(xTrain);
nTestInputs = width(xTest);

R = zeros(nReservoirs,nTrainingInputs);
I = eye(nReservoirs);
k = 0.01;

nTimeSteps = 500;
predictions = zeros(nOutputs,nTimeSteps);

% Training
for iTrainingInput = 1:nTrainingInputs-1

    % Local fields
    for jNeuron = 1:nReservoirs
        b_ij(jNeuron,1) = w_ij(jNeuron,:)*R(:,iTrainingInput);
        b_ik(jNeuron,1) = w_in(jNeuron,:)*xTrain(:,iTrainingInput);
    end

    % r(t+1)
    R(:,iTrainingInput+1) = tanh(b_ij + b_ik);
end

w_out = xTrain*R'/(R*R' + k*I);

% Testing
r = R(:,1); % Zeros
for iTestInput = 1:nTestInputs

    % Local fields
    for jNeuron = 1:nReservoirs
        b_ij(jNeuron,1) = w_ij(jNeuron,:)*r;
        b_ik(jNeuron,1) = w_in(jNeuron,:)*xTest(:,iTestInput);
    end

    % r(t+1)
    r = tanh(b_ij + b_ik);
end
```

```

for iTimeStep = 1:nTimeSteps

    % Output from r(t+1)
    for iNeuron = 1:nOutputs
        predictions(iNeuron,iTimeStep) = w_out(iNeuron,:)*r;
    end

    % Local fields
    for jNeuron = 1:nReservoirs
        b_ij(jNeuron,1) = w_ij(jNeuron,:)*r;
        b_ik(jNeuron,1) = w_in(jNeuron,:)*predictions(:,iTimeStep);
    end

    % r(t+1)
    r = tanh(b_ij + b_ik);

end

yComponent = predictions(2,:);
writematrix(yComponent,"prediction.csv")

```