

A comparison of Elman and Echo State Recurrent Neural Networks

Elliott Forney and Charles Anderson

Background and Motivation

Artificial Neural Networks (ANN) are trainable approximation structures that are composed of a number of simple, interconnected computational units, or neurons.

ANN are trained by adjusting the weights of the connections between neurons.

Recurrent Artificial Neural Networks are the subset of ANN that have feedback connections.

These feedback connections give RNN's an intrinsic state and memory, allowing them to approximate complex spatiotemporal patterns.

Many RNN architectures have been explored in the past several decades. However, few direct comparisons have been performed demonstrating their strengths and weaknesses.

Here, we seek to compare two RNN architectures when applied to three straightforward problems. We then examine their performance as memory requirements are increased.

Elman's Simple Recurrent Networks

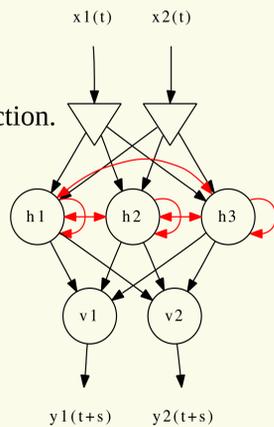
Elman's Simple Recurrent Networks (SRN) contain a single hidden layer with a hyperbolic tangent transfer function and a visible layer with a linear transfer function.

The hidden layer is fully connected, including recurrent connections with a delay of a single timestep.

SRN's are typically trained by unrolling the network a number of steps through time and following the error gradient to adjust the connection weights.

This process is known as Backpropagation Through Time (BPTT).

SRN's have been used to successfully solve a variety of problems since 1990.



Echo State Networks

Echo State Networks (ESN) contain a reservoir of neurons with hyperbolic tangent transfer functions.

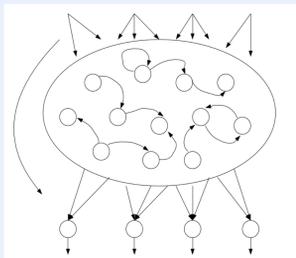
The reservoir is sparsely connected, typically less than 1%, and relatively large, containing hundreds to thousands of neurons.

The reservoir weights are randomly chosen and scaled so that the largest eigenvalue is less than one.

The reservoir typically possesses the Echo State Property, i.e. the same sequence of inputs will asymptotically result in the same reservoir state, regardless of initial conditions.

ESN's contain a linear output layer that is typically trained using linear regression techniques.

ESN's have been relatively popular since their discovery in 2001.



In these preliminary results, we will see that ESN's often outperform SRN's as memory requirements increase. The recent popularity of ESN's may be justified but further experiments are necessary.

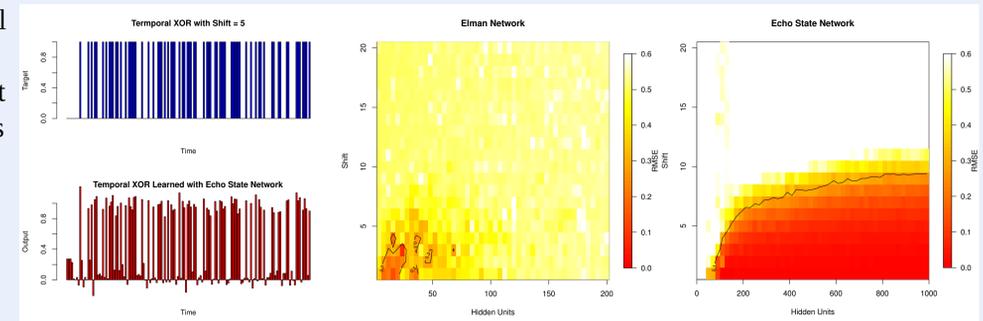
Temporal XOR

Temporal XOR operates over a random stream of bits. The desired output at time t is the XOR of the bits previously encountered at time $t-s$ and $t-s-1$ for a given shift s .

As the shift s is increased, the memory requirement for the RNN also increases.

Here, we explore how well SRN's and ESN's solve Temporal XOR as the shift and the number of neurons varies.

ESN's appear to solve the problem for much larger values of s than SRN's

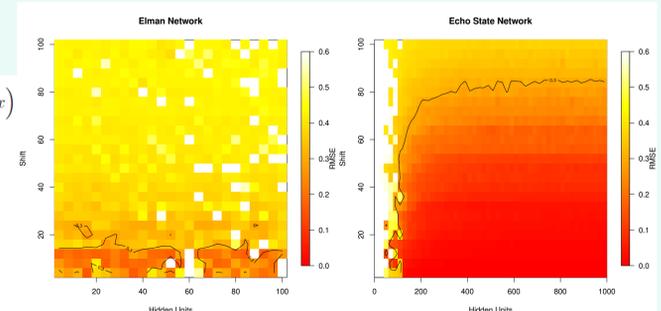
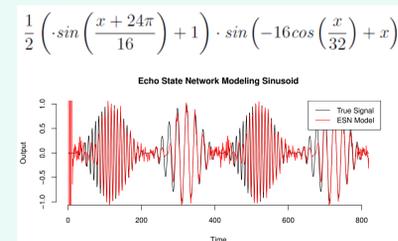


Sinusoid Forecasting

Here, we explore the ability of SRN's and ESN's to forecast a time series generated by sampling a sinusoid that has variations in both frequency and amplitude.

The RNN's memory requirement increases as the number of steps forecast ahead in time increases.

Again, ESN's appear to outperform SRN's for longer-term forecasts.



Chaotic Laser Intensity Forecasting

Next, we examine the ability of ESN's and SRN's to forecast a time series generated by measuring the intensities of a Far-Infrared-Laser in a chaotic state.

This dataset was borrowed from The Santa Fe Time Series Competition dataset and was produced by a well understood physical phenomenon.

Here, ESN's only slightly outperform SRN's as the time series is forecast further ahead in time.

However, overfitting is an issue in this experiment and a more thorough analysis is necessary

