Extending a Neural Implementation of The Kalman Filter to Arbitrary Dimensions

Lane Lewis, Dr. Wilson

The University of Arizona

Introduction

Bayesian statistics or, in short, the mathematics of how to optimally change beliefs based on previous information, has proven to be a highly useful tool across the applied sciences. In addition to this, a growing body of evidence from neurological and psychological experiments suggests that humans and other animals use approximate Bayesian inference to understand their environments. However, the mechanisms through which their nervous systems implement this inference remain an open area of investigation. In this research, a previously derived, biologically plausible attractor neural network implementation of a particular Bayesian inference model, known as a 1-dimensional Kalman filter, is extended to arbitrary dimensions.

Problem Background

The Kalman Filter

A Kalman Filter is an algorithm highly used in tracking systems such as radar. It gives an optimal way to predict the location of an object given some imprecise and incomplete measurements taken about that object in the past. The algorithm is given by the following equations:

EQ. 1
$$\mu_n^{(a^-)} = A(t_\Delta)\mu_n^{(a^-)}$$

EQ. 2
$$\Sigma_n^{(a^-)-1} = A(t_\Delta)^{-T}\Sigma_{(n-1)}^{(a^+)-1}A(t_\Delta)^{-1}$$

EQ. 3
$$\mu_n^{(a^+)} = (\Sigma_n^{(a^-)-1} + H^T R_n^{-1} H)^{-1}(\Sigma_n^{(a^-)-1}\mu_n^{(a^-)} + H^T R_n^{-1} y)$$

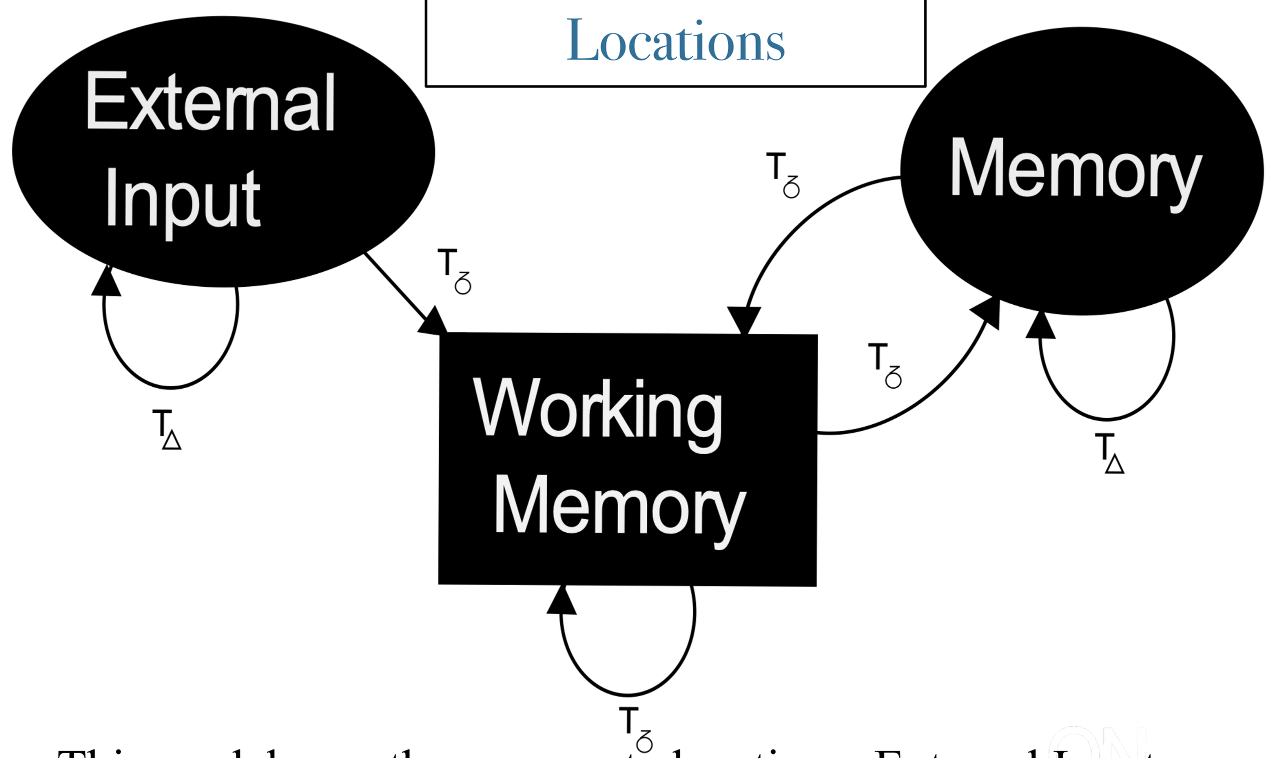
EQ. 4
$$\Sigma_n^{(a^+)-1} = \Sigma_n^{(a^-)-1} + H^T R_n^{-1} H$$

Attractor Neural Networks

Attractor neural networks are a class of mathematical constructs that describe how large quantities of neurons behave using dynamical system theory. They are commonly used to model how animals maintain their orientation in space and time. This research focuses on a particular case where the system's stable activation curve takes a nearly gaussian shape that can travel over time and each neuron has the following biologically plausible activation function:

$$U(x_{n+1}^{(i)}) = \frac{[U(x_n^{(i)})]_+}{S + c\sum_i U(x_n^{(i)})}$$

Time Scales & Locations

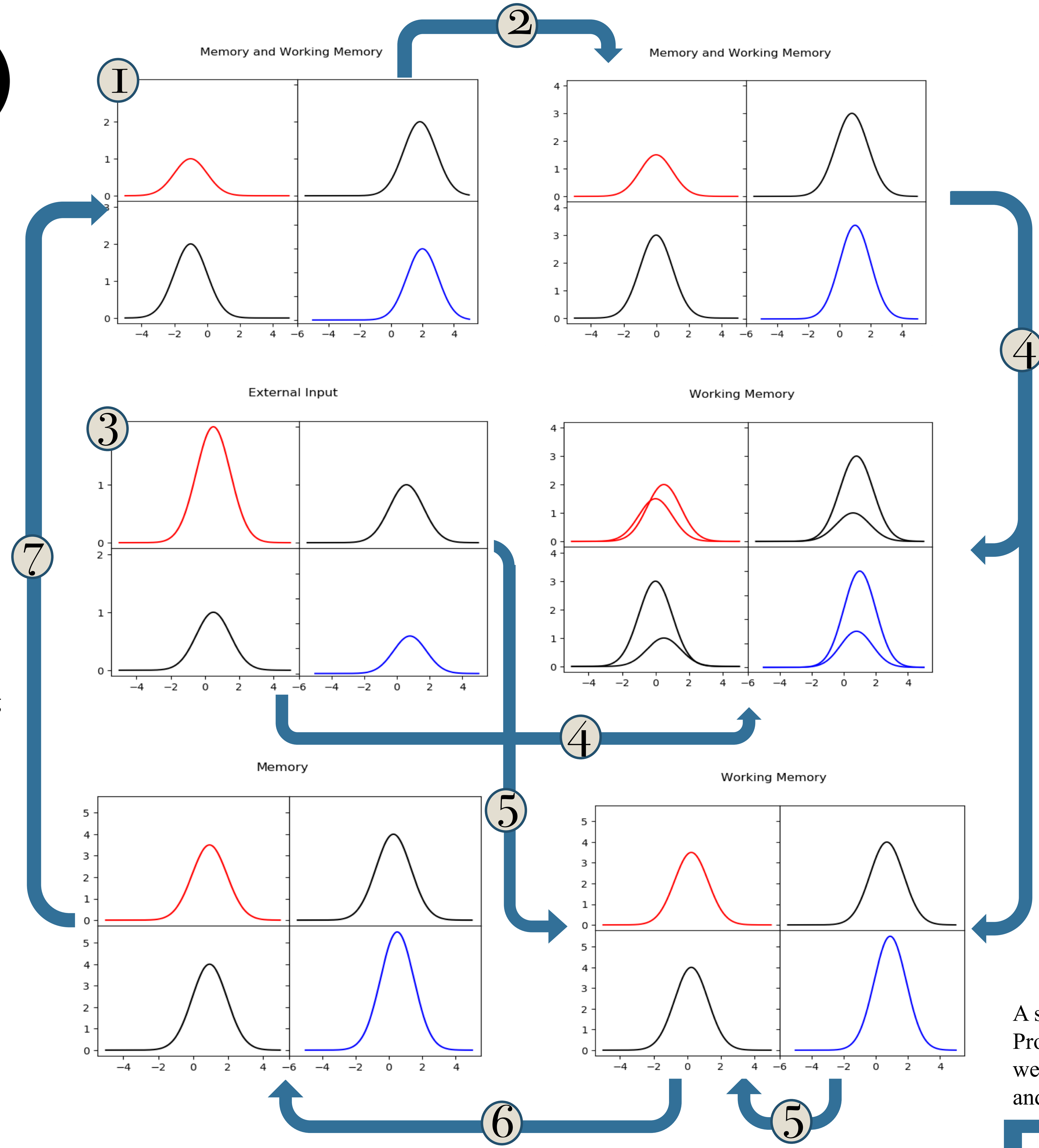


This model uses three separate locations: External Input, Working Memory, and Memory respectively. Each location is composed of a set of N^2 ANNs where N is the size of the Kalman Filter being approximated. An entry of an inverted covariance matrix is encoded in the height of each ANN bump and a 1-dimensional location is given by it's center. Two time scales are present at different locations in this discrete model.

Model Steps

1. The Memory and Working Memory models take the form of two identical sets of bumps on their respective networks.
2. A prediction update is applied to the Memory and Working Memory models. This shifts each bump to a new location. and scales it (performing EQ. 1 and EQ. 2 of the filter).
3. An External Input network set is introduced for some small amount of time.
4. Information supplied by the External Input gets added to the Working Memory model. Assuming that the Memory and External Input predict similar locations, the bump heights will approximately add, mimicking EQ. 4 of the Kalman Filter.
5. Using the information in the Memory, External Input and newly updated Working Memory, the bumps shift according to a gradient descent method of solving EQ. 3. This step is recursive, and gets carried out a number of times until the External Input terminates
6. On termination of the External Input, the Working Memory replaces the current model in Memory.
7. The algorithm is then applied recursively.

The New Model



Differences From the Previous Model

This model differed very significantly from the original, 1-dimensional case. Mainly these differences were:

- The addition of gradient descent to solve for the Kalman Filter's new estimation for the state location. While this is not an issue in the 1-dimensional case, this allowed the arbitrary dimensional case to carry out every operation of the Kalman Filter without the need of inverting any matrices beyond the constant matrix A.
- Introducing three separate network locations. In the original model, this was carried out in a single network. This extension allows for all the needed information to be supplied to the network for gradient descent.
- Adding a second, short time scale. This allowed for gradient descent to take place while approximately not affecting the general update equations.

Model Limitations

While making the original model much more general it suffers from several constraints including:

- The matrix H must be the pseudoinverse of its transpose.
- Covariance entries can't change sign.
- Several functions are "hardcoded" into the network.

Acknowledgments

A special thanks to UBRP and the Office of the Provost for funding and facilitating this research as well as Dr. Wilson and The NRD Lab for funding and mentoring this project.

THE UNIVERSITY OF ARIZONA
Undergraduate Biology Research Program