Using Echo State Networks to Solve Stochastic Optimal Control Problems

Allen G. Hart, Kevin R. Olding, Jon H. P. Dawes, Alex M. G. Cox, Olga Isupova
a.hart@bath.ac.uk, k.r.olding@bath.ac.uk, j.h.p.dawes@bath.ac.uk, a.m.g.cox@bath.ac.uk,
oi260@bath.ac.uk

Echo State Networks (ESNs)

Echo State Networks are special type of recurrent neural network that are suitable for solving problems involving a time series [1]. Suppose we have a time series $u_k \in \mathbb{R}^m$, then we create a sequence r_k of n dimensional ESN states like so:

$$r_{k+1} = \text{relu}(Ar_k + W^{\text{in}}u_k + b)$$

where A is the $n \times n$ adjacency matrix representing the connections between the internal neurons, W^{in} is the $n \times m$ input matrix connecting the input to the neural network, b is n-dimensional bias vector and relu is the rectified linear unit $\text{relu}(x) = \max(x, 0)$.

Training an ESN

Suppose we have a sequence of targets a_k alongside the observations u_k and our goal is to train the ESN to predict the targets a_k given the observations u_k. Then we seek the $m \times n$ output matrix W^{out} that satisfies

$$\min_{W^{\text{out}}} \sum_{k=1}^{K} \|W^{\text{out}}r_k - a_k\|^2 + \lambda\|W^{\text{out}}\|_2^2$$

which is just a regularised least squares problem!

Stochastic Optimal Control

In a discrete time stochastic optimal control problem, we have a sequence of observations u_k and for each of these seek an optimal action a_k, based on the past observations, such that the trajectory of observations minimises some cost, that depends on the observations u_k and actions a_k. If we have some examples of optimal actions, we can attempt to learn the optimal control using an Echo State Network.
Often in practice, we do not have examples of optimal actions - so instead we will try to learn the *value functional* V defined on a sequence of past observations z

$$V(z) = \mathcal{C}(z) + \gamma VT(z)$$

where V is the unique fixed point of the contraction mapping Φ defined

$$\Phi H = \mathcal{C}(z) + \gamma HT$$

where T is a shift operator, $\gamma \in (0, 1)$ is a discount factor and $\mathcal{C}$ is the cost functional.

References

[1] Hart, A. G., Hook, J. L., Dawes, J. H. P. (2020). Embedding and Approximation Theorems for Echo State Networks, *Neural Networks*, 128, Pages 234-247,

Acknowledgements

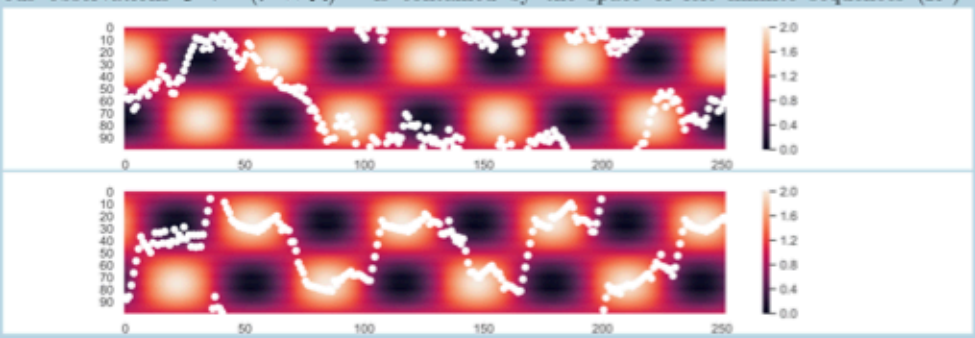
We acknowledge that Allen Hart and Kevin Olding are supported by a scholarship from the ESPRC Centre for Doctoral Training in Statistical Applied Mathematics at Bath (SAMBa), under the project EP/L015684/1.

Playing Bee World with an ESN

Bee World is set on the circle of circumference 1, we will call S^1. We will represent S^1 by a line with edges identified. At every point x on the line, there is a non-negative quantity of pollen, which may be enjoyed by the bee without depletion. Furthermore, the pollen at every point x varies with time t according to a function

$$p(x, t) = 1 + \cos(\omega t)\sin(2\pi x)$$

(which we chose somewhat arbitrarily) that is unknown to the bee. Thus, the amount of pollen enjoyed by the bee at time t is a value that lies in the space $[0, 2]$ we will denote $\mathcal{P}$. Time advances in discrete integer steps $t = 0, 1, 2, ...$, and at any time point t a bee at point x observes the quantity of pollen $p \in \mathcal{P}$ at point x and nothing else. Having made this observation, the bee may choose to move anywhere in the interval $(x - r, x + r)$ for some fixed $0 < r < 1$ and arrive at its chosen destination at time $t + 1$. The interval of possible moves $(-r, r)$ is called the action space and is denoted $\mathcal{A}$. The goal of the bee is to devise a policy whereby, given all its previous observations, the bee makes a decision as to where to move next, such that the discounted sum over all future pollen is as great as possible. The space of all previous observations $\mathcal{S} := (\mathcal{P} \times \mathcal{A})^{\mathbb{Z}^-}$ is contained by the space of left infinite sequences $(\mathbb{R}^2)^{\mathbb{Z}^-}$.



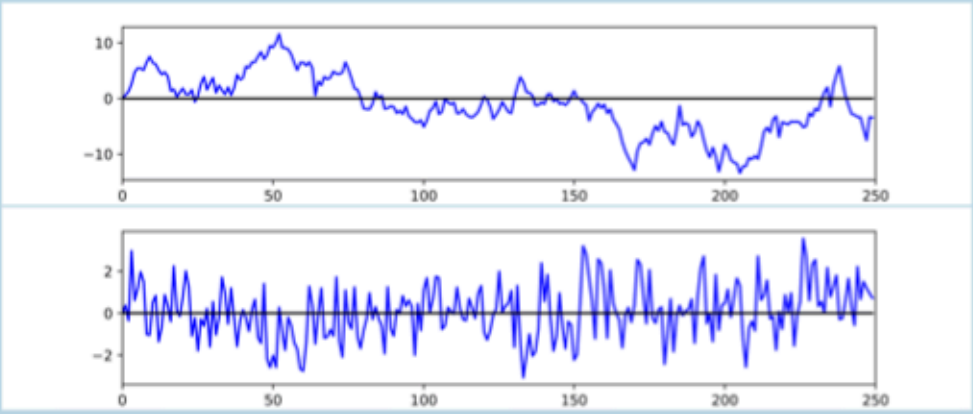
We chose an initial policy of i.i.d uniform moves, the results of which is shown in the first of these 2 plots. The white dots represent the position (x-axis) of the bee at time t (y-axis). Then, we trained an ESN to learn the value functional associated to this policy using regularised least squares. Then, we set the bee to make the move that maximises the value functional learned by the ESN. The result of this improved policy is also shown above.

Solving a Market Making Problem with an ESN

Another control problem we might solve with an ESN is a simplification of a market making problem arising in stochastic finance. We have a Brownian motion that represents an inventory - which we can control by applying a drift term. There is a cost that scales quadratically with the inventory's distance from 0, and quadratically with the magnitude of the applied control. We wish to minimise the total cost of operations over time, with future costs expenentially less important than near term costs. Under the optimal control, the inventory is a Ornstein-Uhlenbeck process

$$dx_t = -\theta x_t dt + \sigma dW$$

where x_t gives the value of the process, $\theta > 0$, $\sigma > 0$, and W_t is a standard Brownian motion.



We trained an ESN to learn the value functional for an initial policy (shown in the first figure) that is allowed to stray quite far from 0. Then, we improved upon this policy my making the move that maximises the value functional learned by the ESN. The inventory under the new policy is shown in the second figure. We can easily compare the solution learned by the ESN (in discrete time) to the analytic solution (in continuous time) - but we did not the space to fit that in here.