BAYESIAN WEIGHT UPDATES STABILIZE AND IMPROVE LOCAL LEARNING IN A RECURRENT NEURAL NETWORK

Jorge A. Menendez [1,2,3,*], *Peter E. Latham* [1]

[1] Gatsby Computational Neuroscience Unit, University College London, London, UK
[2] Sainsbury Wellcome Centre, University College London, London, UK
[3] CoMPLEX, University College London, London, UK
[*] jorge.menendez.15@ucl.ac.uk

Neural networks in the brain must be capable of not only producing time-varying signals but also learning how to produce them. A typical approach to this problem is to build a chaotic recurrent neural network (RNN) and learn a set of readout weights w_j so that a linear readout of the neurons' firing rates $\sum_j w_j r_j(t)$ produces the desired signal $f(t)$ (Sussillo & Abbott, 2009). By feeding back this readout to all the neurons, the RNN will stabilize - so long as the readout stays close to $f(t)$. This can be easily achieved using powerful but non-local learning rules such as recursive-least-squares (RLS).

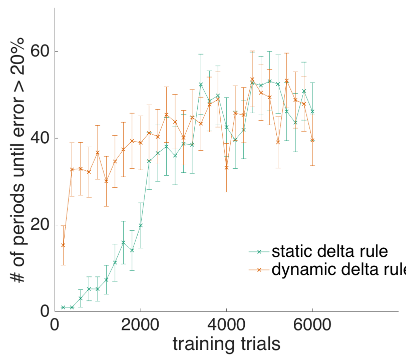

Fig 1: number of periods of $f(t)$ until readout error (mean squared error normalized by variance of $f(t)$) exceeds 20%, using readout weights at different points throughout training. Error bars designate S.E.M.

Local learning rules, however, struggle to solve this problem due to the instabilities inherent to the chaotic dynamics of the RNN. Even a slight deviation from the optimal set of weights can prevent the network from stabilizing, thus producing a readout that diverges from its target within even one or two periods of $f(t)$. Indeed, a standard delta-rule will produce readouts that can maintain low error for only 10 periods or less after 1000 trials of training (fig 1). One reason for this is that the learning rate must be very low for the delta-rule to maintain stability throughout training, meaning that the readout weights will take a long time to align with their optimum. This contrasts with RLS, which typically makes large updates early in training and then slows down. This suggests the learning rate should adapt over time, so it can start big and then get small.

But how quickly should the learning rate decrease? A principled approach to this question arises from considering the fact that any weight update should take into account the uncertainty in the true value of the weight (Aitchison, Pouget & Latham, 2017). Framing the learning problem as Bayesian inference, under certain assumptions we derive the following 'dynamic' delta-rule with a history-dependent decaying learning rate:

$$\Delta w_i(t) = \frac{1}{\beta + \sum_{t'=1}^{t} r_i(t')^2} \left(f(t) - \sum_j w_j r_j(t) \right) r_i(t)$$

We find that these learning rate dynamics help attenuate the instabilities described above, by allowing the network to start with a large learning rate that quickly brings the readout weights closer to the optimal ones. This results in a stable solution within 1000 trials (fig 1), meaning that fixing the weights and running the network at this point will result in a readout with <20% error for over 30 periods of $f(t)$.