Mechanistic neural circuit models of hierarchical inference

Summary: We explore how brains might solve hierarchical inference tasks by training continuous-time recurrent neural networks (RNNs) to do so and analyzing their representations, connectivity, and population dynamics. Such models allow comparisons with animal behavior and neural data, and their state space dynamics reveal the fundamental latent variables and computations relevant for solving the task [6, 8, 9, 4, 7]. We adopt a perceptual decision-making and change-point detection task [3]. Each trial involves estimating whether a visual stimulus is on the left or right. Across a variable number of consecutive trials (a block), the stimulus appears with higher probability on one side; for the next block, the probabilities switch. The task is hierarchical because mice must infer two nested latent variables given the observations: on which side the stimulus is (stimulus side), and on which side the stimulus is more likely to appear (block side). These variables are latent because they are uncued and must be inferred. They are nested because a stimulus side estimate is necessary to update the block side estimate. To solve the task, RNNs must infer the nested latent variables over disparate timescales separated by 1-2 orders of magnitude. We make four discoveries: First, RNNs learn behavior that is quantitatively similar to ideal Bayesian baselines. Second, RNNs perform inference by learning a two-dimensional subspace defining beliefs about the latent variables. Third, the geometry of RNN dynamics reflects an induced coupling between the two separate inference processes necessary to solve the task. Fourth, we reduce the large RNN to a very small highly interpretable circuit model, using a new model compression technique we call Representation and Dynamics Distillation (RADD). We conclude with predictions to guide exploration and analysis of mouse behavior, neural activity and circuitry.

Additional Detail: We train RNNs to perform the task via gradient descent on cross entropy summed over all steps across multiple blocks. The target distribution is the stimulus side in each trial. The RNN dynamics for the n-th step in the t-th trial are defined below where $a_{n,t}$ is the action taken by the RNN:

$$h_{n,t} = (1-\alpha)h_{n,t-1} + \alpha \tanh(W^{rec}h_{n,t-1} + W^{obs}o_{n,t} + b^{rec})$$
$$a_{n,t} = \text{softmax}(W^{out}h_{n,t} + b^{out})$$

Behavior: RNNs nearly match a normative Bayesian model. Both the RNN and the Bayesian actor display similar accuracy as a function of stimulus contrast (Fig. 1A): the fraction of correct actions is highest for high stimulus contrast and lowest for zero contrast. On concordant trials (i.e. when the stimulus and block sides agree), the RNN and the Bayesian actor both perform much better than on discordant trials (i.e. when the stimulus and block sides disagree), direct evidence of the influence of estimated block side on trial-by-trial decision making. The concordant-discordant discrepancy shrinks with increasing stimulus contrast, meaning that when the contrast is high, the stimulus (likelihood) dominates over the block prior, while at low contrasts, the block prior dominates. As further confirmation that the RNN tracks the block side near optimally, we show that the RNN's fraction of correct answers rapidly climbs following a block change-point (Fig. 1B); this steep climb occurs within the first ten trials of a block and closely matches the Bayesian actor.

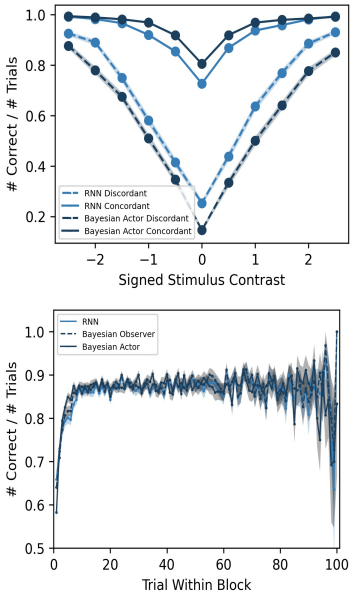

Figure 1:

Representations: The first two principal components (PCs) of RNN activity explain 88.74% of the variance in activity, suggesting the RNN learned a low-dimensional solution. The RNN readout vector, which converts hidden states into actions, explicitly gives us the direction along which the RNN encodes its stimulus side belief. A logistic classifier trained to predict block side at each RNN step reaches 82.6% accuracy on a 67-33% train-test split; a separate classifier for block side trained from the 2-dimensional PCA plane of RNN activity had 82.5% accuracy, affirming that the RNN's PCA plane encompasses the two latent variables being inferred. The block and stimulus readouts are non-orthogonal (72° in the high-dimensional RNN space, 66° in the PCA plane); this deviation from orthogonality is critical to how the network performs hierarchical inference. We found that the RNN infers the stimulus side and block side by integrating observations at different rates. Specifically, the RNN state's velocities along the two directions had positive slopes (0.15 for stimulus, 0.04 for block) as a function of the difference in the right and left observation values, with $p < 1e-5$, confirming that rightward (leftward) stimuli

simultaneously move the state both toward the right (left) side of the stimulus decision boundary and toward the right (left) side of the block decision boundary. The respective magnitude of these two slopes (stimulus slope ≈ 4 * block slope) means stimulus side inference occurs more quickly than block side inference.

Dynamics: Visualizing flows of the RNN in the PCA plane (Fig. 2) reveals the behavior of the system. In the presence of a stimulus, the network states flow toward one of two discrete attractors, in the left-block left-stimulus quadrant (top left) or in the right-block right-stimulus quadrant (top right). When the stimulus is absent and there is no performance feedback, the network exhibits a 1-dimensional line attractor (top middle). The line attractor is mainly aligned with the block readout vector, which allows the RNN to preserve its block side belief across trials. Even though block side is a discrete variable (left or right), the network's estimate of block side is, and should be, a continuous quantity. The line attractor has a small projection along the stimulus readout vector, which translates the block belief into a stimulus prior, in the form of an initial condition for the state that biases the RNN to select the concordant stimulus side on the next trial. Surprisingly, performance feedback has little effect on network state (Fig. 2).

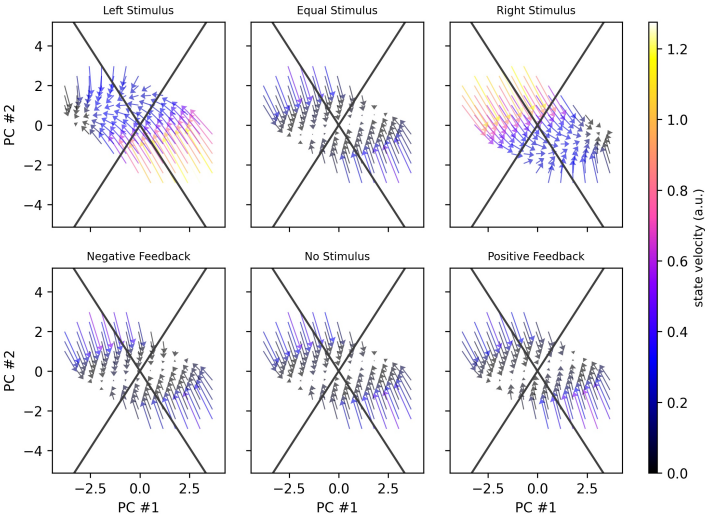

Figure 2:

Mechanism: To extract a low-dimensional, interpretable circuit used by the RNN, we propose the method RADD, a variation of knowledge distillation [2, 1, 5] in which we train a small RNN with hidden states $\hat{z}_t$ to reproduce a low-dimensional projection of the hidden states of the task-trained RNN. The RADD RNN is highly interpretable: activation of its two units correspond to stimulus and block side beliefs, respectively.

$$\hat{z}_{n,t} = \begin{bmatrix} \text{Stim Belief}_{n,t} \\ \text{Block Belief}_{n,t} \end{bmatrix} = \tanh\left(\begin{bmatrix} 0.59 & 0.28 \\ 0.18 & 0.90 \end{bmatrix} \hat{z}_{n,t-1} + \begin{bmatrix} -0.19 & 0.19 & 0.005 \\ -0.04 & 0.04 & 0.019 \end{bmatrix} \begin{bmatrix} o^L_{n,t} \\ o^R_{n,t} \\ r_{n,t} \end{bmatrix} + \begin{bmatrix} 0.00 \\ 0.00 \end{bmatrix} \right)$$

Observations drive the stimulus and block side beliefs in a common direction, but movement is 5 times greater along the stimulus direction than the block direction. The self-excitation of each unit extends its stimulus integration time (for an effective time-constant of $\tau/(1 - w_{self})$); the two effective time-constants reflect the observed differences in integration timescale for stimulus and block beliefs (Fig. 3A). The excitation between units shows that stimulus and block beliefs reinforce one another. The feedback input receives negligible weighting, consistent with our earlier observation that the network disregards feedback. The RADD RNN preserves the phase portrait of the task-trained RNN (Fig. 3B).

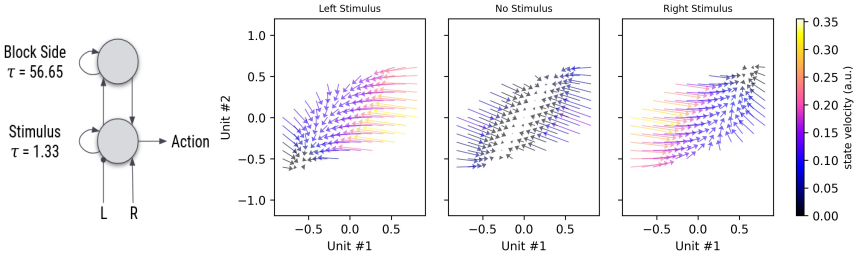

Figure 3:

Predictions for Neural Data: This work makes the following predictions for our collaborators who are currently collecting mouse behavioral and electrophysiological data. First, mice should learn a 2-dimensional encoding of the stimulus and block sides. Second, sensory evidence should move the state with different amplitudes - related by the ratio of the integration timescales for the two variables - along stimulus and block directions. Third, two kinds of dynamical behavior should be observed: (1) flows to one of two discrete fixed points, driven by sensory evidence, and (2) relaxation to a line attractor, maintaining the internal estimate of block side. Fourth, the block line attractor should display a small projection onto the stimulus side direction; this enables updating the block posterior after each trial as well as generating a stimulus side prior for the next trial.