Learning Dynamics and Geometry in Recurrent Neural Controllers

Ann Huang (annhuang@g.harvard.edu)
Satpreet H. Singh (satpreet_singh@hms.harvard.edu)
Kanaka Rajan (kanaka_rajan@hms.harvard.edu)
Harvard Medical School, Cambridge, MA, 02139, USA

Abstract

Understanding how recurrent neural networks (RNNs) learn to perform complex tasks through interaction with an environment, i.e. as *agents* or *controllers*, is important for both artificial intelligence and neuroscience. A lot of previous work has analyzed RNNs trained using supervised learning, and relatively less attention has been paid to reinforcement learning (RL) in the context of recurrent architectures and to their learning dynamics. Here, we take a step towards addressing this gap by thoroughly analyzing the learning dynamics of RNN-based artificial agents trained by reinforcement to solve a classic nonlinear continuous control problem – the Inverted Pendulum. Our framework provides key intuitions on the evolution of the control policy, neural dynamics, representational geometry, and memory in RNN-based agents.

Keywords: recurrent neural network; deep reinforcement learning; learning dynamics; dynamical systems;

Introduction

Recurrent Neural Networks (RNNs) are versatile and widely used models of neural activity and behavior in Neuroscience (Rajan, Harvey, & Tank, 2016; Barak, 2017). Most current research has focused on analyzing the internal dynamics in fully trained networks (Rajan & Abbott, 2006; Vyas, Golub, Sussillo, & Shenoy, 2020). However, how such dynamics emerge through the process of learning and how changes in learning dynamics affect task performance are relatively less well explored. In recent work (Marschall & Savin, 2023; Hocker, Constantinople, & Savin, 2024; Driscoll, Shenoy, & Sussillo, 2022), early progress has been made towards understanding how the attractor landscape of RNNs changes during learning in the *supervised learning* setting. Here we consider the less studied setting of *reinforcement learning* (RL) to train RNN *agents* or *controllers*, where the interaction between an artificial agent and an external environment shapes its internal dynamics and behavioral policy throughout learning. We provide a thorough analysis of the inverted pendulum problem as a first step, developing a flexible analysis framework that can be applied to probe the learning dynamics and resulting behaviors of neural network-based agents in more complex or ethologically relevant RL tasks.

Methods

We trained RNN agents to balance an Inverted Pendulum in the upright position by applying appropriate torque (Fig. 1a) in the OpenAI Gym Pendulum environment (Brockman et al., 2016) using policy gradients (Ni, Eysenbach, & Salakhutdinov, 2021). Agents consisted of vanilla RNNs followed by 1-layer feedforward policy and value networks (all 64 units wide, *tanh* nonlinearity). We considered both partially observable (PO) agents which only received the angular position θ as inputs, and fully observable (FO) agents, which additionally received the angular velocity $\dot{\theta}$. Training was performed for 100 gradient updates, with 1024 simulation steps per update.

The update equation for the RNN is $\mathbf{h}_t = F(\mathbf{h}_{t-1}, \mathbf{x}_t) = \tanh(\mathbf{W}_h \mathbf{h}_{t-1} + \mathbf{W}_x \mathbf{x}_t + \mathbf{b})$, where $\mathbf{h}_t$ is its hidden state, $\mathbf{W}_h$ the recurrent weight matrix, $\mathbf{x}_t$ the input vector, $\mathbf{W}_x$ the input-to-hidden weight matrix, and $\mathbf{b}$ a bias term (Sussillo & Barak, 2013). Linearizing this around a an expansion point $(\mathbf{h}^{\mathrm{e}}, \mathbf{x}^{\mathrm{e}})$, we obtain a linear dynamical system approximation: $\mathbf{h}_t \approx F(\mathbf{h}^{\mathrm{e}}, \mathbf{x}^{\mathrm{e}}) + \mathbf{J}^{\mathrm{rec}}\big|_{(\mathbf{h}^{\mathrm{e}}, \mathbf{x}^{\mathrm{e}})} \Delta \mathbf{h}_{t-1} + \mathbf{J}^{\mathrm{inp}}\big|_{(\mathbf{h}^{\mathrm{e}}, \mathbf{x}^{\mathrm{e}})} \Delta \mathbf{x}_t$, where $\Delta \mathbf{h}_{t-1} = \mathbf{h}_{t-1} - \mathbf{h}^{\mathrm{e}}$ represents the linearized system's state, $\Delta \mathbf{x}_t = \mathbf{x}_t - \mathbf{x}^{\mathrm{e}}$ denotes the input, $\mathbf{J}^{\mathrm{rec}}$ is the recurrence Jacobian matrix, and $\mathbf{J}^{\mathrm{inp}}$ is the input Jacobian matrix. $\mathbf{J}^{\mathrm{rec}}\big|_{(0,0)} = \mathbf{W}_{\mathbf{h}}$ and $\mathbf{J}^{\mathrm{inp}}\big|_{(0,0)} = \mathbf{W}_{\mathbf{x}}$. Previous studies have investigated the eigenvalues and eigenvectors of the recurrence matrix and recurrence Jacobian to understand the impact of connectivity on network dynamics (Rajan & Abbott, 2006; Singh, van Breugel, Rao, & Brunton, 2023). In particular, (Maheswaranathan, Williams, Golub, Ganguli, & Sussillo, 2019) derive the stimulus integration timescale τ_i for a stable eigenvalue λ_i (i.e., $|\lambda_i| \leq 1$) by considering the discrete-time iteration $h_i(t) = \lambda_i^t h_i(0)$, which governs stimulus integration along the direction of the eigenvector v_i corresponding to λ_i and then compare this with the equivalent continuous-time equation $h_i(t) = h_i(0)e^{-t/\tau_i}$ to get $\tau_i = |(1/\ln|\lambda_i|)|$. We use 1000 timesteps (5 episodes) to generate estimates of τ_i.

Results

Training gradually sharpens the policy decision boundaries between positive and negative torques, with sharper boundaries observed for the FO environment (Fig. 1b). In the PO case, training prunes the recurrent dynamics into a ring to efficiently represent the circular state variable θ (Fig. 1c). During training, a stable fixed point (FP), associated with the upright pendulum, emerges earlier, followed by the appearance of an unstable FP for the free-hanging pendulum state. The stable FP's proximity to the goal location is significantly correlated with the reward obtained by the controller (Fig. 1d). Stimulus integration exhibits distinct regimes in its evolution (Fig. 1e).

Future work

We will investigate more complex and biologically inspired environments, with longer evidence integration memory requirements, and complex sequential decision making and planning.

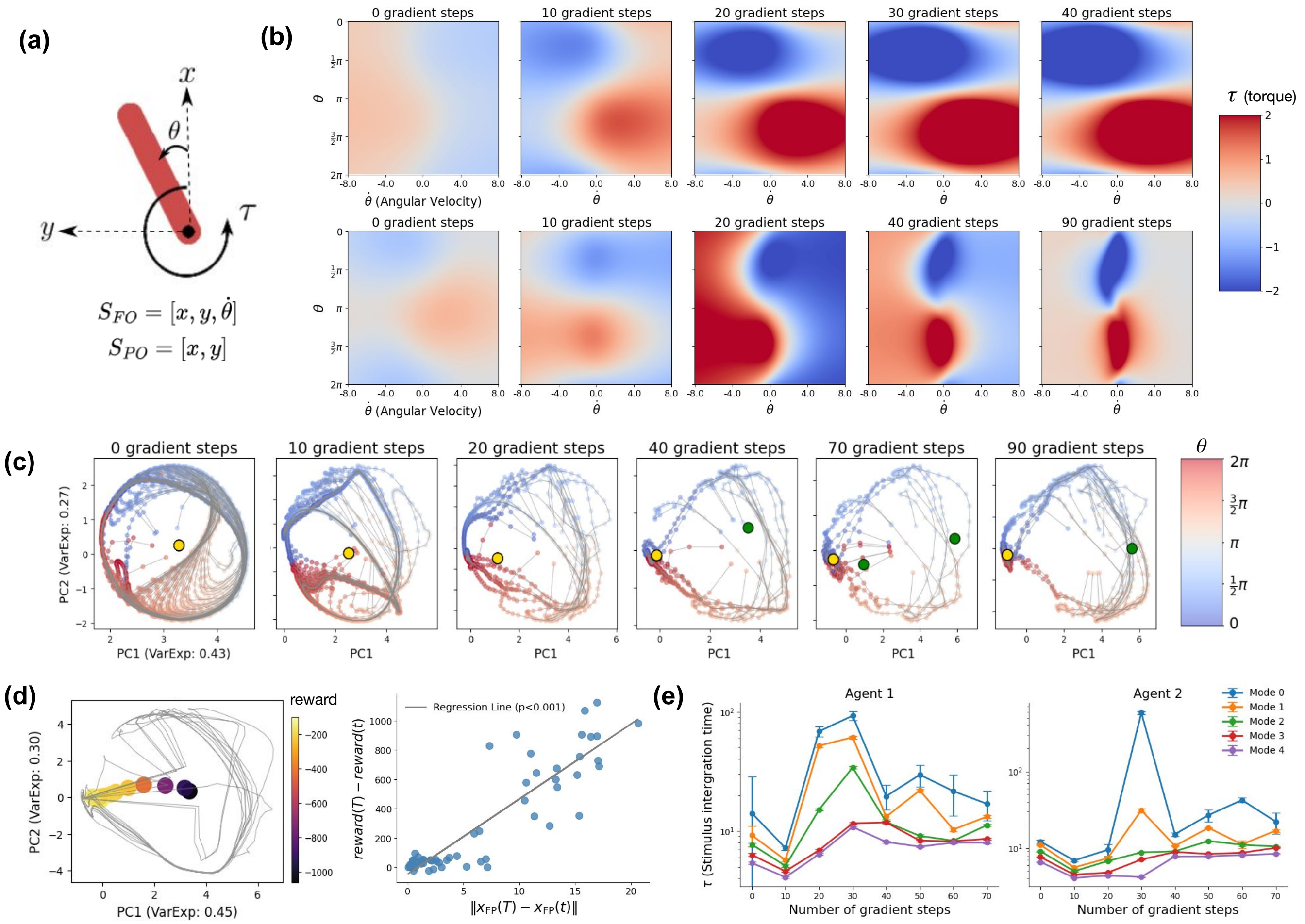

Figure 1: **Evolution of the control policy, recurrent dynamics, geometry, and memory of recurrent neural controllers in the Pendulum task. (a) Schematic representation of the Pendulum**, with different state spaces in fully observable (FO) and partially observable (PO) environments. **(b) Policy evolution of the recurrent neural controller**, visualized as a function of θ and $\dot{\theta}$. The top panel depicts the PO environment; the bottom, the FO environment. For both PO and FO environments, training sharpens the policy landscape, distinguishing between positive and negative torques more clearly. In the FO environment, the controller seems to use the extra $\dot{\theta}$ information to learn a more precise control policy, as indicated by the near-zero torques at high $\dot{\theta}$ values. **(c) Recurrent dynamics in the PO environment**, shown in the top two principal components, colored by θ at different training points. Training prunes recurrent dynamics into a ring to efficiently represent the circular variable θ. During training, the stable FP moves across the state space, gradually approaching the goal location where $\theta = 0$ (or equivalently, $\theta = 2\pi$). Meanwhile, an unstable FP, representing the free-hanging pendulum, emerges later in the training process and converges to the coordinates corresponding to the vertically-down pendulum position. Together, these FPs help the agent represent both its goal and key aspects of environmental physics within its recurrent dynamics. Note that the controller's stable FP corresponds to the unstable FP of the physical system, and vice versa. **(d)** [Left] **Trajectory of the stable FP moving across the state space in the PO environment**, colored by 10-episode-average reward at the corresponding time point. Interestingly, how close the FP is to its goal location seems to be correlated with reward. [Right] We quantify this trend over five seeds to find that indeed **the proximity of the FP to its final state is correlated with the episode reward-to-go** (the difference between final reward and episode reward at the corresponding time point. This demonstrates a statistically significant linear relationship between reward-to-go and FP-goal location proximity, directly linking representational geometry to task performance. **(e) Evolution of top-5 stimulus integration times τ_i over training steps in the PO environment.** Initially, we see an increase in τs, suggesting that the agent is extending its period of information integration to gather more stimulus information. This increase is followed by a decrease in τs, indicating a transition to more efficient memory usage that retains only task-relevant information in the shorter timescale. This pattern, which we call "explore then compress" , reflects the balance between thorough information integration during the exploration phase and a more focused, efficient retention of only task-relevant information in the compression phase. The error bar indicates 99% confidence interval in the estimation of τ.

References

Barak, O. (2017). Recurrent neural networks as versatile tools of neuroscience research. *Current opinion in neurobiology*, *46*, 1–6.

Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., & Zaremba, W. (2016). OpenAI gym. *arXiv preprint arXiv:1606.01540*.

Driscoll, L., Shenoy, K., & Sussillo, D. (2022). Flexible multitask computation in recurrent networks utilizes shared dynamical motifs. *bioRxiv*, 2022–08.

Hocker, D. L., Constantinople, C. M., & Savin, C. (2024, January). *Curriculum learning inspired by behavioral shaping trains neural networks to adopt animal-like decision making strategies.* bioRxiv. doi: 10.1101/2024.01.12.575461

Maheswaranathan, N., Williams, A., Golub, M., Ganguli, S., & Sussillo, D. (2019). Reverse engineering recurrent networks for sentiment classification reveals line attractor dynamics. In *Advances in neural information processing systems* (pp. 15696–15705).

Marschall, O., & Savin, C. (2023, September). *Probing learning through the lens of changes in circuit dynamics* (Preprint). Neuroscience. doi: 10.1101/2023.09.13.557585

Ni, T., Eysenbach, B., & Salakhutdinov, R. (2021). Recurrent model-free rl is a strong baseline for many POMDPs. *arXiv preprint arXiv:2110.05038*.

Rajan, K., & Abbott, L. F. (2006). Eigenvalue spectra of random matrices for neural networks. *Physical Review Letters*, *97*(18), 188104.

Rajan, K., Harvey, C. D., & Tank, D. W. (2016). Recurrent network models of sequence generation and memory. *Neuron*, *90*(1), 128–142.

Singh, S. H., van Breugel, F., Rao, R. P., & Brunton, B. W. (2023). Emergent behaviour and neural dynamics in artificial agents tracking odour plumes. *Nature Machine Intelligence*, *5*(1), 58–70.

Sussillo, D., & Barak, O. (2013). Opening the black box: low-dimensional dynamics in high-dimensional recurrent neural networks. *Neural computation*, *25*(3), 626–649.

Vyas, S., Golub, M. D., Sussillo, D., & Shenoy, K. V. (2020). Computation through neural population dynamics. *Annual Review of Neuroscience*, *43*, 249–275.