

Partially recurrent neural networks maximize performance and minimize wiring

Marcus Ghosh & Dan F. M. Goodman

Department of Electrical and Electronic Engineering, Imperial College London, United Kingdom

Abstract

Many circuits in the brain are bidirectional and sparse. Meaning that signals flow from sensory inputs to later areas and back; yet, between any two connected areas there exist some but not all pathways. What advantages or disadvantages do these architectures confer, compared to feedforward or fully connected networks? To address this question, we introduce a new class of partially recurrent neural network architectures, between these two extremes. An exhaustive search of these architectures reveals significant differences in their performance, learning speed and robustness to noise. Though, surprisingly, many perform as well as, or even better than, fully connected networks, despite having fewer parameters (a proxy for wiring cost). To explain these functional differences, we show that different architectures learn distinct input-output mappings and memory dynamics, both of which are predictive of function. Ultimately, our results demonstrate that partial recurrence allows networks to maximize performance with minimal wiring. More broadly, our work provides a general framework for linking network structure to function.

Keywords: recurrent neural networks; reinforcement learning

Introduction

Experimental data, across species, has demonstrated that neural circuits are bidirectional, yet sparse (Cook et al., 2019; Lin et al., 2024). How do these structural properties shape network function? Determining this *in vivo* is challenging, but *in silico* one can design and compare networks with different structures. E.g. recurrent neural networks with low-rank (Mastrogiuseppe & Ostojic, 2018; Liu et al., 2024) or modular (Achterberg, Akarca, Strouse, Duncan, & Astle, 2023; Béna & Goodman, 2025) weight matrices. Functional results from these approaches suggest that either many structures are degenerate, i.e., they yield the same outputs (Edelman & Gally, 2001; Marder & Taylor, 2011), or that densely connected networks, with more parameters, would always be more performant. However, more challenging tasks or alternative measures of function, e.g. robustness to noise, could reveal differences between these structures, and perhaps even scenarios in which sparser networks outperform dense networks.

Methods

To explore this, we compare how neural networks with different architectures perform on a set of maze tasks. In these tasks, we simulate networks as agents in grid environments, with paths the agents can move on, impassable walls and a

set of noisy sensors, which provide clues about the shortest path through the maze. To perform these tasks, each network implements a three-step sensation-action loop. First, they sense their local environment. Often, we add independent Gaussian noise to each sensor at each time step. Next, they pass these inputs through a series of units and weighted connections. Finally, networks choose and implement an action; either pausing or moving in a cardinal direction. To optimise each network's weights we used deep Q-learning (Jensen, 2024; Mnih et al., 2013). To quantify performance / fitness we define a function which ranges from 0 - ending the trial at the furthest point from the goal, to 1 - taking the shortest path from start to goal. To measure learning speed we score networks from 0 - not learning the task, to 100 - learning the task as fast as possible. To quantify robustness we measure each trained network's fitness as a function of increasing sensor noise, and take the area under this curve. For statistical comparisons across architectures we use Mann-Whitney U tests with corrections for multiple comparisons.

Results

Partially recurrent neural networks

Given an artificial neural network with: a layer of inputs, a hidden layer and an output layer, there are 9 possible connection pathways / weight matrices (Figure 1A). Assuming the feedforward connections (W_{ih} and W_{ho}) are always present, and each of the other 7 matrices can be present in any combination yields $2^7 = 128$ distinct architectures. In this model, excluding all 7 additional weight matrices yields a pure layer-layer feedforward network. While, including all 7 generates a fully connected network - in which every unit connects bidirectionally to every other unit (Figure 1B). Between these two extremes, are 126 architectures. We term these architectures *partially recurrent* as information flow can be *bidirectional* - from input sensors to output actions and back, yet *sparse* - in the sense that it cannot flow via all possible paths.

Different architectures realise different fitness, sample efficiency and robustness to noise

We began by training all 128 architectures on a maze task, which requires some memory. To capture intra-architecture variability, across random seeds (Patterson, Neumann, White, & White, 2024), and to enable fair inter-architecture comparisons we trained 50 networks per architecture; yielding 6,400 trained networks.

Fully connected networks learned the task well, achieving a median fitness of 0.93 ± 0.01 std (Figure 1C). By contrast, only two architectures were consistently unable to learn

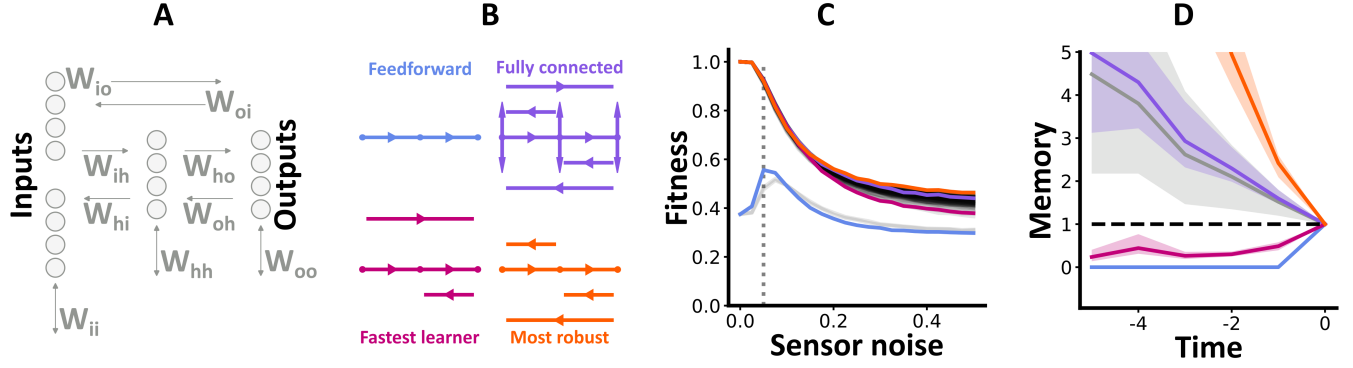


Figure 1: **A.** For a neural network with an input layer (i), hidden layer (h) and output layer (o), there are 9 possible connection pathways / weight matrices; labelled as $W_{from-to}$. **B.** Assuming the feedforward pathways are always present, and the other 7 can be present or absent yields 128 architectures. Here, we highlight 4 of these architectures; with each layer/weight matrix represented by a single node/arrow. **C.** Different architectures are differently robust to noise. We plot the median fitness per architecture (across 50 networks) as a function of increasing sensor noise. **D.** Different architectures learn distinct memory dynamics. We plot the median (solid line) and interquartile range (shaded surround) memory per architecture as a function of time. In panels C and D all 128 architectures are plotted in grey, and the 4 from panel B are highlighted.

the task well: the pure layer-layer feedforward architecture (0.48 ± 0.02 std), and an architecture with additional forward skip connections W_{io} (0.49 ± 0.03 std). Notably, larger feed-forward networks, with the same number of learnable parameters as the fully connected networks, were still unable to learn the task well (0.56 ± 0.01 std). This is due to the fact that neither of these architectures retain any information from prior time steps, and so cannot solve tasks which require some form of memory. Surprisingly, the remaining 125 architectures all achieved median fitnesses between 0.91 and 0.93. Though none performed significantly better than the fully connected network. By contrast, 19 architectures learned significantly faster than the fully connected network, and 9 were significantly more robust to noise. The fastest learning and most robust architectures are shown in Figure 1B.

Together, these results demonstrate that many architectures can implement robust behaviours; equivalent, or even better than a fully-connected network. Despite having distinct structures and far fewer learnable parameters (in some cases as low as a quarter of the fully connected network).

Different architectures learn distinct input-output mappings and memory dynamics

To understand *why* different architectures function differently, we measured two properties from each trained network (using data from 1,000 test trials per network).

First, we define a measure of a network's sensitivity at time t to its inputs at time $t+k$ ($k \leq 0$) by how much a change in those inputs leads to a change in those outputs. More precisely, the Frobenius norm of the Jacobian matrix of the partial derivatives of the outputs with respect to the inputs:

$$S(t, k) = \left\| \frac{\partial O_t}{\partial I_{t+k}} \right\|_F$$

We then define a network's *input sensitivity* as the mean value of the sensitivity at time t to inputs at time t : $S = \langle S(t, 0) \rangle$. We also define the *memory at lag k* to be the ratio of the sensitivity to past inputs compared to current inputs, $M_k = \langle S(t, k) / S(t, 0) \rangle$.

As such, when $M_k = 0$, the inputs from a previous time, have no influence on the outputs at the current time. In feed-forward networks, for example, $M_0 = 1$ at lag 0, but $M_k = 0$ for $k < 0$ - as prior inputs have no influence on the network's outputs at a given time (Figure 1D). By contrast, values $M_k > 0$ indicate an influence.

Using these metrics we found that different architectures are differently sensitive to their inputs and learn distinct memory dynamics (Figure 1D). For example, the fastest learning architecture weights its current inputs more highly than prior inputs. By contrast, the most robust architecture relies more on its past inputs, and so implements a positive feedback strategy; which works in this particular set of mazes. Most networks fall somewhere between these two extremes.

Finally, we trained random forest models to predict each network's function (fitness, sample efficiency, robustness) from either of these metrics (S , M). All of these models perform significantly better than chance, showing that these metrics meaningfully describe network computation.

Discussion

Together, our results demonstrate that networks with specific structures can perform as well as, or even better than fully connected networks, while using far fewer parameters. Thus, our work provides a starting point for considering *why* we observe specific types of connectivity *in vivo*. In this direction, these partially recurrent architectures, and measures of network sensitivity and memory, provide a new framework for exploring structure-function relations in neuroscience.

Acknowledgements

MG is supported by the Eric and Wendy Schmidt AI in Science Postdoctoral Fellowship, a Schmidt Sciences program.

References

- Achterberg, J., Akarca, D., Strouse, D., Duncan, J., & Astle, D. E. (2023). Spatially embedded recurrent neural networks reveal widespread links between structural and functional neuroscience findings. *Nature Machine Intelligence*, 5(12), 1369–1381.
- Béna, G., & Goodman, D. F. (2025). Dynamics of specialization in neural modules under resource constraints. *Nature Communications*, 16(1), 187.
- Cook, S. J., Jarrell, T. A., Brittin, C. A., Wang, Y., Bloniarz, A. E., Yakovlev, M. A., ... others (2019). Whole-animal connectomes of both *Caenorhabditis elegans* sexes. *Nature*, 571(7763), 63–71.
- Edelman, G. M., & Gally, J. A. (2001). Degeneracy and complexity in biological systems. *Proceedings of the national academy of sciences*, 98(24), 13763–13768.
- Jensen, K. T. (2024). An introduction to reinforcement learning for neuroscience. *Neurons, Behavior, Data, Analysis and Theory*.
- Lin, A., Yang, R., Dorkenwald, S., Matsliah, A., Sterling, A. R., Schlegel, P., ... others (2024). Network statistics of the whole-brain connectome of *Drosophila*. *Nature*, 634(8032), 153–165.
- Liu, Y. H., Baratin, A., Cornford, J., Mihalas, S., Shea-Brown, E., & Lajoie, G. (2024). How connectivity structure shapes rich and lazy learning in neural circuits. *ArXiv*, arXiv–2310.
- Marder, E., & Taylor, A. L. (2011). Multiple models to capture the variability in biological neurons and networks. *Nature neuroscience*, 14(2), 133–138.
- Mastrogiuseppe, F., & Ostojic, S. (2018). Linking connectivity, dynamics, and computations in low-rank recurrent neural networks. *Neuron*, 99(3), 609–623.
- Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., & Riedmiller, M. (2013). Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*.
- Patterson, A., Neumann, S., White, M., & White, A. (2024). Empirical design in reinforcement learning. *Journal of Machine Learning Research*, 25(318), 1–63.