The AAAI 2017 Spring Symposium on
Science of Intelligence: Computational Principles of Natural and Artificial Intelligence
Technical Report SS-17-07

Markov Transitions between Attractor States in a Recurrent Neural Network

Jeremy Bernstein[*]
Computation and Neural Systems
California Institute of Technology, USA
bernstein@caltech.edu

Ishita Dasgupta[*]
Department of Physics
Harvard University, USA
ishitadasgupta@g.harvard.edu

David Rolnick[*]
Department of Mathematics
Massachusetts Institute of Technology, USA
drolnick@mit.edu

Haim Sompolinsky[*][†]
The Edmond and Lily Safra Center for Brain Sciences
Hebrew University of Jerusalem, Israel
haim@fiz.huji.ac.il

Stochasticity is an essential part of explaining the world. Increasingly, neuroscientists and cognitive scientists are identifying mechanisms whereby the brain uses probabilistic reasoning in representational, predictive, and generative settings. But stochasticity is not always useful: robust perception and memory retrieval require representations that are immune to corruption by stochastic noise. In an effort to combine these robust representations with stochastic computation, we present an architecture that generalizes traditional recurrent attractor networks to follow probabilistic Markov dynamics between stable and noise-resistant fixed points.

Motivation

With the advancement of probabilistic theories of human cognition (Griffiths et al. 2010), there has been increasing interest in neural mechanisms that can represent and compute these probabilities. Several new models of neural computation carry out Bayesian probabilistic inference taking into account both data and prior knowledge, and can represent uncertainty about the conclusions they draw (Ma, Beck, and Pouget 2008; Pecevski, Buesing, and Maass 2011; Shi et al. 2010). In many tasks, neural mechanisms are required that can transition stochastically to a new state depending on the current state: for example, to predict the path of a moving object (Vul et al. 2009), gauge the effect of a collision (Sanborn and Griffiths 2009), or estimate the dynamic motion of fluids (Bates et al. 2015), as well as in the general context of carrying out correlated sampling over a posterior distribution (Gershman, Vul, and Tenenbaum 2012; Bonawitz et al. 2014; Denison et al. 2013). The Markov transition probabilities in these cases are dictated by knowledge of the world. The stochasticity of transitions allows decisions that are tempered by uncertainty, rather than making a "best guess" or point estimate that is agnostic to uncertainty and is chosen deterministically based on some measure of optimality. Further, Markov chain Monte Carlo methods (Neal 1993) allow us to engineer a Markov chain with stationary distribution equal to any distribution of interest. Therefore a simple Markov chain with the right transition probabilities can also form the basis for neurally plausible probabilistic inference on a discrete state space.

It is important here to distinguish between stochasticity in our perception or neural representation of states, and stochasticity incorporated into a computational step. The first is unavoidable and due to noise in our sensory modalities and communication channels. The second is inherent to a process the brain is carrying out in order to make probabilistic judgments, and represents useful information about the structure of the environment. While it is difficult to tease apart these sources of noise and variability, (Beck et al. 2012) suggest that sensory or representational noise is not the primary reason for trial-to-trial variability seen in human responses and that there are other sources of stochasticity arising from the process of inference that might be more important and influential in explaining observed behavioral variability. Humans are in fact remarkably immune to noise in percepts - for example when identifying occluded objects (Johnson and Olshausen 2005) and filtering out one source of sound amid ambient noise (Handel 1993).

Hopfield networks represent an effective model for storage and representation that is largely immune to noise; different noisy or partial sensory percepts all converge to the same memory as long as they fall within that memory's basin of attraction. These "memory" states are represented in a distributed system and are robust to the death of individual neurons. Stochastic transitions in Hopfield networks therefore are a step towards stochastic computation that still ensures a noise-robust representation of states.

The Markov chain dynamics we model also have applications in systems where experimental verification is more lucid. For example, the Bengalese finch's song has been effectively modeled as a hidden Markov model (Jin and Kozhevnikov 2011). While deterministic birdsong in the

[*] All authors contributed equally to this work.

[†] Also: Center for Brain Science, Harvard University, USA

zebra finch has previously been modeled by feedforward chains of neurons in HVC (Long and Fee 2008), our network provides a potential neural model for stochastic birdsong. Further, its specific structure has possible parallels in songbird neural architecture, as we later detail.

Background

A Hopfield network (Hopfield 1982) is a network of binary neurons with recurrent connections given by a symmetric synaptic weight matrix, J_{ij}. The state x_i of the ith neuron is updated according to the following rule:

$$x_i \leftarrow \text{sign}\left(\sum_{j=1}^{n} J_{ij} x_j\right) \tag{1}$$

With this update rule, every initial state of the network deterministically falls into one of a number of stable fixed points which are preserved under updates. The identity of these fixed points (*attractors* or *memories*) can be controlled by appropriate choice of J_{ij}, according to any of various learning rules (Hebb 2005; Rosenblatt 1957; Storkey 1997; Hillar, Sohl-Dickstein, and Koepsell 2012). If the network is initialized at a corrupted version of a memory, it is then able to converge to the true memory, provided that the corrupted/noisy initialization falls within the true memory's basin of attraction. This allows Hopfield networks to be a model for *content-addressable, associative* memory.

Due to symmetry of weights, a traditional Hopfield network always converges to a stable attractor state. By adding asymmetric connections, it is possible to induce transitions between the attractor states. (Sompolinsky and Kanter 1986) show that a set of *deterministic* transitions between attractor states can be learned with a Hebbian learning rule, by means of time-delayed *slow connections*. Here, the transition structure is built into the synapses of the network and is not stochastic. The challenge we address in this paper is to leverage what we know from past work about deterministic transitions in attractor networks and combine it with a source of noise to make these transitions stochastic, with controllable Markov probabilities for each transition.

Network architecture

We propose a network consisting of three parts: A *memory network*, a *noise network*, and a *mixed network* (see Fig. 1). The memory network taken by itself is an attractor network with stabilizing recurrent connections; it stores states of the Markov chain as attractors. The noise network also stores a number of attractor states (the *noise states*); in its case, the transitions between attractors occur uniformly at random.

The mixed network is another attractor network, which receives input from both the memory and noise networks, according to fixed random weights. The attractors (*mixed states*) of the mixed network are chosen according to the memory and noise states; thus, a different pair of memory state and noise state will induce the mixed network to fall into a different attractor. The memory network receives input from the mixed network, which induces it to transition between the memory attractor states.

The key insight in our design is that given the combined state of the noise and memory networks (as captured in the mixed network), the next memory state is fully determined. Stochasticity arises from resampling the noise network and allowing it to fall uniformly at random into a new attractor. This is in fact the sole source of stochasticity in the model, and it is in a sense analogous to the reparameterization trick used in (Kingma and Welling 2013).

In order for transitions between memory states to be determined by the state of mixed network, the attractors for the mixed network should be linearly separable. A simple concatenation of memory and noise states would result in a strong linear dependence between mixed states, making them difficult to linearly separate (Cover 1965). We recover linear separability in our model by instead constructing the mixed network as a random projection of memory and noise states into a higher dimensional space (Barak, Rigotti, and Fusi 2013).

The connections from the mixed network back to the memory network that induce the transition are *slow connections* (see (Sompolinsky and Kanter 1986)); they are time-delayed by a constant τ and are active at intervals of τ. This allows the memory network to stabilize its previous state before a transition occurs. Thus, at every time step, each memory neuron takes a time-delayed linear readout from the mixed representation, adds it to the Hopfield contribution from the memory network and passes the sum through a threshold non-linearity.

Formally, the dynamics are given by the following equations, where x_i^M $(0 \le i < n_M)$, x_j^N $(0 \le j < n_N)$, and x_k^Q $(0 \le k < n_Q)$ denote states of neurons in the memory network, noise network, and mixed network, respectively. The function $\delta_\tau^{\text{mod}}(t)$ is 1 when $t \equiv 0 \pmod{\tau}$, otherwise 0; and the notation $x(t-\tau)$ denotes the state x at time $t-\tau$ (otherwise assumed to be time t). The function $\nu(t,\tau)$ represents a noise function that is resampled uniformly at random at intervals of τ.[1]

$$\begin{aligned}
x_i^M &\leftarrow \text{sign}\left(\sum_{\ell=1}^{n_M} J_{i\ell}^M x_\ell^M + \delta_\tau^{\text{mod}}(t) \sum_{k=1}^{n_Q} J_{ik}^{MQ} x_k^Q(t-\tau)\right), \\
x_j^N &\leftarrow \text{sign}\left(\sum_{\ell=1}^{n_N} J_{j\ell}^N x_\ell^N + \nu(t,\tau)\right), \\
x_k^Q &\leftarrow \text{sign}\left(\sum_{\ell=1}^{n_Q} J_{k\ell}^Q x_\ell^Q + \sum_{i=1}^{n_M} J_{ki}^{QM} x_i^M + \sum_{j=1}^{n_N} J_{kj}^{QN} x_j^N\right),
\end{aligned}$$

The weight matrices J^M, J^N, J^Q, and J^{MQ} are learned (see below), while J^{QM} and J^{QN} are random, with $n_Q \gg n_M, n_N$.

We implement the noise network as a *ring attractor* — a ring of neurons where activating any contiguous half-ring yields an attractor state. Here we have adapted the model described in (Ben-Yishai, Bar-Or, and Sompolinsky 1995)

[1] This "clocked" activity, while not biologically implausible, might be unnecessary; future work might aim to replace it, perhaps by combining time-delayed neurons with a *sparse*, high-dimensional projection to the mixed representation.

to the discrete setting according to the following dynamics:

$$J_{ij}^N = \begin{cases} -1 & \text{for } n_N/4 \leq |i-j| \leq 3n_N/4, \\ +1 & \text{otherwise}, \end{cases}$$

for $1 \leq i, j \leq n_N$, where we require that n_N be even. There are, then, $a_N = n_N$ attractor states A_i, which take the form:

$$A_i = \begin{cases} x_{i+k}^N = -1 & \text{for } 0 \leq k < n_N/2, \\ x_{i+k}^N = 1 & \text{for } n_N/2 \leq k < n_N, \end{cases}$$

where indices are taken modulo n_N.

Another possible construction for the noise network is simply to have a small set of randomly activated neurons with no recurrent stabilizing connections. In this construction, the number of noise attractor states is exponential in the number of noise neurons, allowing higher precision in probabilities for the same number of noise neurons. However, this construction has the disadvantage that it is highly sensitive to the perturbation of single neurons, and so it may be difficult to distinguish between incidental noise and a resampling of the noise network.

The components in our network all have biological analogues. We use slow neurons to prompt transitions between memories only after the memories have been allowed to stabilize. These could be implemented via the *autapses* in (Seung et al. 2000). We use large random expansions to increase linear separability of states, as suggested in (Babadi and Sompolinsky 2014). Also, the noise network in our architecture has a promising parallel in the LMAN region of the songbird brain, which has been linked to generating variability in songs during learning (Ölveczky, Andalman, and Fee 2005). Alternatively, the noise network in our model could just be any uncorrelated brain region.

Learning

There are several sets of weights that must be determined within our model. Those denoted J^M, J^N, J^Q, and J^{MQ} above are learned, while J^{QM} and J^{QN} are random.

The recurrent connections within the memory network (J^M) and noise network (J^N) are learned using Hebb's rule, ensuring that each of the three subnetworks has the desired attractor states. The weights for slow synapses from the mixed network to the memory network (J^{MQ}) are chosen according to methods described in (Sompolinsky and Kanter 1986) for inducing deterministic sequences of attractors. The weights within the mixed network (J^Q) are learned using the perceptron learning rule, yielding a larger capacity than the Hebbian approach used in (Sompolinsky and Kanter 1986).

Finally, it is necessary that we determine which (`memory`, `noise`) state pairs should transition to which new memories. For a desired transition $S_1 \to S_2$ between memory states, having probability p in the Markov chain, we assign (approximately) a p-fraction of noise states, so that (S_1, N) induces a transition to S_2 for all N in the p-fraction. Thus, several noise states, in the presence of a particular memory state, could result in the same transition; the number of noise states assigned to a transition is proportional to the probability of that transition. Probabilities may be approximated to within an accuracy of $\epsilon = O(1/a_N)$, where a_N is the number of attractor states in the noise network.

Let us consider the biological feasibility of our learning approach. We use a Hebbian rule for stabilizing the patterns, in keeping with the established hypothesis of Hebbian learning within the brain. For learning state transitions, we have so far opted to use the perceptron rule in the interest of increasing capacity. This, however, requires storing and iterating over a list of pairs of the form (`memory`, `noise`). It is perhaps more biologically feasible to use an online algorithm for which such a list need not be learned and stored, where learning proceeds by sampling trajectories from the desired Markov chain.

Such an online learning rule indeed seems possible if we use Hebbian learning for the state transitions. Specifically, every time a transition occurs in the real world, we strengthen the corresponding connections in our network. Since our transitions take the form of (`memory + noise`) → `new memory`, the noise state used in the transition is simply whatever (arbitrary) state the noise network is in at that moment. A slight weakness to this approach is the network must recognize when a particular transition had already been learned, to prevent overwriting; however, this seems by no means biologically insurmountable.

Future directions

The connection weights from the mixed network to the memory network must discriminate between the various mixed states in order to elicit the right transition. We use a perceptron learning rule to learn these weights and use a high-dimensional random projection to form our mixed state, to increase the number of mixed states that are linearly separable (Barak, Rigotti, and Fusi 2013; Cover 1965), thereby allowing us to encode a larger number of transitions into our network. In effect, points that were close together in the original space are far apart in the higher dimensional space, and thus easy to separate. This comes with a downside, however, since small, accidental perturbations in the original state will likewise be blown up by the random expansion. In (Barak, Rigotti, and Fusi 2013) this problem is referred to as a generalization-discrimination trade off. We did not encounter this problem in our preliminary simulations, since we did not consider noise in our neural update rule, Equation 1. This ensured that our system always fell to the very bottom of its attractor states, and in the statistical mechanics analogy corresponds to operating the system at zero temperature.

We plan to test empirically the limits of our system's performance with respect to the level of noise in our update rule (the temperature of our system), the expansion ratio of our random projection, and the number of mixed states that need to be correctly classified. Checking the agreement of our simulations with theoretical predictions will reveal more about the capabilities of a system of neural computations based on stochastic transitions between attractor states.

Our architecture can also implement Markov chain Monte Carlo, specifically some version of Gibbs sampling. Each new state is sampled from a distribution conditioned on the

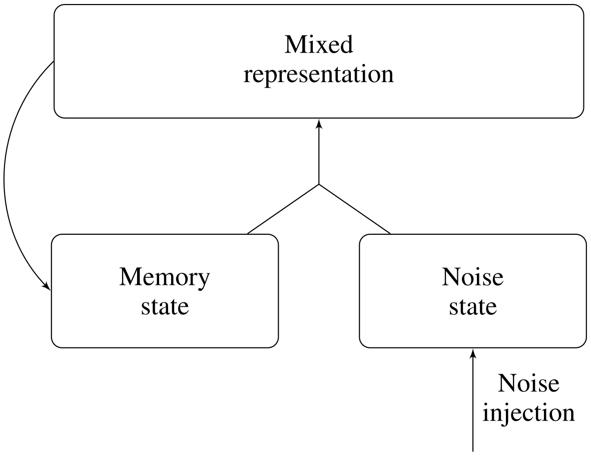

Figure 1: A schematic of our recurrent network. The current memory state is paired with a randomized noise state, and that pairing determines the next memory state. Since the noise state is sampled uniformly at random, it follows that the probability of a particular transition from memory state S_1 to S_2 is given by the fraction of noise states that pair with S_1 to produce S_2. To implement these transitions, we consider each memory neuron at time $t+1$ to be a perceptron readout of the mixed representation. To increase the separability ability of these perceptrons, the mixed representation is a large, random expansion of the (`memory`, `noise`) pairings. The state transitions operate on a slow timescale due to the slow neurons. Not pictured are the self-connections within the memory network, noise network, and mixed network that serve to stabilize the corresponding states on a fast timescale.

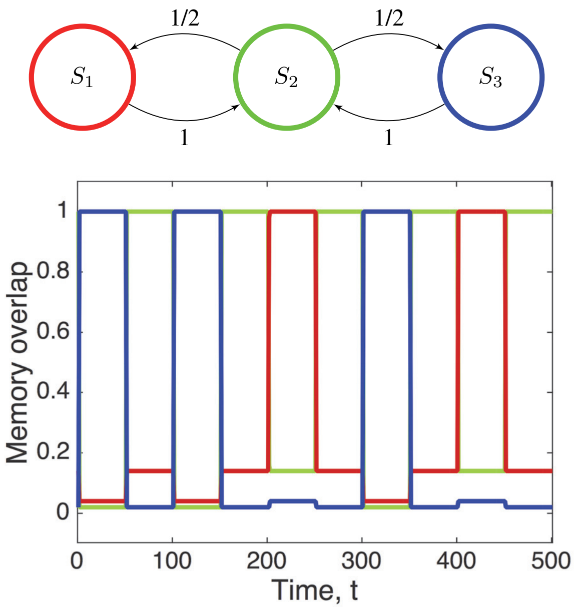

Figure 2: Top: a simple Markov chain, where the dynamics ensure the central state (green) is visited on alternate time steps with probability 1. The number of times state S1 is visited is therefore a binomial random variable $\sim B(n, p = 0.5)$. Bottom: results of our network simulating the dynamics of this Markov chain. We associate a distinct 'memory vector' with each Markov state S_i, and the plot shows the overlap of each such memory vector with the system state at every time step of the simulation. Averaging over 4000 state transitions, our simulation yielded empirical transition probabilities of $P(S_1|S_2) = 0.512 \approx P(S_3|S_2) = 0.488$. These values are within one standard deviation of the mean of the binomial distribution $\sim B(n = 2000, p = 0.5)$, indicating that our recurrent neural network is operating correctly. Note that we use $n = 2000$ instead of 4000 because half of the transitions go deterministically to state S_2.

current state - this is analogous to sampling from a conditional distribution as in Gibbs sampling. Future work could involve learning and representing these conditionals within our network and implementing a noise-robust stochastic sampler over discrete state spaces.

Acknowledgments

We would like to thank the Center for Brains, Minds, and Machines and the Marine Biological Laboratory at Woods Hole for their support during this research. We thank Samuel Gershman for useful comments and discussions. D.R. is additionally supported by NSF grant no. 1122374.

References

Babadi, B., and Sompolinsky, H. 2014. Sparseness and expansion in sensory representations. *Neuron* 83(5):1213–1226.

Barak, O.; Rigotti, M.; and Fusi, S. 2013. The sparseness of mixed selectivity neurons controls the generalization–discrimination trade-off. *The Journal of Neuroscience* 33(9):3844–3856.

Bates, C. J.; Yildirim, I.; Tenenbaum, J. B.; and Battaglia, P. W. 2015. Humans predict liquid dynamics using probabilistic simulation. In *Proceedings of the 37th annual conference of the cognitive science society*.

Beck, J. M.; Ma, W. J.; Pitkow, X.; Latham, P. E.; and Pouget, A. 2012. Not noisy, just wrong: the role of suboptimal inference in behavioral variability. *Neuron* 74(1):30–39.

Ben-Yishai, R.; Bar-Or, R. L.; and Sompolinsky, H. 1995. Theory of orientation tuning in visual cortex. *Proceedings of the National Academy of Sciences* 92(9):3844–3848.

Bonawitz, E.; Denison, S.; Gopnik, A.; and Griffiths, T. L. 2014. Win-Stay, Lose-Sample: A simple sequential algorithm for approximating Bayesian inference. *Cognitive Psychology* 74:35–65.

Cover, T. M. 1965. Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. *IEEE transactions on electronic computers* (3):326–334.

Denison, S.; Bonawitz, E.; Gopnik, A.; and Griffiths, T. L.

2013. Rational variability in children's causal inferences: The Sampling Hypothesis. *Cognition* 126(2):280–300.

Gershman, S. J.; Vul, E.; and Tenenbaum, J. B. 2012. Multistability and perceptual inference. *Neural computation* 24(1):1–24.

Griffiths, T. L.; Chater, N.; Kemp, C.; Perfors, A.; and Tenenbaum, J. B. 2010. Probabilistic models of cognition: Exploring representations and inductive biases. *Trends in cognitive sciences* 14(8):357–364.

Handel, S. 1993. *Listening: An introduction to the perception of auditory events.* The MIT Press.

Hebb, D. O. 2005. *The organization of behavior: A neuropsychological theory*. Psychology Press.

Hillar, C.; Sohl-Dickstein, J.; and Koepsell, K. 2012. Efficient and optimal binary Hopfield associative memory storage using minimum probability flow. *arXiv preprint arXiv:1204.2916*.

Hopfield, J. J. 1982. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the national academy of sciences* 79(8):2554–2558.

Jin, D. Z., and Kozhevnikov, A. A. 2011. A compact statistical model of the song syntax in Bengalese finch. *PLoS Comput Biol* 7(3):1–19.

Johnson, J. S., and Olshausen, B. A. 2005. The recognition of partially visible natural objects in the presence and absence of their occluders. *Vision research* 45(25):3262–3276.

Kingma, D. P., and Welling, M. 2013. Auto-encoding variational bayes. *CoRR* abs/1312.6114.

Long, M. A., and Fee, M. S. 2008. Using temperature to analyse temporal dynamics in the songbird motor pathway. *Nature* 456(7219):189–194.

Ma, W. J.; Beck, J. M.; and Pouget, A. 2008. Spiking networks for Bayesian inference and choice. *Current opinion in neurobiology* 18(2):217–222.

Neal, R. M. 1993. Probabilistic inference using Markov chain Monte Carlo methods.

Ölveczky, B. P.; Andalman, A. S.; and Fee, M. S. 2005. Vocal experimentation in the juvenile songbird requires a basal ganglia circuit. *PLoS Biol* 3(5):e153.

Pecevski, D.; Buesing, L.; and Maass, W. 2011. Probabilistic inference in general graphical models through sampling in stochastic networks of spiking neurons. *PLoS Comput Biol* 7(12):e1002294.

Rosenblatt, F. 1957. *The perceptron, a perceiving and recognizing automaton Project Para*. Cornell Aeronautical Laboratory.

Sanborn, A. N., and Griffiths, T. L. 2009. A Bayesian Framework for Modeling Intuitive Dynamics. In *Proceedings of the 31st Annual Conference of the Cognitive Science Society*, 1145–1150.

Seung, H. S.; Lee, D. D.; Reis, B. Y.; and Tank, D. W. 2000. The autapse: A simple illustration of short-term analog memory storage by tuned synaptic feedback. *Journal of Computational Neuroscience* 9(2):171–185.

Shi, L.; Griffiths, T. L.; Feldman, N. H.; and Sanborn, A. N. 2010. Exemplar models as a mechanism for performing Bayesian inference. *Psychonomic bulletin & review* 17(4):443–464.

Sompolinsky, H., and Kanter, I. 1986. Temporal association in asymmetric neural networks. *Physical Review Letters* 57(22):2861.

Storkey, A. 1997. Increasing the capacity of a Hopfield network without sacrificing functionality. In *International Conference on Artificial Neural Networks*, 451–456. Springer.

Vul, E.; Alvarez, G.; Tenenbaum, J. B.; and Black, M. J. 2009. Explaining human multiple object tracking as resource-constrained approximate inference in a dynamic probabilistic model. In *Advances in Neural Information Processing Systems*, 1955–1963.