BIOLOGICALLY PLAUSIBLE INFOMAX LEARNING IN RECURRENT SPIKING NETWORKS

Chong Wang

November 20, 2020

1 Abstract

Being able to form good representations of the world is an important prerequisite for high-level cognition. We examine the problem of representation learning at a neuronal level, namely, how can streams of input be mapped to spike patterns (formalized as binary vectors) that are maximally informative. To do so, we use a Stochastic Recurrent Neural Network with biologically feasible dynamics. The goodness of the representation is measured by the mutual information between the network's past inputs and its current spiking output. To maximize the mutual information (InfoMax), we use a biologically plausible approximation to the Real-Time Recurrent Learning algorithm. We derive a learning rule that is functionally similar to the sum of a Hebbian and an anti-Hebbian term. In an unsupervised learning task, the network is able to learn representations of input sequences and develop homeostatic firing rates. In a supervised learning task, the InfoMax objective can be used as a regularizer in addition to the maximum likelihood objective function, which leads to better performance compared to maximum likelihood learning alone.

2 Methods

2.1 Network Model

We use a Recurrent Neural Network as our model, given by the following equations:

$$\frac{1}{\tau_{v_i}}\dot{v_i} = -v_i + \sum_j W_{ij}^{rec} h_j + \sum_j W_{ij}^{in} x_j + b_i$$

$$h_i = H(v_i + \epsilon_i)$$

$$\epsilon_i \sim Logistic(0, 1)$$

Here x is the external input and h is the spike pattern. v is the membrane potential, τ_v is a vector of inverse membrane time constants. We randomly initialize it as $\tau_{v_i} \sim \mathcal{N}(\widetilde{\tau_v}, \sigma_\tau)$ with hyperparameters $\widetilde{\tau_v}$ and σ_τ. This allows the extraction of information at multiple timescales. W^{rec} and W^{in} are the recurrent and input weights respectively.

W^{rec}'s diagonals are fixed to be 0 to exclude self-to-self recurrent connections. It is initialized sparsely where each weight has a $1-p$ chance of being zero, with model parameter p. This sparsity is not enforced during learning, so initially zero weights can have non-zero weights after training. $H()$ is the Heaviside step function which implements a deterministic spiking function. Using the reparameterization trick, the stochasticity of the spiking rule is attributed to the logistic noise with parameters $(0, 1)$[8], the effect of which will be seen in the derivation of the learning rule.

Descretizing the network dynamics, we arrive at

$$v_{t,i} = (1 - \tau_{v_i})v_{t-1,i} + \tau_{v_i}(\sum_j W^{rec}_{ij} h_{t-1,j} + \sum_j W^{in}_{ij} x_{t,j} + b_i) \tag{1}$$

$$h_{t,i} = H(v_{t,i} + \epsilon_{t,i}) \tag{2}$$

$$\epsilon_{t,i} \sim Logistic(0, 1) \tag{3}$$

In the supervised setting, there is also a readout layer given by

$$o_t = f(Uh_t)$$

with a readout matrix U and activation function f which varies by the task.

2.2 Mutual Information Objective

At each time step, the RNN learns to optimize the mutual information objective given by

$$\begin{aligned} I \equiv I(h_t; h_{<t}, x_{\leq t}) &= \sum_{x_{\leq t}, h_{\leq t}} p(h_{\leq t}, x_{\leq t}) \log \frac{p(h_{\leq t}, x_{\leq t})}{p(h_t)p(h_{<t}, x_{\leq t})} \\ &= \mathbb{E}_{x_{\leq t}, h_{<t}} \left[\sum_{h_t} p(h_t|x_{\leq t}, h_{<t}) \log \frac{p(h_t|x_{\leq t}, h_{<t})}{p(h_t)} \right] \\ &= \mathbb{E}_{x_{\leq t}, h_{<t}} \left[\sum_{h_t} p(h_t|x_{\leq t}, h_{<t}) \log \frac{p(h_t|x_{\leq t}, h_{<t})}{\mathbb{E}_{x_{\leq t}, h_{<t}}[p(h_t|x_{\leq t}, h_{<t})]} \right] \end{aligned}$$

We model the probability of spiking $\sigma_{t,i} = p(h_{t,i} = 1|x_{\leq t}, h_{<t})$ with our model, where

$$p(h_{t,i} = 1|x_{\leq t}, h_{<t}) = \frac{1}{1 + e^{-v_{t,i}}}$$

Sampling from this distribution is equivalent to the spiking rule given by the eq. (2)-(3) [8]. In a supervised task setting, where we are given sequences of input and a target for each timestep $\{(x_{\leq T}, y_{\leq T})\}$, we add the mutual information objective as a regularizer to the following objective

$$E = \sum_{h_t} p(h_t|x_{\leq t}, h_{<t}) \log p(o_t = y_t|h_t)$$

This is a lower bound on the log-likelihood, following a similar derivation as in [1]

$$\begin{aligned} \log p(o_t = y_t|x_{\leq t}, h_{<t}) &= \log \sum_{h_t} p(o_t = y_t|h_t)p(h_t|x_{\leq t}, h_{<t}) \\ &\geq \sum_{h_t} p(h_t|x_{\leq t}, h_{<t}) \log p(o_t = y_t|h_t) \end{aligned}$$

This can be approximated by a single sample (sampling a spike pattern). Together the objective becomes

$$\max L = E + \beta I$$

2.3 Learning Rule

2.3.1 The Supervised Objective

We use Stochastic Gradient Descent to train the RNN. Consider the gradient at each timestep. For the readout matrix, the update rule follows a simple delta-rule.

$$\Delta U_{t,ij} \propto (o_{t,i} - y_{t,i}) h_{t,j}$$

Note that using the reparameterization trick, we can rewrite the objective E as

$$\begin{aligned} E &= \mathbb{E}_{h_t|x_{\leq t}, h_{<t}} \Big[\log p(\hat{y}_t = y_t | h_t) \Big] \\ &= \mathbb{E}_{\epsilon} \Big[\log p(\hat{y}_t = y_t | h_t) \Big] \end{aligned}$$

So taking the derivative

$$\begin{aligned} \frac{\partial}{\partial W_{ij}} E &= \mathbb{E}_{\epsilon} \Big[\frac{\partial}{\partial W_{ij}} \log p(o_t = y_t | h_t = H(v_t + \epsilon)) \Big] \\ &= \mathbb{E}_{\epsilon} \Big[\frac{\partial}{\partial h_{t_i}} \log p(o_t = y_t | h_t) \big|_{h_{t,i} = H(v_{t,i} + \epsilon_{t,i})} \cdot \frac{\partial h_{t,i}}{\partial W_{ij}} \Big] \\ &\approx \mathbb{E}_{\epsilon} \Big[(\sum_k e_k U_{ki}) (\sigma'_{t,i}) \frac{\partial v_{t,i}}{\partial W_{ij}} \Big] \end{aligned}$$

Here $e_k = o_{t,k} - y_{t,k}$ and $\sigma'_{t,i} = \frac{\partial \sigma_{t,i}}{\partial v_{t,i}}$. We use the straight-through estimator [5, 12] to replace the gradient of the step function with the gradient of the sigmoid in the backward pass. Also note that by backpropagation through time [15], the derivative $\frac{\partial v_t}{\partial W}$ has to be calculated by unrolling the network backward in time, which is not biologically plausible. The real-time recurrent learning algorithm and its recently developed approximations [16, 14, 3] allow for online computing of the gradient but is still biologically implausible due to the global dependence of the learning rule (for a review, see [10]). For example, the exact gradient w.r.t. a recurrent weight is given by the recursive relation:

$$\frac{\partial v_{t,i}}{\partial W_{jk}} = (1 - \tau_{v_i}) \frac{\partial v_{t-1,i}}{\partial W_{jk}} + \tau_{v_i} (\delta_{ij} h_{t-1,k} + \sum_m W^{rec}_{im} \frac{\partial h_{t-1,m}}{\partial v_{t-1,m}} \frac{\partial v_{t-1,m}}{\partial W_{jk}})$$

Calculating this exact gradient requires maintaining and operating on a 3D tensor $\frac{\partial v_{t,i}}{\partial W_{jk}}$, which renders it inefficient and biologically implausible. Inspired by newly developed local approximations [2, 11, 7], we define an *eligibility trace* for each input and recurrent synapse $e_{t,ij} \approx \frac{\partial v_{t,i}}{\partial W_{ij}}$. All other entries in the 3D tensor are set to 0. The eligibility traces are updated using the following rule (for the eligibility traces of the recurrent weight matrix, and similarly for the input weights)

$$e_{t,ij} = (1 - \tau_{v_i}) e_{t-1,ij} + \tau_{v_i} h_{t-1,j}$$

Notice that each eligibility trace corresponds to a exponentially filtered presynaptic spike train, with the filter's time constant being the membrane time constant of the postsynaptic neuron. The update rule for the input and recurrent weights from the supervised objective E takes the form of the product of a pre-synaptic term, a post-synaptic term, and the error feedback from the next layer.

$$\Delta W_{t,ij} \propto (\sum_k e_k U_{ki}) \sigma'_{t,i} e_{t,ij}$$

2.3.2 The Mutual Information Objective

The term $p(h_t) = \mathbb{E}_{x_{\leq t}, h_{<t}}[p(h_t|x_{\leq t}, h_{<t})]$ in I is intractable to calculate. Using techniques of variational inference [13], lower bounds of the mutual information requires either a encoder-decoder architecture, or an approximation based on unbiased samples. The current model doesn't contain an explicit decoder, and approximations based on sample trajectories requires information that is not local in time.

We use the following approximations $p(h_t) \approx \prod_i \mathbb{E}_{x_{\leq t}, h_{<t}}[p(h_{t,i}|x_{\leq t}, h_{<t})] \approx \prod_i r(h_{t,i})$, where $r(h_{t,i} = 1)$ equals a running average of the previous $\sigma_{\cdot,i}$'s of each individual neuron. We call this running average $\rho(\sigma_{t,i})$. This in fact results in an upper bound of the mutual information:

$$
\begin{aligned}
I &= \sum_{h_t} p(x_{\leq t}, h_{\leq t}) \log \frac{p(h_t|x_{\leq t}, h_{<t})}{p(h_t)} \\
&= \sum_{h_t} p(x_{\leq t}, h_{\leq t}) \log \frac{\prod_i p(h_{t_i}|x_{\leq t}, h_{<t})}{\prod_i p(h_{t,i})} - \sum_{h_t} p(x_{\leq t}, h_{\leq t}) \log \frac{p(h_t)}{\prod_i p(h_{t,i})} \\
&= \sum_i I(h_{t,i}|x_{\leq t}, h_{<t}) - KL(p(h_t)||\prod_i p(h_{t,i})) \\
&\leq \sum_i I(h_{t,i}|x_{\leq t}, h_{<t}) \\
&\approx \sum_i \sum_{h_{t_i}} p(h_{t_i}|x_{\leq t}, h_{<t}) \log \frac{p(h_{t_i}|x_{\leq t}, h_{<t})}{r(h_{t,i})} \\
&= \sum_i (\sigma_{t,i} \log \frac{\sigma_{t,i}}{\rho(\sigma_{t,i})} + (1 - \sigma_{t,i}) \log \frac{1 - \sigma_{t,i}}{1 - \rho(\sigma_{t,i})}) \\
&\equiv \tilde{I}
\end{aligned}
$$

The second equation follows from the fact that $h_{t,i}$ are conditionally independent given the history $h_{<t}, x_{\leq t}$. In order to close the variational gap, we must attempt to minimize the KL divergence appearing in the third line, which is the Total Correlation [4]. This suggests that in order to maximize mutual information, we must minimize the total correlation [6]. Other heuristic functions can be substituted in the place of the total correlation, which may lead to local learning rules, but we don't treat it for now.

So the weight update contributed by the mutual information objective is given by

$$
\begin{aligned}
\Delta W_{t,ij} &\propto \frac{\partial \tilde{I}}{\partial W_{ij}} \\
&= \sigma'_{t,i} e_{t,ij} (\log \frac{\sigma_{t,i}}{1 - \sigma_{t,i}} - \log \frac{\rho(\sigma_{t,i})}{1 - \rho(\sigma_{t,i})}) - \rho(\sigma'_{t,i} e_{t,ij}) (\frac{\sigma_{t,i} - \rho(\sigma_{t,i})}{\rho(\sigma_{t,i})(1 - \rho(\sigma_{t,i}))}) \\
&= \sigma'_{t,i} e_{t,ij} (v_{t,i} - \log \frac{\rho(\sigma_{t,i})}{1 - \rho(\sigma_{t,i})}) - \rho(\sigma'_{t,i} e_{t,ij}) (\frac{\sigma_{t,i} - \rho(\sigma_{t,i})}{\rho(\sigma_{t,i})(1 - \rho(\sigma_{t,i}))})
\end{aligned}
$$

Here $\rho(\sigma'_{t,i} e_{t,ij})$ is a running average of the term $\sigma'_{t,i} e_{t,ij}$. The first term is the hebbian product of a pre- and post-synaptic term and a voltage-dependent term, with a threshold dependent on the firing probability history of the network. The second term serves the role of an anti-hebbian term which penalizes firing probabilities that are far from its history average. [9]

References

[1] David Barber and Felix V Agakov. The im algorithm: a variational approach to information maximization. In *Advances in neural information processing systems*, page None, 2003.

[2] Guillaume Bellec, Franz Scherr, Elias Hajek, Darjan Salaj, Robert Legenstein, and Wolfgang Maass. Biologically inspired alternatives to backpropagation through time for learning in recurrent neural nets, 2019.

[3] Frederik Benzing, Marcelo Matheus Gauy, Asier Mujika, Anders Martinsson, and Angelika Steger. Optimal kronecker-sum approximation of real time recurrent learning. *ArXiv*, abs/1902.03993, 2019.

[4] Ricky T. Q. Chen, Xuechen Li, Roger Grosse, and David Duvenaud. Isolating sources of disentanglement in variational autoencoders, 2018.

[5] Matthieu Courbariaux, Itay Hubara, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. Binarized neural networks: Training deep neural networks with weights and activations constrained to +1 or -1, 2016.

[6] Irina Higgins, David Amos, David Pfau, Sebastien Racaniere, Loic Matthey, Danilo Rezende, and Alexander Lerchner. Towards a definition of disentangled representations, 2018.

[7] Eugene M. Izhikevich. Solving the Distal Reward Problem through Linkage of STDP and Dopamine Signaling. *Cerebral Cortex*, 17(10):2443–2452, 01 2007.

[8] Eric Jang, Shixiang Gu, and Ben Poole. Categorical Reparameterization with Gumbel-Softmax. *arXiv e-prints*, page arXiv:1611.01144, Nov 2016.

[9] Sensen Liu and ShiNung Ching. Recurrent information optimization with local, metaplastic synaptic dynamics. *Neural Computation*, 29:2528–2552, 2017.

[10] Owen Marschall, Kyunghyun Cho, and Cristina Savin. A unified framework of online learning algorithms for training recurrent neural networks. *ArXiv*, abs/1907.02649, 2019.

[11] James M Murray. Local online learning in recurrent networks with random feedback. *eLife*, 8:e43299, 2019.

[12] Emre Neftci, Hesham Mostafa, and Friedemann Zenke. Surrogate gradient learning in spiking neural networks. *ArXiv*, abs/1901.09948, 2019.

[13] Ben Poole, Sherjil Ozair, Aäron van den Oord, Alexander A Alemi, and George Tucker. On variational lower bounds of mutual information. In *NeurIPS Workshop on Bayesian Deep Learning*, 2018.

[14] Corentin Tallec and Yann Ollivier. Unbiased online recurrent optimization. *ArXiv*, abs/1702.05043, 2017.

[15] Paul J Werbos et al. Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, 78(10):1550–1560, 1990.

[16] Ronald J Williams and David Zipser. A learning algorithm for continually running fully recurrent neural networks. *Neural computation*, 1(2):270–280, 1989.