Probabilistic Inference and Learning in a Spiking Neural Network

Pavel Sountsov, Paul Miller

sl157@brandeis.edu, pmiller@brandeis.edu

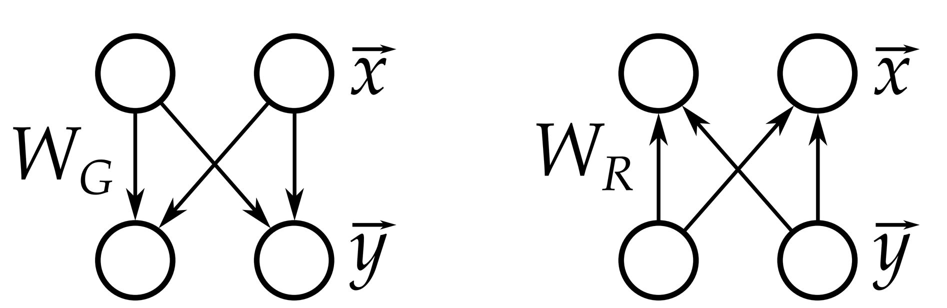
Brandeis University

1. Introduction

Performing perceptual inference requires inverting a learned generative model of the world. This inversion is often intractable, so a separate recognition model is often used to approximate the exact posterior distribution. This project concerns itself with the biological implementation of this approximate inference as well as the mechanisms responsible for learning of the parameters of the recognition model.

The Helmholtz machine (Hinton et al., 1995) shows how such a recognition model can be constructed and trained using local computations. The aim of this project is to implement the Helmholtz machine in a network of model conductance based spiking neurons and experimentally supported synaptic plasticity rules.

2. Computational model



$$P(\vec{x}) = \mathcal{U}(\vec{x};\, 10,\, 50)$$
$$P(\vec{y}|\vec{x}) = \mathcal{N}(\vec{y};\, W_G\vec{x},\, \sigma^2 I)$$
$$Q(\vec{x};\vec{y}) = \mathcal{N}(\vec{x};\, W_R\vec{y},\, \sigma^2 I)$$

3. Learning using Wake-Sleep algorithm

Wake-Sleep algorithm updates W_G and W_R in two distinct phases in an unsupervised fashion, approximating the classical EM algorithm.

- **Wake:** $\vec{y}$ is set to the sensory (training) data, and the recognition model is used to infer top layer activities $\vec{x}$. The generative model is then used to reconstruct the training data given the inferred activity in the top layer. The error in this reconstruction is used to train the generative model weights W_G.
- **Sleep:** The top layer activities are set to a sample from the prior distribution $P(\vec{x})$, and the generative model is used to generate fantasy sensory activity in the bottom layer. The recognition network is then used to infer the activities in the top layer given this fantasy data. The error in this inference is used to train the recognition model weights W_R.

The gaussian activation functions imply a local learning rule identical to the classical perceptron learning rule, the delta rule, for both phases of learning:

$$\Delta w = \epsilon \mathrm{r}_i(\mathrm{r}_\tau - \mathrm{r}_o)$$

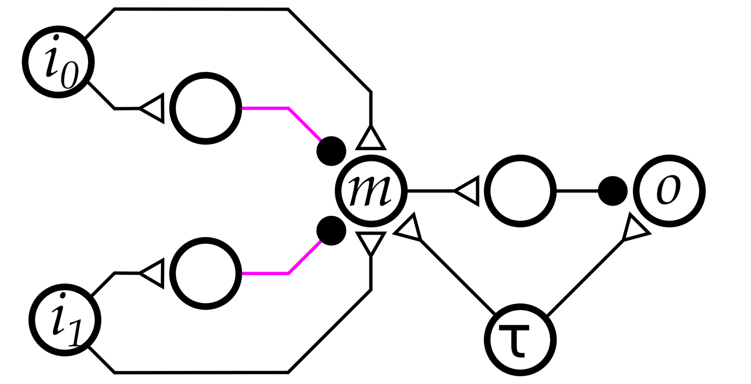
References

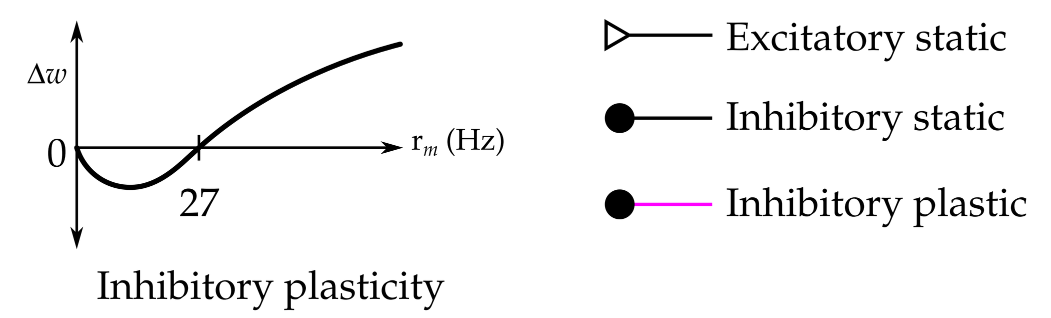
Haas, J. S., Nowotny, T., & Abarbanel, H. D. I. (2006). Spike-timing-dependent plasticity of inhibitory synapses in the entorhinal cortex. *Journal of neurophysiology*, 96(6), 3305-13.

Hinton, G., Dayan, P., Frey, B., & Neal, R. (1995). The "wake-sleep" algorithm for unsupervised neural networks. *Science*, 268(5214), 1158-1161.

4. Delta rule network

Previous efforts to model the delta rule required the use of discrete neuronal firing rates or relied on a special timing relationship between the target and output neuronal activities. The network below improves on those efforts and implements the delta rule with simultaneous, continuously varying firing rates.

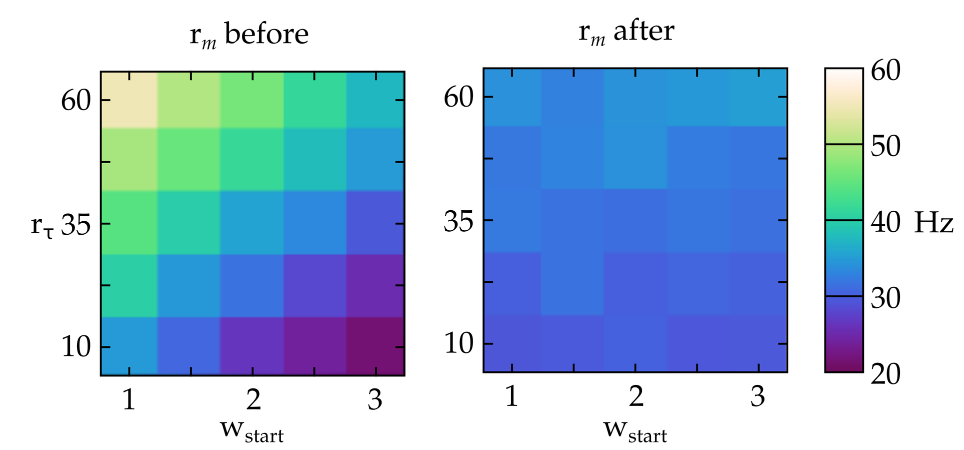




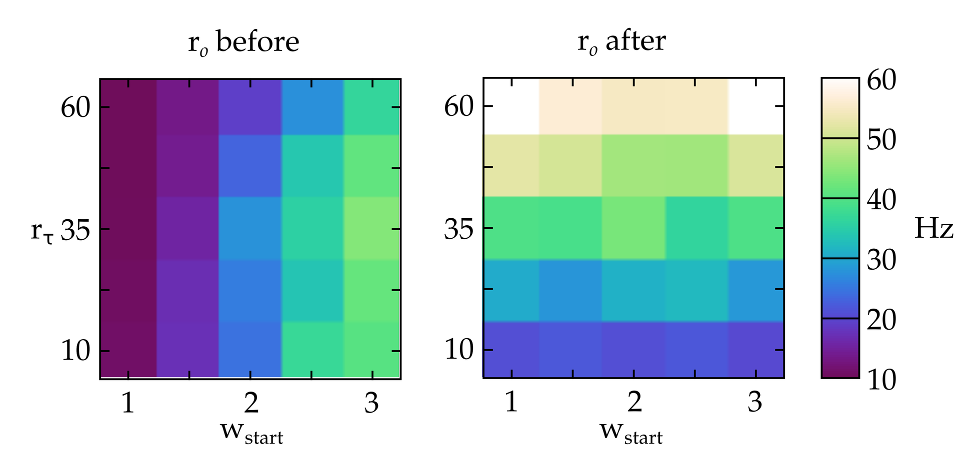
The network is composed of interconnected pools of 10-20 spiking neurons each. Plastic inhibitory connections follow an anti-Hebbian synaptic plasticity rule similar to one characterized by Haas et al. (2006) in the rat entorhinal cortex.

5. Delta network results

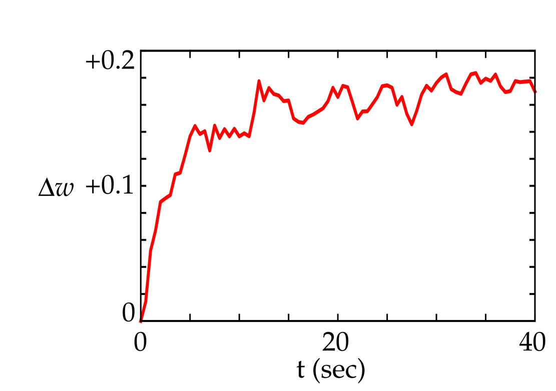
25 separate networks were trained with a steady input, but different initial plastic inhibitory weights and target rates. r_m approaches the rate of ≈27 Hz in all networks after 40 seconds of stimulus presentation:



Initially the r_o does not depend on r_τ, but after training it matches it closely (given steady $\vec{\mathrm{r}}_i$):

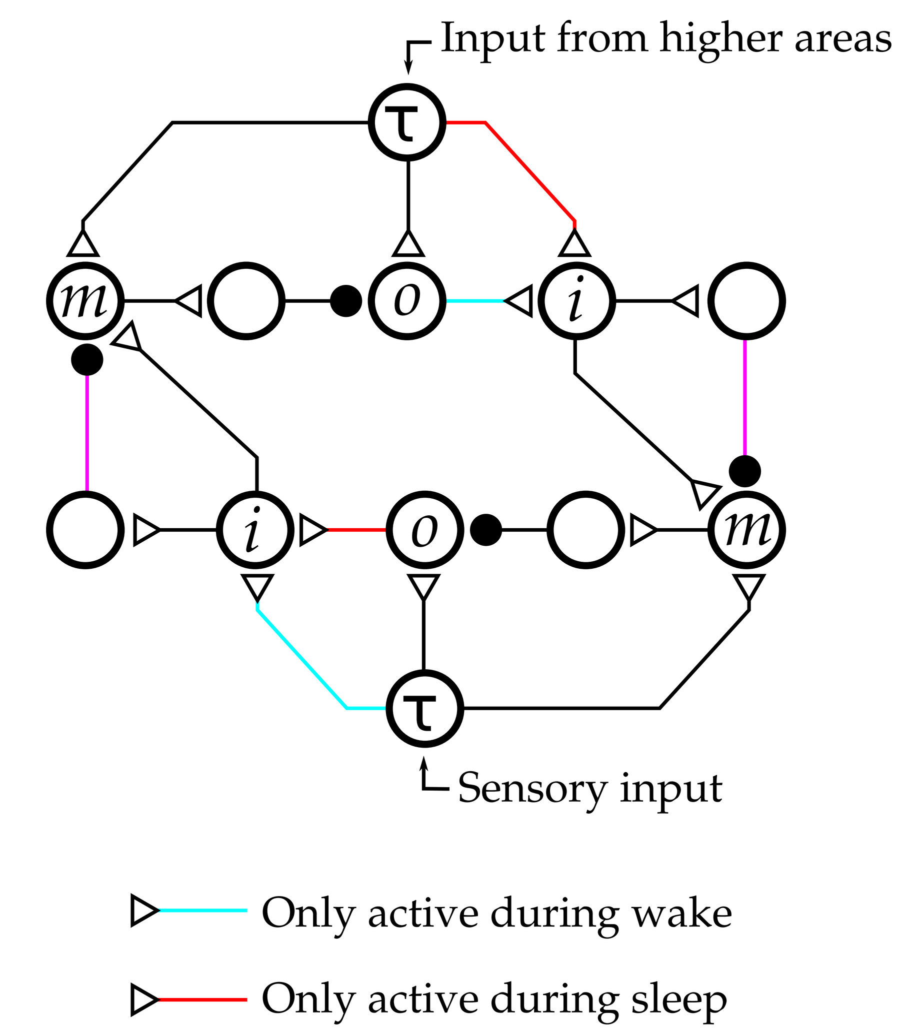


Weights converge to a steady state after ≈20 seconds of stimulus presentation:



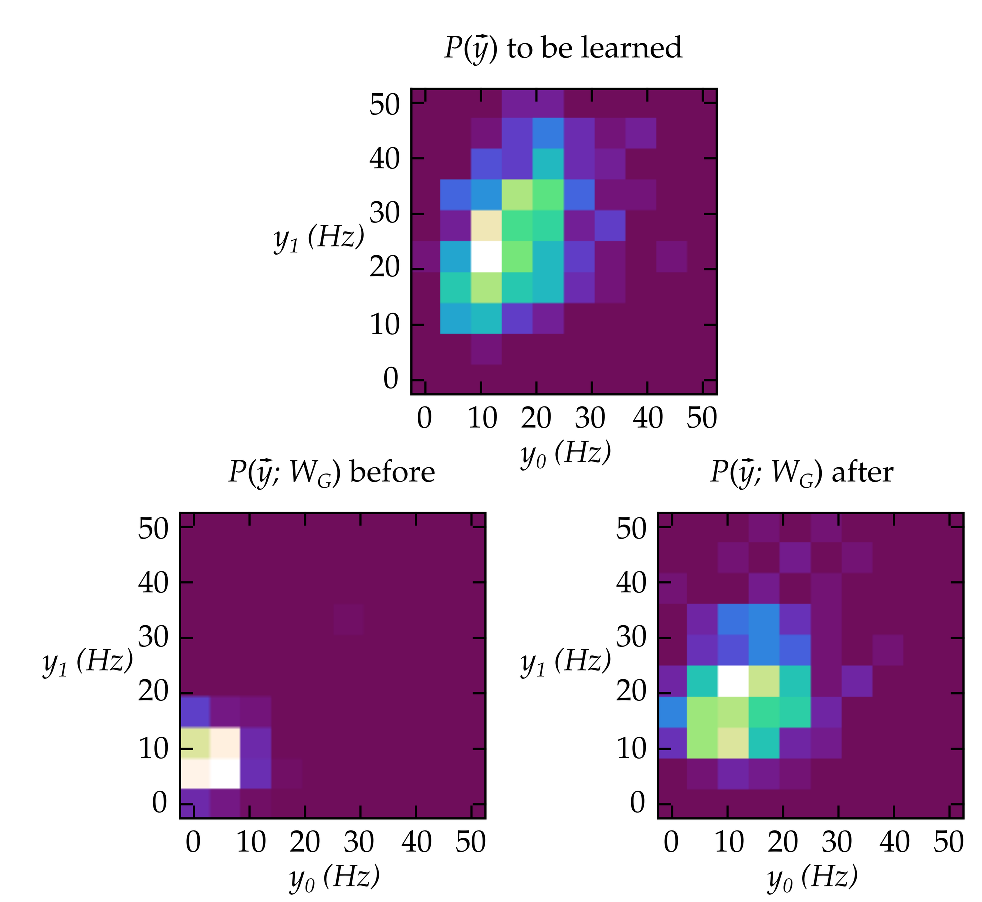
6. Helmholtz machine neural network

The full Helmholtz machine can be built by linking multiple delta rule networks (each one functioning as a stochastic unit) together. Certain connections need to be inactivated during the different phases of the Wake-Sleep algorithm through either a neuro-modulatory signal, or silencing of the pre-synaptic neurons through inhibition. Plasticity needs to be controlled in a similar fashion.

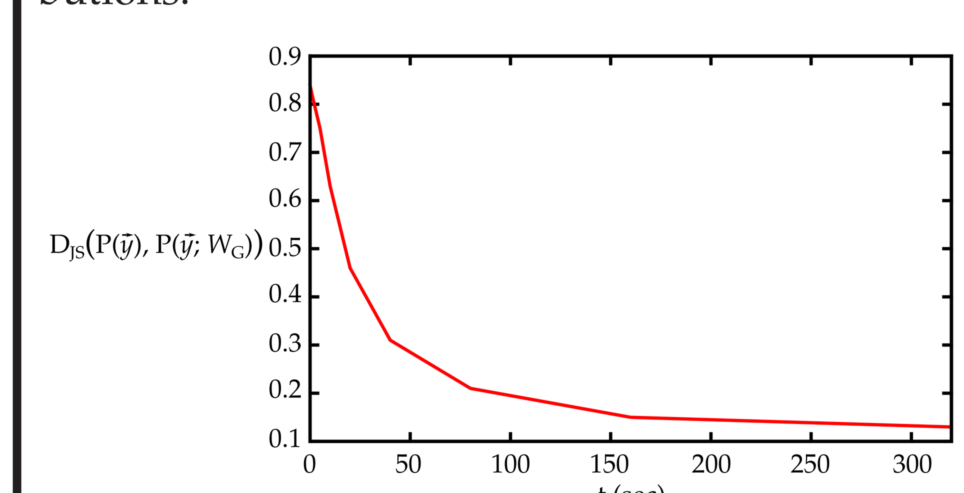


7. Neural network results

The neural network can model simple distributions well. Each phase of the Wake-Sleep algorithm lasts 100 ms.

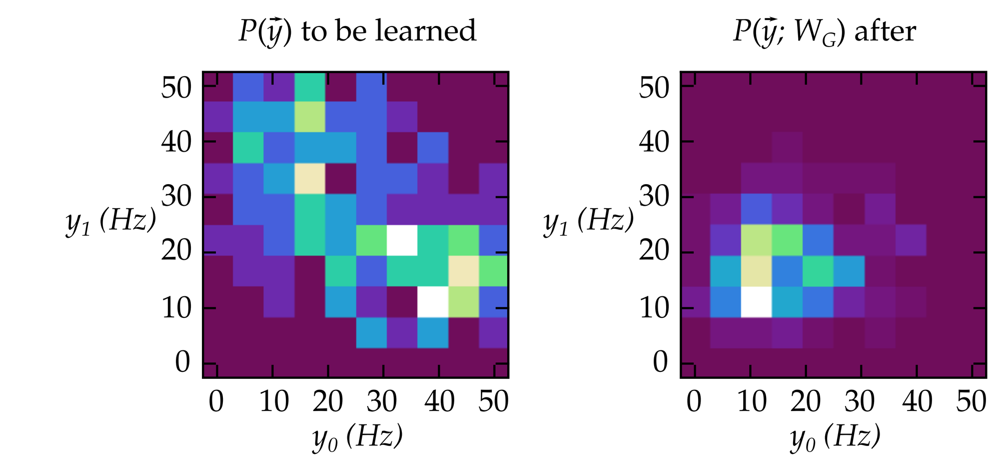


Jensen-Shannon divergence between the data and learned distributions.



8. Bimodal distribution peformance

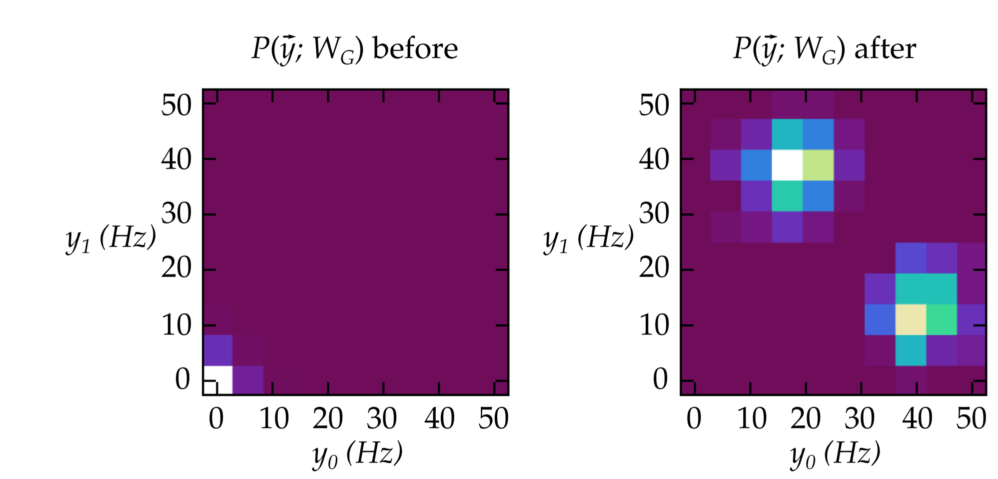
The unmodified neural model has trouble with complicated distributions such as bimodal data distributions, able to only match the mean. Same behavior occurs with the computational implementation (not shown).



9. Modified computational model results

Better performance can be achieved if the model generative model is made to use the correct prior and the recognition model is modified to be closer to the form of the true posterior distribution. When this is done in the computational implementation, it manages to discover the two modes. The corresponding modification neural network implementation is in progress.

$$P(\vec{x}) = \frac{1}{2}\mathcal{N}(\vec{x};\, \vec{\mu}_1,\, \zeta^2 I) + \frac{1}{2}\mathcal{N}(\vec{x};\, \vec{\mu}_2,\, \zeta^2 I)$$
$$P(\vec{y}|\vec{x}) = \mathcal{N}(\vec{y};\, W_G\vec{x},\, \sigma^2 I)$$
$$Q(\vec{x};\vec{y}) = \mathcal{N}(\vec{x};\, W_R\vec{y},\, \sigma^2 I)P(\vec{x})$$



10. Conclusions and future directions

1. The delta rule network seems very capable at implementing the computation in spite of the non-ideal behavior of the plasticity rule and spiking neurons.
2. The neural network implementation of the Helmholtz machine can model simple distributions, but fails with more complicated distributions (e.g. bimodal). In all cases it can only match the means of the activity distributions of individual data units.
3. The computational model with an enhanced generative and recognition model is able to model bimodal distributions, going beyond pure mean-matching.
4. Future work will implement the enhanced recognition model in the neural network through the use of winner-take-all dynamics in the top layer.

Acknowledgements

Thanks to József Fiser for suggesting to look into the Wake-Sleep algorithm and his continued assistance.

Funded by NIH CNTG T90 DA032432 and IGERT 0549390.