Learning multisensory integration with stochastic variational learning in recurrent spiking networks

Presented by : Sharbatanu Chatterjee[1,2]
Guided by : Aditya Gilra[1] & Johanni Brea[1]

[1]Laboratory of Computational Neuroscience, EPFL, Switzerland

[2]Department of Computer Science & Engineering, IIT Kanpur, India

22 Aaugust 2015

Contents

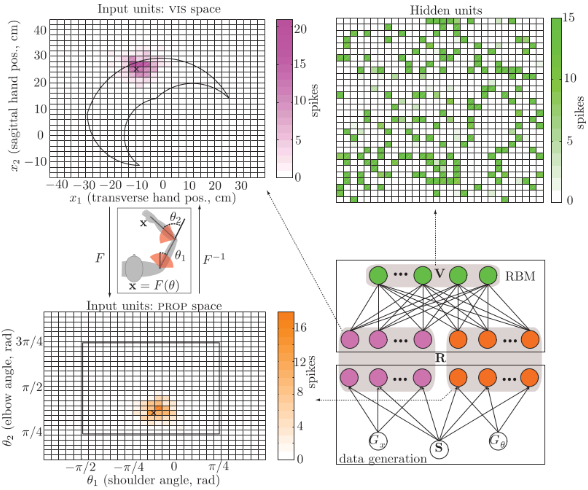

Figure: Architecture for Multisensory Integration[1]

[1] *Sabes et al 2013*

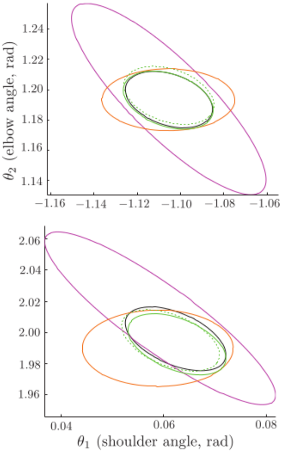

Figure: Regenerating model from learnt hidden neurons[2]

[1] *Sabes et al 2013*

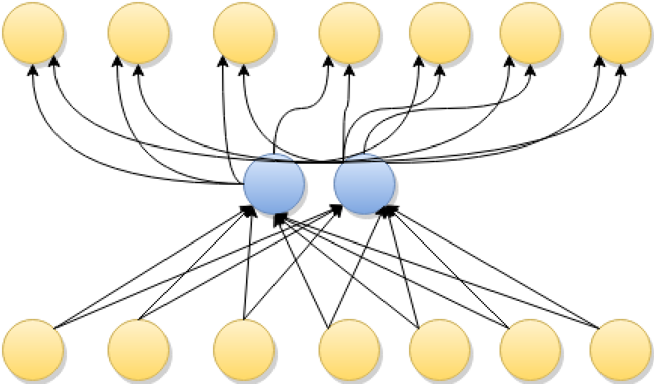

Figure: An autoencoder

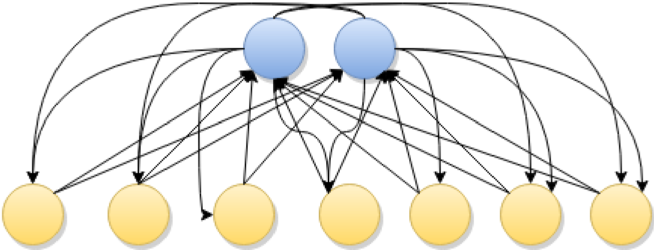

Figure: A schematic for our model

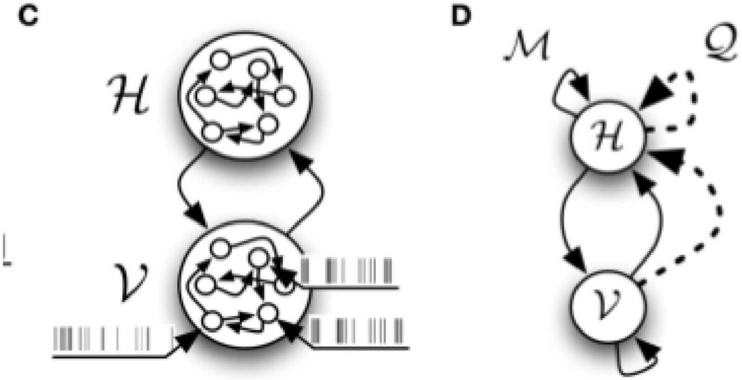

Figure: Architecture for Danilo's model[3]

[2] *Danilo et al 2014*

Danilo's model

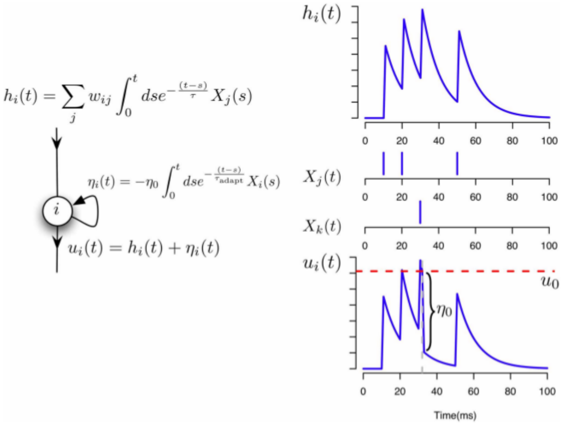

Figure: The Neuron Model[4]

[3] *Danilo et al 2014*

The log-likelihood of seeing the spikes

The probability of producing a spike train $X_i(t)$ is:
The log-likelihood is:

$$logp\left(X\left(0\dots t\right)\right) = \sum_{i \in V \cup H} \int_0^T \left[log \rho_i(\tau) X_i(\tau) - \rho_i(\tau)\right] \quad (1)$$

The marginalized log-likelihood of the visible neurons

$$\boxed{p\left(X_V\right) = \int DX_H p\left(X_V, X_H\right)} \quad (2)$$

A ML approach to the problem

We use another distribution “q” to approximate the posterior

$$
\begin{aligned}
KL(q|p) &= \int DX_H q(X_H|X_V) log \frac{q(X_H|X_V)}{p(X_H|X_V)} \\
&= \boxed{\int DX_H q(X_H|X_V) log \frac{q(X_H|X_V)}{p(X_H, X_V)}} + logp(X_V)
\end{aligned}
\tag{3}
$$

The first term is the *Helmholtz Free Energy* F

We use another distribution "q" to approximate the posterior

$$
\begin{aligned}
0 \leq KL(q|p) &= \int DX_H q(X_H|X_V) log \frac{q(X_H|X_V)}{p(X_H|X_V)} \\
&= \boxed{F} + logp(X_V)
\end{aligned}
\tag{4}
$$

The first term is the *Helmholtz Free Energy* F

We use another distribution "q" to approximate the posterior

$$0 \leq KL(q|p) = \int DX_H q(X_H|X_V) log \frac{q(X_H|X_V)}{p(X_H|X_V)} \\ = \boxed{F} + logp(X_V) \tag{4}$$

The first term is the *Helmholtz Free Energy* F

Since KL divergence is positive

$$F + logp(X_V) \geq 0 \tag{5}$$

We use another distribution "q" to approximate the posterior

$$0 \leq KL(q|p) = \int DX_H q(X_H|X_V) log \frac{q(X_H|X_V)}{p(X_H|X_V)} \tag{4}$$
$$= \boxed{F} + logp(X_V)$$

The first term is the *Helmholtz Free Energy* F

Since KL divergence is positive

$$F + logp(X_V) \geq 0 \tag{5}$$

$$\boxed{logp(X_V) \geq -F} \tag{6}$$

We use another distribution "q" to approximate the posterior

$$0 \leq KL(q|p) = \int DX_H q(X_H|X_V) log \frac{q(X_H|X_V)}{p(X_H|X_V)} \\ = \boxed{F} + logp(X_V) \quad (4)$$

The first term is the *Helmholtz Free Energy* F

Since KL divergence is positive

$$F + logp(X_V) \geq 0 \quad (5)$$

$$\boxed{logp(X_V) \geq -F} \quad (6)$$

Problem reduced to minimising the Free Energy with respect to q and the original model p

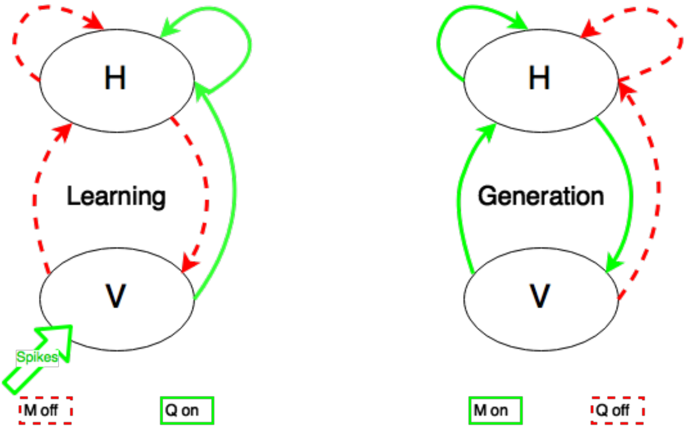

Figure: The Neuron Model

The final weight updates are simple gradient descent on the free energy

$$\dot{w}_{ij}^{M} = -\mu^{M}\nabla_{w_{ij}^{M}}F \tag{7}$$

$$\dot{w}_{ij}^{Q} = -\mu^{Q}\nabla_{w_{ij}^{Q}}F \tag{8}$$

Final Equations

$$\dot{w}_{ij}^{M} = \mu^{M} H_{ij}^{M}(t) \tag{9}$$

$$\dot{w}_{ij}^{Q} = -\mu^{Q} e_{N}(t) H_{ij}^{Q}(t) \tag{10}$$

Final Equations

$$\dot{w}_{ij}^{M} = \mu^{M} H_{ij}^{M}(t) \tag{9}$$

$$\dot{w}_{ij}^{Q} = -\mu^{Q} e_N(t) H_{ij}^{Q}(t) \tag{10}$$

$$(X_i - \rho_i^{Q/M}) * \phi_j \tag{11}$$

Final Equations

$$\dot{w}_{ij}^{M} = \mu^{M} H_{ij}^{M}(t) \quad (9)$$

$$\dot{w}_{ij}^{Q} = -\mu^{Q} e_N(t) H_{ij}^{Q}(t) \quad (10)$$

$$(X_i - \rho_i^{Q/M}) * \phi_j \quad (11)$$

$$e_N(t) = \hat{F} - \bar{F} \quad (12)$$

Global signal

$$e_N(t) = \hat{F} - \bar{F} \tag{13}$$

Global signal

$$e_N(t) = \hat{F} - \bar{F} \tag{13}$$

$$\hat{F} = \int F_\tau d\tau \tag{14}$$

$$F_\tau = F_Q - F_M \tag{15}$$

$$F_Q = \sum_{i \in H} \left[log \rho_i^Q(\tau) X_i(\tau) - \rho_i^Q(\tau) \right] \tag{16}$$

$$F_M = \sum_{i \in V \cup H} \left[log \rho_i^M(\tau) X_i(\tau) - \rho_i^M(\tau) \right] \tag{17}$$

Results

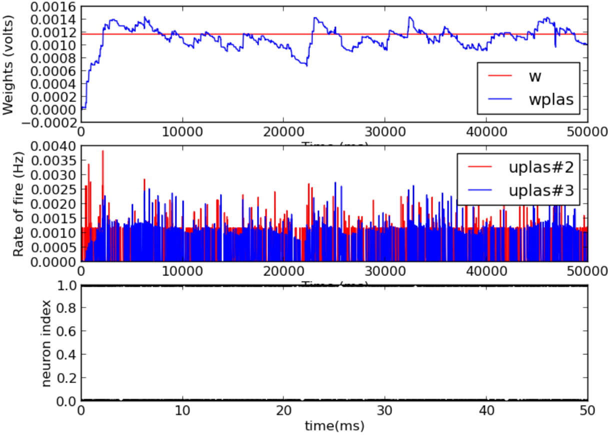

Figure: 2 neurons, only M network, no global factor

Results

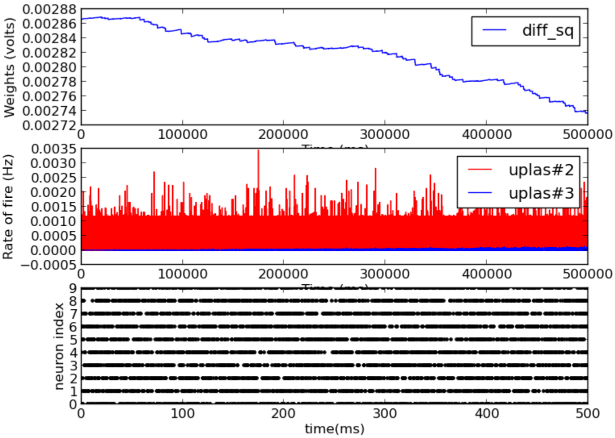

Figure: 10 neurons, same as above[5]

Results

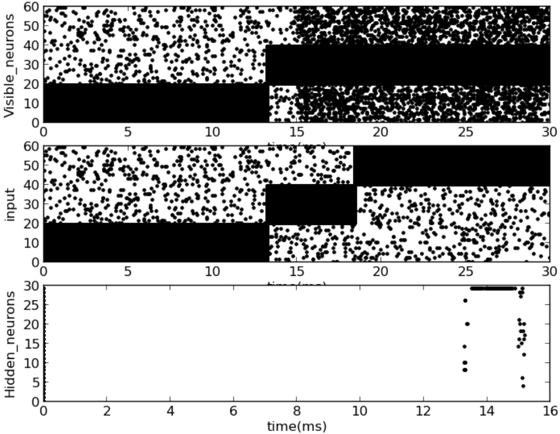

Figure: Entire network, malfunctional hidden neurons, learning stops at halfway

Results

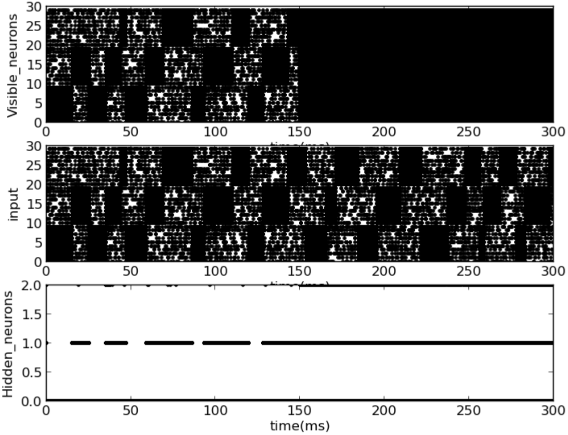

Figure: The entire required result, learning stops halfway

Results

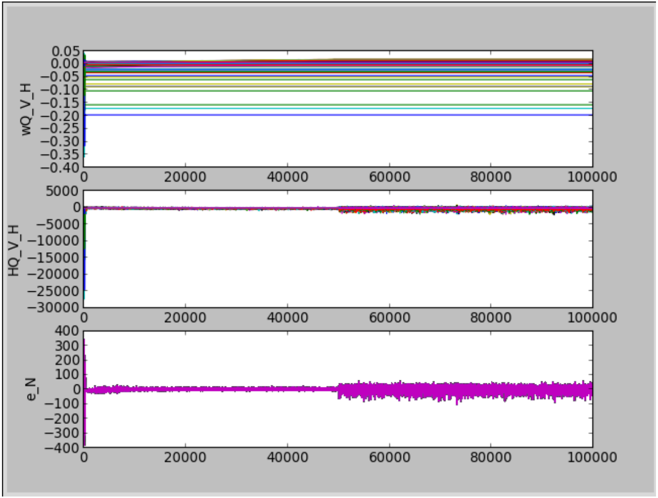

Figure: The error terms with constant firing

Results

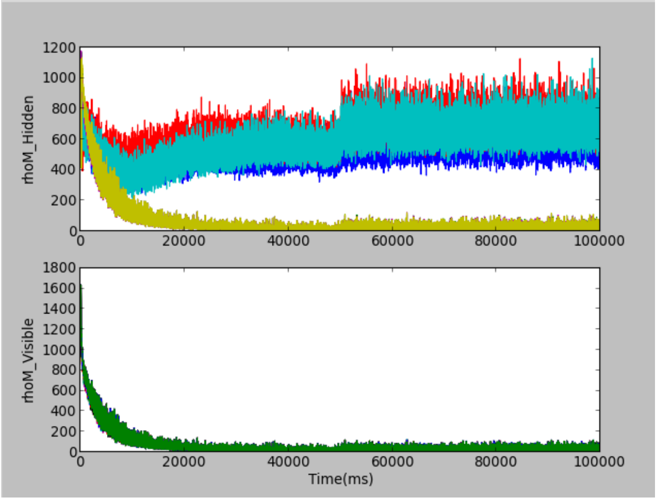

Figure: The rho being approached with constant firing

Further work and implication

- Get my implementation of Danilo's model to function flawlessly.
- Sabes' paper does not introduce a temporal factor, try to incorporate so.
- Encoding for other key processes of sensory processing - integration of prior information and coordinate transforms.

Thank You!