Fast inference with spiking networks

Mihai A. Petrovici

Computantional Neuroscience Group
Department of Physiology
University of Bern

Electronic Vision(s)
Kirchhoff Institute for Physics
University of Heidelberg
Probabilistic computation with spikes
Probabilistic (Bayesian) computing: motivation

get an estimate of the distribution of the sought function (expected value and uncertainty)
Probabilistic (Bayesian) computing: experimental evidence

Rabbit/duck ambiguity

Knill-Kersten illusion

Necker cube

reflectance  $z_2$: 3D shape

$z_3$: shading  $z_4$: contour
Probabilistic (Bayesian) computing: experimental evidence

**Starkweather et al. (2017)**

- **Task 1**: 100% rewarded
- **Odor ON**: 1.2 - 2.8 s
- **Odor A reward**: 100

**Task 2**: 90% rewarded

**Firing rate (Hz)**

- **Time (s)**: 0, 1, 2, 3

**Task 1**

**Latest reward**

**Task 2**

**Earliest reward**

**Firing rate (Hz)**

- **Time (s)**: 0, 1, 2, 3

- **100% rewarded**
- **90% rewarded**

**Berkes et al. (2011)**

- **Visual stimulation**
- **Decreasing contrast**
- **No stimulus**

- **Posterior (EA)**
- **Prior**

- **Multiunit recording**

- **Posterior = Prior (SA)**

**Divergence (KL, bits/sec)**

- **Postnatal age (days)**

- **29-30**
- **34-45**
- **83-92**
- **129-151**
Probabilistic (Bayesian) computing in machine learning

Deep generative models
Salakhutdinov & Hinton (2009)

Neuromorphic hardware
Schemmel et al. (2010)
A system that performs probabilistic inference has to

→ represent probability distributions \( p(z_1, z_2, \ldots) \)

→ calculate posterior (conditional) distributions \( p(z_1, z_2, \ldots | z_k, z_{k+1}, \ldots) \)

→ evaluate marginal distributions \( p(z_1, z_2) = \sum_{z_3, z_4, \ldots} p(z_1, z_2, z_3, \ldots) \)
Representation of probability distributions

\[ p(z) = \frac{1}{Z} \exp \left[ \frac{1}{2} z^T W z + z^T b \right] \]
Sampling vs. parametric representation

temporal aspects:
- increasingly correct representation
- anytime computing

computational complexity aspects:
- computation of conditionals is simple
- marginalization is free
Spike-based encoding of an ensemble state

\[ z_k = 1 \iff \text{neuron has spiked in } [t - \tau, t) \]

\[ \rightarrow \text{spike pattern encodes states } z^{(t)} \]
Emulation of Boltzmann machines

\[ z_k = 1 \iff \text{neuron has spiked in } [t - \tau, t) \]

\[ \rightarrow \text{spike pattern encodes states } z^{(t)} \]

**Boltzmann distribution over** \( z_k \in \{0, 1\} \)

\[ p(z) = \frac{1}{Z} \exp \left[ \frac{1}{2} z^T W z + z^T b \right] \]

**Neural computability condition**

\[ u_k = \log \frac{p(z_k = 1 \mid z_{\setminus k})}{p(z_k = 0 \mid z_{\setminus k})} \]

(which is equivalent to a logistic activation function \( p(z_k = 1 \mid z_{\setminus k}) = \frac{1}{1 + \exp(-u_k)} \)).

**mediated by synaptic weights:**

\[ u_k = \sum_{i=1}^{K} W_{ki} z_i + b_k \]
Emulation of Boltzmann machines

Neural computability condition

\[ u_k = \log \frac{p(z_k = 1 \mid z_{\setminus k})}{p(z_k = 0 \mid z_{\setminus k})} \]

(which is equivalent to a logistic activation function \( p(z_k = 1 \mid z_{\setminus k}) = \frac{1}{1 + \exp(-u_k)} \)).
$$p_{\text{spike}} \approx \text{erf}[\alpha \cdot (u_{\text{eff}} - \langle u_0 \rangle)]$$
$$\approx \sigma[\alpha \cdot (u_{\text{eff}} - \langle u_0 \rangle)]$$

Idea: Stochasticity by Poisson background

Unfortunately, neurons are a bit more complicated...
The diffusion approximation

noise source: Poisson spike trains  
high background firing rates  
relatively low synaptic weights  
\[ \Rightarrow \text{membrane as Ornstein-Uhlenbeck process} \]
\[ du(t) = \Theta \cdot [\mu - u(t)]dt + \sigma dW(t) \]

Ricciardi & Sacerdote (1979)

for COBA LIF:

\[ \Theta = \frac{1}{\tau_{syn}} \]
\[ \mu = \frac{I^\text{ext} + g_l E_l + \sum_{i=1}^{n} v_i w_i E_i^{\text{rev}} \tau_{syn}}{\langle g^{\text{tot}} \rangle} \]
\[ \sigma^2 = \frac{\sum_{i=1}^{n} v_i \left( w_i (E_i^{\text{rev}} - \mu) \right)^2 \tau_{syn}}{2 \langle g^{\text{tot}} \rangle^2} \]
First-passage-time calculations

Brunel & Sergi (1998)

assumption: $\tau_{\text{syn}} \ll \tau_m$

this allows expansion in $\sqrt{\frac{\tau_{\text{syn}}}{\tau_m}}$:

$$\langle T \rangle = \tau \sqrt{\pi} \int_{-\rho+\mu/\sigma}^{\phi_{\text{eff}}-\mu/\sigma} [1 + \text{erf}(x)] \exp(x^2) \, dx$$

Brunel, Sergi (1998)

$$V_k = \frac{1}{\tau_{\text{ref}} + \text{FPT}(V_{th}, V_0)}$$

Thomas (1975)

$$\text{FPT}(a, 0) := \langle \inf\{t \geq 0 : V_t = a | V_0 = 0 \} \rangle$$

$$= \sqrt{\frac{\pi}{\rho \sigma^2}} \int_0^{\frac{a}{\sqrt{\rho \sigma^2}}} \left[ 1 + \text{erf}(\frac{\rho x}{\sigma}) \right] \exp\left(\frac{\rho x^2}{\sigma^2}\right) \, dx$$

however, remember that $\tau_{\text{ref}} \approx \tau_{\text{syn}}$!

$\Rightarrow$ membrane does not forget!

Brunel, Sergi (1998)

Moreno-Bote, Parga (2004)

Simulation results

Equation (2) without $t^*$

Equation (2)
Moreno-Bote & Parga (2004)

**assumption:** $\tau_{\text{syn}} \gg \tau_m$

then, the synaptic input appears quasistatic to the membrane

$$
\tilde{v} = \left( \tau_m \frac{\vartheta - \tilde{u}}{\vartheta - \tilde{u}} \right)^{-1}
$$

$$
\nu = \int_{\vartheta}^{\infty} \tilde{v}(\tilde{u}) p(\tilde{u}) \, d\tilde{u}
$$

however, remember that $\tau_{\text{ref}} \approx \tau_{\text{syn}}$

$\Rightarrow$ adiabatic approximation does not hold!
The membrane autocorrelation propagation

\[ p(z_k = 1) = \frac{t_{k, \text{refractory}}}{t_{\text{total}}} = \frac{\sum_n P_n n \tau_{\text{ref}}}{\sum_n P_n \left( n \tau_{\text{ref}} + \sum_{k=1}^{n-1} \overline{\tau}_k^b + T_n \right)} \]

\[ P_n = \left(1 - \sum_{i=1}^{n-1} P_i \right) \int_{V_{\text{thr}}}^{\infty} dV_{n-1} \ p(V_{n-1} | V_{n-1} > V_{\text{thr}}) \left[ \int_{V_{\text{thr}}}^{\infty} dV_n \ p(V_n | V_{n-1}) (FPT(V_{\text{thr}}, V_n)) \right] \]

\[ T_n = \int_{V_{\text{thr}}}^{\infty} dV_{n-1} \ p(V_{n-1} | V_{n-1} > V_{\text{thr}}) \left[ \int_{-\infty}^{V_{\text{thr}}} dV_n \ p(V_n | V_{n-1}) (FPT(V_{\text{thr}}, V_n)) \right] \]

\[ \overline{\tau}_k^b = \int_{\theta}^{\infty} du_k \ \tau_{\text{eff}} \ln \left( \frac{\theta - u_k}{\theta - u_{k-1}} \right) p(u_k | u_k > \theta, u_{k-1}) \]

A

B

---

**HCS!**

---

Petrovici & Bill et al. (2016)
(Fully visible) LIF-based Boltzmann machines
Beyond Boltzmann: Spiking Bayesian networks
Deep spiking discriminative architectures

Training of RBMs/DBMs on MNIST:
- maximum likelihood learning
  \[ \Delta w_{ij} \propto \langle z_i z_j \rangle_{\text{data}} - \langle z_i z_j \rangle_{\text{model}} \]
  \[ \Delta b_i \propto \langle z_i \rangle_{\text{data}} - \langle z_i \rangle_{\text{model}} \]
- coupled adaptive tempering

96.9 % correct classification with less than 2000 neurons

Leng & Petrovici et al. (2016)
Deep pong

Roth, Zenk (2017)
Training of RBMs/DBMs on MNIST:
- maximum likelihood learning
  \[ \Delta w_{ij} \propto \langle z_i z_j \rangle_{\text{data}} - \langle z_i z_j \rangle_{\text{model}} \]
  \[ \Delta b_i \propto \langle z_i \rangle_{\text{data}} - \langle z_i \rangle_{\text{model}} \]
- coupled adaptive tempering

Deep spiking generative architectures

96.9% correct classification with less than 2000 neurons

Leng & Petrovici et al. (2016)
Short-term plasticity enables superior mixing

STP model: Tsodyks & Markram (1997)

Leng & Petrovici et al. (2016)
... so where does the noise come from?

1st approximation: independent Poisson sources

unrealistic in both biological & artificial systems
better: common pool of presynaptic partners
⇒ correlated inputs
⇒ deviation from target distribution
Embedded stochastic inference machines

more realistic: sea of noise

\[ C_{kl}^{\text{in}} = C_{\text{shared,kl}}^{\text{in}} + C_{\text{corr,kl}}^{\text{in}} \]

\[ C_{\text{corr}}^{\text{in}} = \begin{cases} 0 & \text{if } C_{\text{corr}}^{\text{in}} = 0 \\ < 0 & \text{if } C_{\text{corr}}^{\text{in}} < 0 \end{cases} \]

Sampling duration $T$ (ms)

Network state $s$
Noiseless stochastic computation

ongoing work with Dominik Dold and Ilja Bytschok
Physical emulation of spiking networks
Microprocessor Transistor Counts 1971-2011 & Moore’s Law

simulation speed 1520:1
compared to biological real-time

Driesmann (2012)

<table>
<thead>
<tr>
<th>nature</th>
<th>simulation</th>
</tr>
</thead>
<tbody>
<tr>
<td>synaptic plasticity</td>
<td>seconds</td>
</tr>
<tr>
<td>learning</td>
<td>days</td>
</tr>
<tr>
<td>development</td>
<td>years</td>
</tr>
<tr>
<td>evolution</td>
<td>&gt; millennia</td>
</tr>
</tbody>
</table>
Simulation & emulation: energy scaling

Energy Scales:

- $10^6$ J
  - 1 Joule

- $10^{-4}$ J
  - 0.1 milliJoule

- $10^{-8}$ J
  - 10 nanoJoule

- $10^{-14}$ J
  - 10 femtoJoule

1,000,000 times more energy-efficient

10,000 times less energy-efficient
Analog neuromorphic hardware

Electrophysiology

Hodgkin-Huxley-Model

\[ C_m \ddot{u} = g_L(u - E_L) + g_{syn}(u - E_{syn}) + g_K n^4(u - E_K) + g_{Na} m^3 h(u - E_{Na}) \]

Adaptive Exponential I&F Model

\[ C_m \ddot{u} = g_L(u - E_L) + g_{syn}(u - E_{syn}) + g_L A_T \exp\left(\frac{V - V_T}{\Delta_T}\right) - w \]
\[ \tau_w \dot{w} = a(V - E_L) - w \]

Schemmel et al. (2010)
Mixed-signal VLSI:
membrane → analog
spikes → digital

Inherent speedup: $10^3 - 10^5$

Adaptive Exponential I&F Model

\[
C_m \ddot{u} = g_L (u - E_L) + g_{syn} (u - E_{syn}) + g_L \Delta T \exp \left( \frac{V - V_T}{\Delta T} \right) - w
\]

\[
\tau_w \dot{w} = a (V - E_L) - w
\]

Schemmel et al. (2010)
Waferscale integration → BrainScaleS system

Schemmel et al. (2010)
The Hybrid Modeling Facility in Heidelberg

4 million AdEx neurons, 1 billion conductance-based synapses, under construction
Hardware is not software...
LIF sampling on accelerated hardware

bio time: 100 s
software simulation: 1 s
hardware emulation: 10 ms
Robustness from structure

A

label (6)

hidden (50)

visible (144)

C

D

E

F

classification rate $R$

$\Delta T_{\text{delay}}/\tau_{\text{ref}}$

$\sigma_{\tau_{\text{ref}}}/\tau_{\text{ref}}$

$\tau_{m} \text{ [ms]}$

weight resolution [bits]

classification rate $R$

ideal SW simulated HW Spikey Spikey runs with single neurons

Petrovici & Schröder et al. (2017)
Outlook / Work in progress
Ensemble dynamics

\[
p(\sigma) = \frac{1}{Z} \exp \left[ \frac{1}{2} \sigma^T J \sigma + \sigma^T b \right]
\]

spiking networks
modeling
magnetic systems

\[
p(z) = \frac{1}{Z} \exp \left[ \frac{1}{2} z^T W z + z^T b \right]
\]

ongoing work with Andreas Baumbach
Quantum many-body problems

\[ \Delta W \propto \frac{\partial}{\partial w} \frac{\langle \Psi_M | H | \Psi_M \rangle}{\langle \Psi_M | \Psi_M \rangle} \]
Learning rules

- **maximum likelihood learning**
  \[ \Delta w_{ij} \propto \langle z_i z_j \rangle_{\text{data}} - \langle z_i z_j \rangle_{\text{model}} \]
  \[ \Delta b_i \propto \langle z_i \rangle_{\text{data}} - \langle z_i \rangle_{\text{model}} \]

- **backprop**
  \[ \Delta w_{ij} \propto \frac{\partial E}{\partial o_j} \frac{\partial o_j}{\partial \text{net}_j} \frac{\partial \text{net}_j}{\partial w_{ij}} \]

\[ \dot{w}_k^a = \eta (\phi(U) - \phi(\alpha V^a)) \text{ PSP}_k \]
References