Building blocks for learning and inference in neuromorphic systems

Dr. Michael Pfeiffer
pfeiffer@ini.phys.ethz.ch
Institute of Neuroinformatics
University of Zurich and ETH Zurich, Switzerland

NICE Workshop, Albuquerque NM, February 26th 2014
Presenting work and ideas by (among others) …

Giacomo Indiveri
Tobi Delbruck
Shih-Chii Liu
Rodney Douglas

Emre Neftci (now UCSD)
Kevan Martin
Peter O’Connor (now Braincorp)
Danny Neil

Ueli Rutishauser (Caltech)
Matthew Cook
Elisabetta Chicca (now Univ. Bielefeld)
Jonathan Binas
Why investigate spike-based computation?

Richard Feynman (1918-1988)

And this is how we create cognitive behavior in neuromorphic systems: a) ...
Building Blocks...
Building Blocks...

Relevant for real-world problems

Scalable and adaptable

Mapping onto neuromorphic hardware

Efficient input representations

Brain-inspired
Building Blocks for Neuromorphic Engineering

Applications
Probabilistic Inference
Sensor Fusion
Learning

Building Blocks in Biology
Sensors

Mapping onto neuromorphic hardware
Building Blocks…

Sensors
Conventional vs. Event-based Sensors

Conventional camera:
- Shoots still images or sequences of frames at constant frame rate
- Same high resolution over the entire image
- Every pixel behaves similarly
- Massive amounts of data
- Mostly redundant data for processing sequences / videos
DVS – The silicon retina
Tobi Delbruck: siliconretina.ini.uzh.ch

DVS128 Sensor

128x128 Pixel DVS Chip

Absolute Intensity

Time

Function of a single pixel

Activity of all pixels
Advantages of DVS

+ High Speed and precise time resolution
+ Low data rate
+ Ideally suited for real-time tracking, robotics, ...
- Low spatial resolution
- No intensity measurement

Lichtsteiner et al. 2006
Silicon Cochlea: Shih-Chii Liu
Challenges of Event-based Sensory Processing

1. Standard computer vision or machine learning approaches do not work on *spatio-temporal spike patterns*
2. Working on *asynchronous* sequences is different than on still images
3. Problem of *sensor fusion*
4. Challenge and opportunity of *real-time scenarios*

**Our goals:**

1. Relate *event-based learning and computation* to established machine learning and inference mechanisms
2. Make event-based algorithms suitable for neuromorphic hardware to run in *real-time and with low energy consumption*
3. Understanding *neural computation* in biology better by creating functional bio-inspired silicon solutions
Application: Gesture Recognition

- Remote-free control of devices with arm and hand gestures
- Collaboration with Jun-Haeng Lee, Hyunsurk Ryu, et al. (SAIT)

![Diagram of gesture recognition with examples](image)
Stereo-DVS-based Gesture Recognition

- **Motion detection** is already performed at sensor level
- **Stereo-DVS** setup reduces noise
- Simple **clustering** of events for tracking
- **Data reduction** for post-processing
- Use data-dependent input rate for **segmentation** of gestures
- HMM-based classification of trajectories
  - ~97% recognition
  - Negligible latency (~20 ms)

[Lee et al. ISCAS 2012, ICIP 2012, IEEE TNNLS 2014]
Building Blocks…
[Douglas and Martin, 1991] suggested **Winner-take-all architecture** for canonical circuits of cat V1 (based on anatomy and electrophysiology)
Building Blocks…
sWTA networks as computational modules

- Inhibitory neurons
- Excitatory neurons

Global Inhibition
Nearest-N Excitation

Linear and non-linear properties:

- Linear behaviors:
  - Analog gain
  - Locus invariance
  - Gain control by common mode input

- Non-linear behaviors:
  - Selective amplification
  - Signal restoration
  - Multi-stability

[Douglas and Martin, 2007]
sWTA networks in neuromorphic VLSI
Soft-state-machines

Single WTA
- Inhibitory neurons
- Excitatory neurons
- Global Inhibition
- Nearest-N Excitation

Coupled WTAs

[Rutishauser Douglas, 2009; Rutishauser et al., 2010; Neftci et al., 2010]
Synthesizing Cognition in VLSI

Real-time context-dependent visual processing on multi-chip neuromorphic system, using neuromorphic vision sensors (Neftci et al. PNAS 2013)
Building Blocks…
Spike-based learning of Bayesian models
(with Bernhard Nessler, Wolfgang Maass; TU Graz)

Graphical model (Bayesian network)

hidden cause

 observable variables

$$p(z_k \text{ fires at time } t | y) = \frac{e^{u_k(t)}}{\sum_{l=1}^{K} e^{u_l(t)}}$$

$$u_k(t) = \sum_{i=1}^{n} w_{ki} \tilde{y}_i(t) + w_{k0}$$

Nessler et al. 2009: STDP enables spiking neurons to detect hidden causes of their inputs. NIPS 2009
Nessler et al. 2013: Bayesian Computation emerges in generic cortical microcircuits through STDP. PLoS CB
Spike-based EM learning

20 random samples from the 70 000 samples in the MNIST dataset.

50ms per digit

weights of one neuron
Learning of generative models with STDP

We can rigorously prove that this STDP curve in this circuit approximates the Expectation-Maximization (EM) algorithm

– Most general and most widely used tool for unsupervised machine learning (clustering, HMM learning, …)
– Spike-based Expectation Maximization (SEM)

Weights converge to conditional log-probabilities:

\[ \log p( \text{presyn. neuron has fired just before time } t \mid \text{postsyn. neuron fires at time } t) \]

\[ w^*_{ki} := \log p_w^*(y_i = 1 \mid z_k = 1) \quad \text{and} \quad w^*_{k0} := \log p_w^*(z_k = 1) \]

A spike-based view of Bayesian computation

– Synapses learn generative models of their inputs
– Output spike is probabilistic sample from posterior distribution
– Building block for learning and inference

Nessler et al. 2013: Bayesian Computation emerges in generic cortical microcircuits through STDP. PLoS CB
Learning of long spatio-temporal patterns

Output neurons learn to fire in characteristic sequence

- A state-machine or HMM-like approach can learn to recognize such sequences [Corneil et al., Cosyne 2014]
Building Blocks...
Building Blocks…

- Sensors
- Building Blocks in Biology
- Graphical Models
- Deep Belief Networks
- WTA
- Probabilistic Inference
- Sensor Fusion
- Learning
Principles of Deep Learning

1. Use Joint Dataset to learn a hierarchy of task-independent features
   - Restricted Boltzmann Machines (RBMs)
   - Deep Belief Network (DBN)

Unlabeled Data

more abstract features

general
low level features

Input Image
Principles of Deep Learning

1. Use Joint Dataset to learn a hierarchy of task-independent features
2. Optimize for specific task

Human Faces

- task specific features
- more abstract features
- general low level features

Input Image
Unsolved Problems of Deep Learning

Google Data Center (2013)
Unsolved Problems of Deep Learning

Proposed Solution: Event-based Deep Belief Networks

- Massively parallel
- Asynchronous
- Sparse updates
- Online learning
- Scalable
Analogies and Advantages of Brain-like computation

Scaling up without slowing down

Hierarchical organization

Massively parallel computation of independent units

Asynchronous, sparse distributed event codes

Figures from: Markov and Kennedy, 2013; Paul DeKoninck Lab; Krüger and Aiple, 1988
### Event-based Deep Belief Networks

#### DVS 128 Vision Sensor

![DVS 128 Vision Sensor Image]

#### Table: Layers of the Model

<table>
<thead>
<tr>
<th>Layer Type</th>
<th>Visual Input Layer (Bottom Up)</th>
<th>Visual Abstraction Layer</th>
<th>Assoc. Layer</th>
<th>Label Layer</th>
<th>Visual Input Layer (Top Down)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Retina Input</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><img src="image" alt="Input Image" /></td>
<td><img src="image" alt="Abstraction Image" /></td>
<td><img src="image" alt="Assoc. Image" /></td>
<td><img src="image" alt="Label Image" /></td>
<td><img src="image" alt="Top Down Image" /></td>
</tr>
</tbody>
</table>

#### Diagram: Architecture

- **Top RBM:**
  - Visual Abstraction Layer (500 Units)
  - Label Layer (10 Units)
- **Bottom RBM:**
  - Visual Input Layer (784 Units)
  - Visual Input Layer (784 Units)

#### Software Simulation (jAER):
- 5.8ms latency
- 94.1% accuracy

---

Training Spiking Deep Belief Networks

- Offline training of RBMs with Contrastive Divergence
- Use linear-threshold units instead of binary units, replace by LIF
- Approximate LIF firing rate with **Siegert function**
- Usual RBM training, replacing sigmoid transfer function with Siegert
- Transfer trained weights to equivalent spiking DBN

\[ \mu_Q = \tau_m \cdot w \cdot \lambda_{in} \]
\[ \sigma_Q^2 = \frac{\tau_m}{2} \cdot w^2 \cdot \lambda_{in} \]
\[ \Upsilon = V_{\text{rest}} + \mu_Q \]
\[ \Gamma = \sigma_Q \]
\[ \lambda_{out} = \Phi(\Upsilon, \Gamma) \]
\[ = \left( t_{\text{ref}} + \frac{V_{\text{th}} + k \gamma \Gamma}{V_{\text{reset}} + k \gamma \Gamma} \int_{V_{\text{reset}} + k \gamma \Gamma}^{V_{\text{th}} + k \gamma \Gamma} du \cdot \exp \left[ \frac{(u - \gamma)^2}{2 \Gamma^2} \right] \cdot \left[ 1 + \text{erf} \left( \frac{u - \gamma}{\Gamma \sqrt{2}} \right) \right] \right)^{-1} \]

Siegert model: \( \lambda_{\text{in/out}} \) ... Poisson input/output rate

Siegert (1951); Jug et al. (2012)
Multi-sensory Association in Real-time
Real-time Classification and Sensor Fusion

Link to event-based RBM / DBN / sensor-fusion videos:

https://sites.google.com/site/thebrainbells/home/event-driven-rbms
## Visual Recognition with Distractors

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td><img src="image1.png" alt="Image" /></td>
<td><img src="image2.png" alt="Image" /></td>
<td><img src="image3.png" alt="Image" /></td>
<td><img src="image4.png" alt="Image" /></td>
<td><img src="image5.png" alt="Image" /></td>
<td><img src="image6.png" alt="Image" /></td>
</tr>
<tr>
<td>B</td>
<td><img src="image7.png" alt="Image" /></td>
<td><img src="image8.png" alt="Image" /></td>
<td><img src="image9.png" alt="Image" /></td>
<td><img src="image10.png" alt="Image" /></td>
<td><img src="image11.png" alt="Image" /></td>
<td><img src="image12.png" alt="Image" /></td>
</tr>
<tr>
<td>C</td>
<td><img src="image13.png" alt="Image" /></td>
<td><img src="image14.png" alt="Image" /></td>
<td><img src="image15.png" alt="Image" /></td>
<td><img src="image16.png" alt="Image" /></td>
<td><img src="image17.png" alt="Image" /></td>
<td><img src="image18.png" alt="Image" /></td>
</tr>
<tr>
<td>D</td>
<td><img src="image19.png" alt="Image" /></td>
<td><img src="image20.png" alt="Image" /></td>
<td><img src="image21.png" alt="Image" /></td>
<td><img src="image22.png" alt="Image" /></td>
<td><img src="image23.png" alt="Image" /></td>
<td><img src="image24.png" alt="Image" /></td>
</tr>
</tbody>
</table>
The final block
Applications: What to build?

Specialized sensory processing (e.g. gestures, robotics, fusion, …)

Event-based Machine Learning (e.g. DBN, EM, …)

Models of biology (e.g. WTA, cortical hierarchies, …)

Spatio-temporal processing (e.g. state-machines, HMM, …)

Configuring hardware (e.g. mismatch, scaling, …)
Summary

• Building blocks of spiking components for specialized and general purpose applications
  • Sensors as first stage of processing
  • Synthesizing state-machines
  • Learning and probabilistic inference

• Links between machine learning methods and biological plasticity paradigms like STDP

• Deep architectures are more efficient in event-based systems, and can be used for complex classification and sensory fusion tasks

• Suitable for hardware implementation
  • Open issues: reliability, configuration, online adaptation, scaling
Capo Caccia Cognitive Neuromorphic Engineering Workshop

Alghero, Sardinia (Italy)
28 April – 10 May 2014

capocaccia.ethz.ch
Acknowledgements

UZH and ETHZ
Rodney Douglas
Tobi Delbruck
Shih-Chii Liu
Giacomo Indiveri
Danny Neil
Peter O’Connor
Dane Corneil
Emre Neftci

SAIT
Jun Haeng Lee
Hyunsurk Ryu

TU Graz
Wolfgang Maass
Bernhard Nessler

Caltech
Ueli Rutishauser

Funding