Spiking Neural Networks for Low-Power Edge Intelligence

Osvaldo Simeone
Joint work with Hyeryung Jang (KCL), Bipin Rajendran (NJIT), Brian Gardner (USurrey), and André Grüning (HStralsund)

King's College London

University of Luxembourg, 18/9/2019

kclip
King's Centre for Learning & Information Processing

KING'S College LONDON

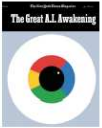
Overview

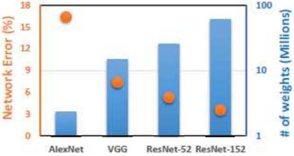
- Motivation and Introduction
- Applications
- Models
- Learning Algorithms
- Examples

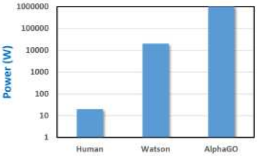
Overview

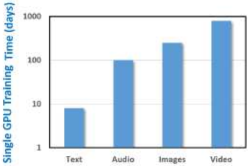
- Motivation and Introduction
- Applications
- Models
- Learning Algorithms
- Examples

Machine Learning Today









[Rajendran '18]

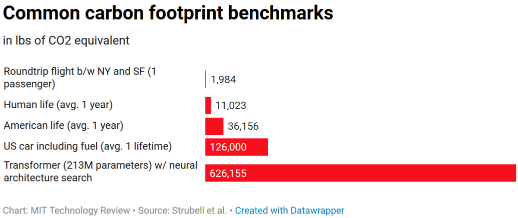
- Breakthroughs in ML using (deep) Artificial Neural Networks (ANNs) have come at the expense of massive memory, energy, and time requirements...

Machine Learning Today

Artificial Intelligence / Machine Learning

MIT Technology Review

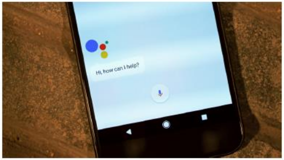
Training a single AI model can emit as much carbon as five cars in their lifetimes



- Breakthroughs in ML using (deep) Artificial Neural Networks (ANNs) have come at the expense of massive memory, energy, and time requirements...

Resource-Constrained Machine Learning

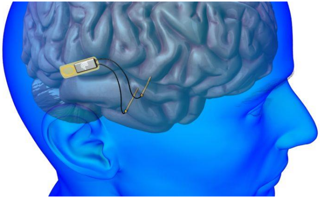
- How to implement ML (inference and learning) on mobile or embedded devices with limited energy and memory resources? [Welling '18]



mobile personal assistants

medical and health wearables

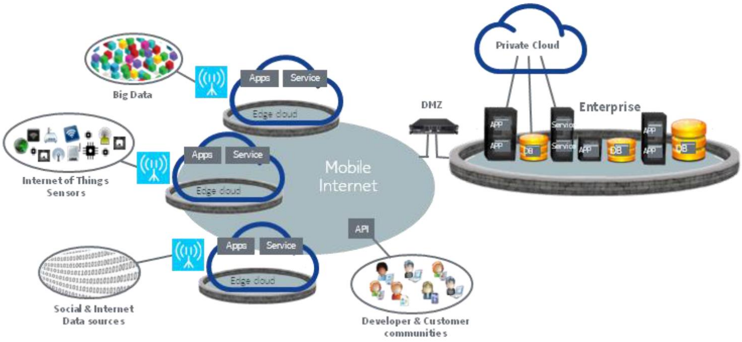
IoT mobile or embedded devices



neural prosthetics

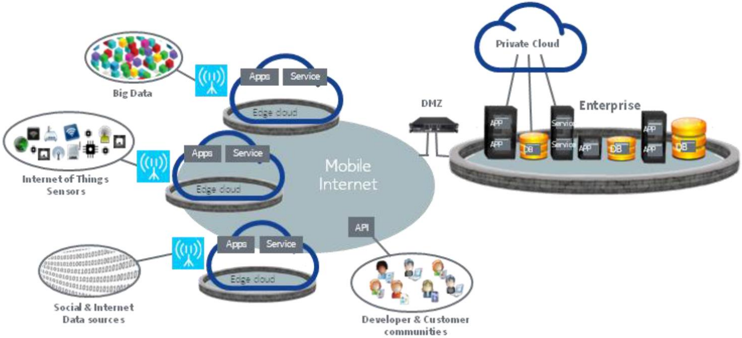
Machine Learning at the Edge

- A solution is mobile edge or cloud computing: offload computations to an edge or cloud server.
- Possible privacy and latency issues



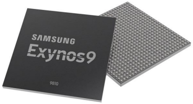
Machine Learning at the Edge

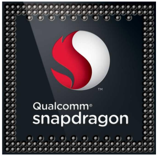
- A solution is mobile edge or cloud computing: offload computations to an edge or cloud server.
- Possible privacy and latency issues

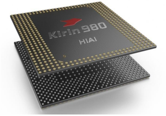


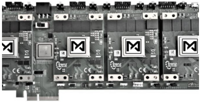
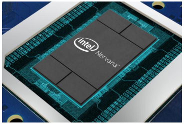
Machine Learning on Mobile Devices

- Another solution is to scale down energy and memory requirements of ANNs via tailored hardware implementations for mobile devices.
- Active field with established players and start-ups
- Trade-offs between accuracy and complexity
- Limited to inference



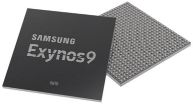


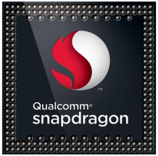


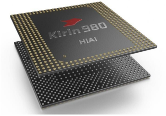



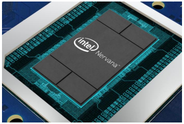
Machine Learning on Mobile Devices

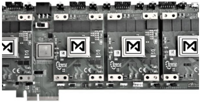
- Another solution is to scale down energy and memory requirements of ANNs via tailored hardware implementations for mobile devices.
- Active field with established players and start-ups
- Trade-offs between accuracy and complexity
- Limited to inference







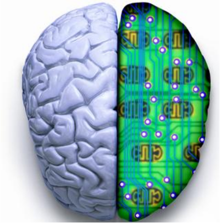




Neuromorphic Computing

- Spiking Neural Networks (SNNs) aim at attaining the efficiency of the human brain...

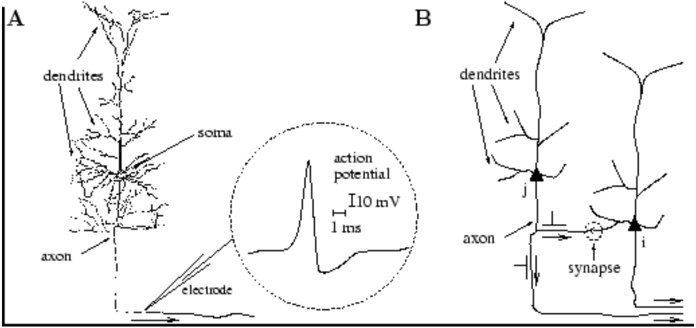
13 Million Watts
5600 sq. ft. & 340 tons
$\sim 10^{10}$ ops/J



$\sim$ 20 Watts
2 sq. ft. & 1.4 Kg
$\sim 10^{15}$ ops/J

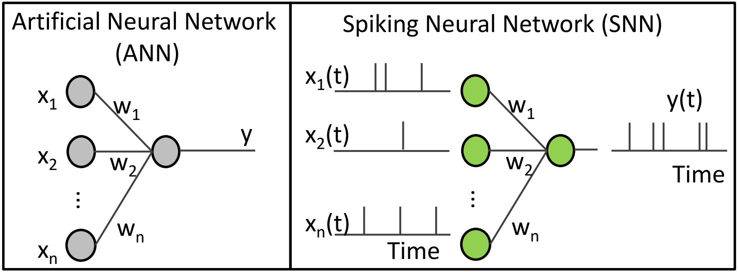
Neuromorphic Computing

- ... by taking inspiration from the dynamic, sparse, and event-driven learning and inference operation of the human brain.
- Neurons in the brain sense, process, and communicate over time using sparse binary signals (spikes or action potentials).



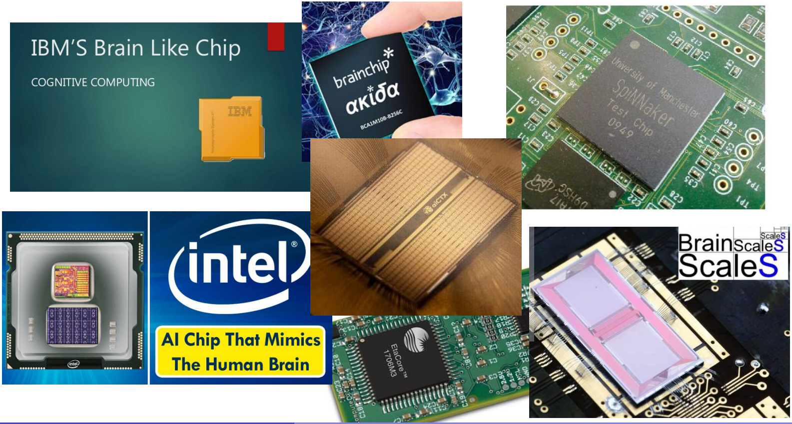
Spiking Neural Networks

- SNNs are networks of spiking neurons [Maas '97].
- Mostly studied in theoretical neuroscience, but recent interest from machine learning and hardware design researchers...



Spiking Neural Networks

- Proof-of-concept and commercial hardware implementations of SNNs have demonstrated significant energy savings as compared to ANNs [Rajendran et al '19].
- Energy consumed only when spikes are produced (binary values $\rightarrow$ pJ/spike)



Spiking Neural Networks

- Increasing press coverage and positive market predictions...

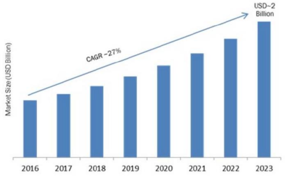
WIRED

The future of AI is neuromorphic. Meet the scientists building digital 'brains' for your phone

Neuromorphic chips are being designed to specifically mimic the human brain – and they could soon replace CPUs

Self-Learning Neuromorphic Chip Market Research Report- Global Forecast 2023

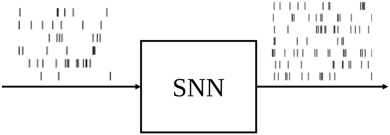
MARKET RESEARCH FUTURE®



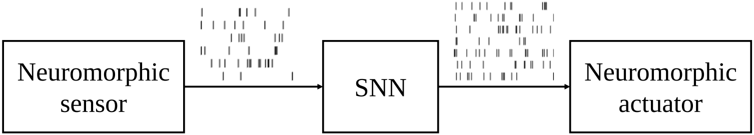
Overview

- Motivation and Introduction
- Applications
- Models
- Learning Algorithms
- Examples

I/O Interfaces

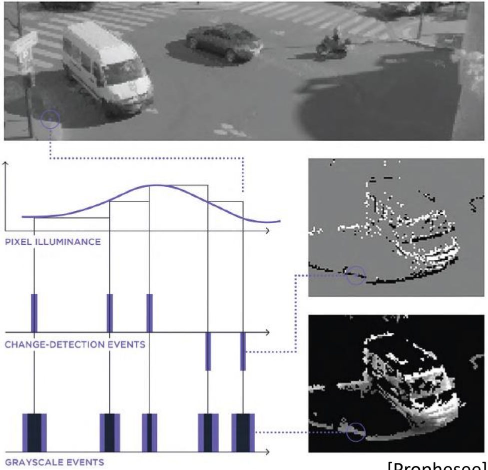


I/O Interfaces



I/O Interfaces

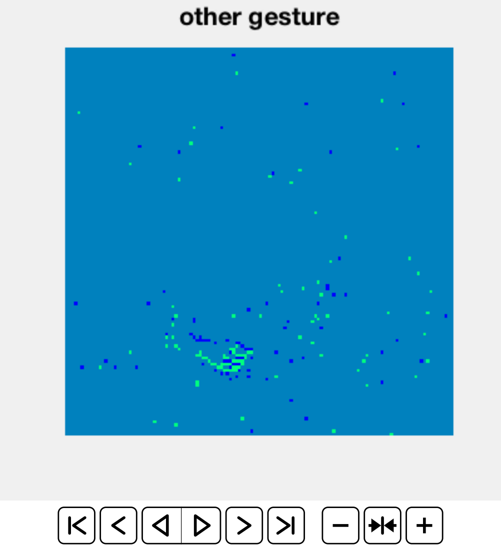
- iniVation's Dynamic Vision Sensor (DVS), iniVation and AiCTX's Speck, Prophesee



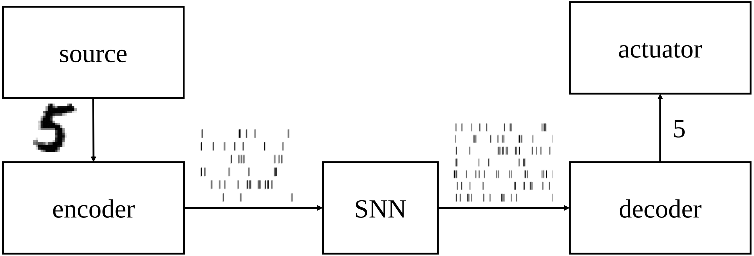
[Prophesee]

I/O Interfaces

- From IBM's DVS128 Gesture Dataset...

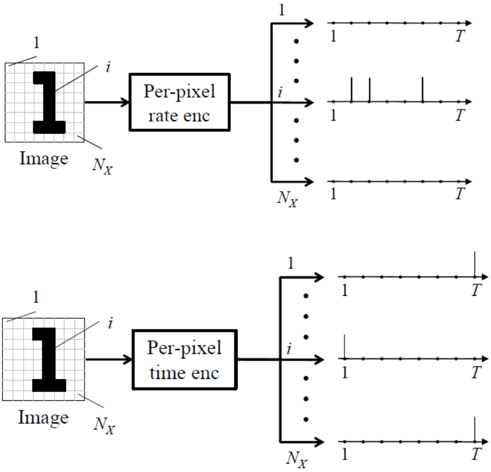


I/O Interfaces



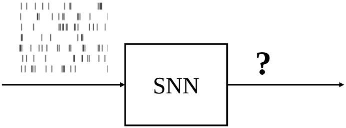
I/O Interfaces

- Conversion from natural signals to spike signals – theory [Lazar '06]
- Practice: Rate encoding, time encoding, population encoding; rate decoding, first-to-spike decoding,...



Applications

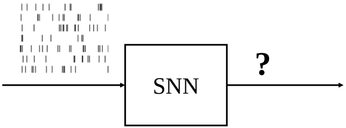
- SNNs can perform
  - Inference/ control: classification, regression, prediction, ...



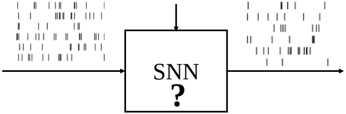
(a) inference/ control

Applications

- SNNs can perform
  - Inference/ control: classification, regression, prediction, ...
  - Learning: supervised, unsupervised, reinforced



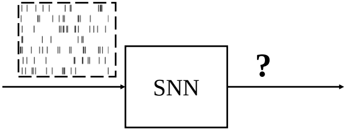
(a) inference/ control



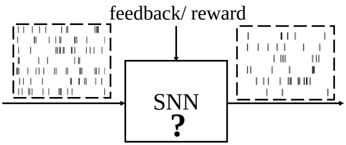
(b) learning

Applications

- SNNs can perform
  - Inference/ control: classification, regression, prediction, ...
  - Learning: supervised, unsupervised, reinforced



(a) inference/ control

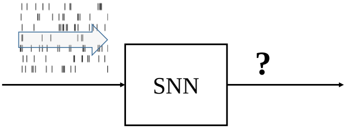


(b) learning

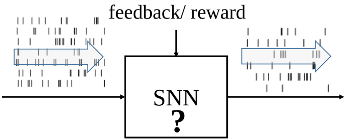
- Data presentation through I/O interfaces:
  - Frame-based (or batch)

Applications

- SNNs can perform
  - Inference/ control: classification, regression, prediction, ...
  - Learning: supervised, unsupervised, reinforced



(a) inference/ control



(b) learning

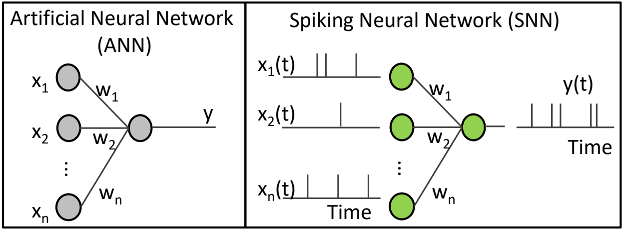
- Data presentation through I/O interfaces:
  - Frame-based (or batch)
  - Streaming (or online)

Overview

- Motivation and Introduction
- Applications
- Models
- Learning Algorithms
- Examples

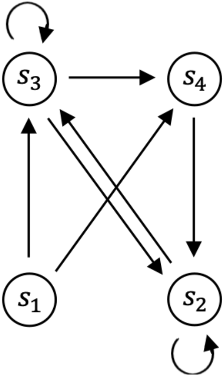
SNN Models

- SNNs are networks of spiking neurons.
- Their operation is defined by:
  - topology (connectivity)
  - neuron model



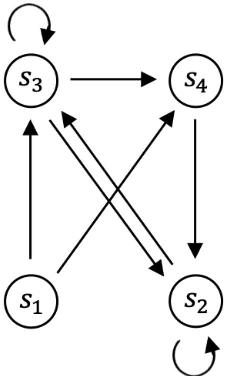
Topology

- Arbitrary directed graph with directed links representing synaptic connections
- "Parent", or pre-synaptic, neuron affects causally spiking behavior of "child", or post-synaptic, neuron
- Enables recurrent connectivity (directed loops), including self-loops



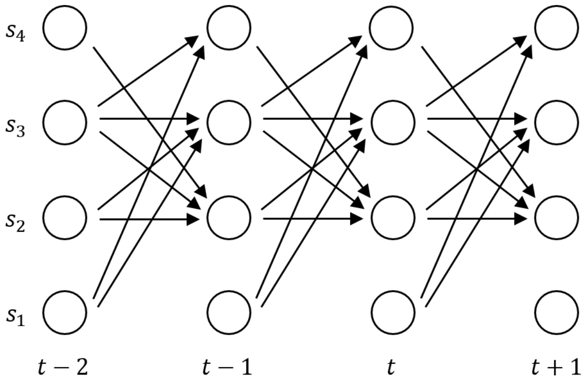
Topology

- Arbitrary directed graph with directed links representing synaptic connections
- "Parent", or pre-synaptic, neuron affects causally spiking behavior of "child", or post-synaptic, neuron
- Enables recurrent connectivity (directed loops), including self-loops



Topology

- Discrete (algorithmic) time, as in many practical implementations (e.g., Intel's Loihi)
- Binary outputs: $0 =$ no spike; and $1 =$ spike (energy consumed)



Neuron Model

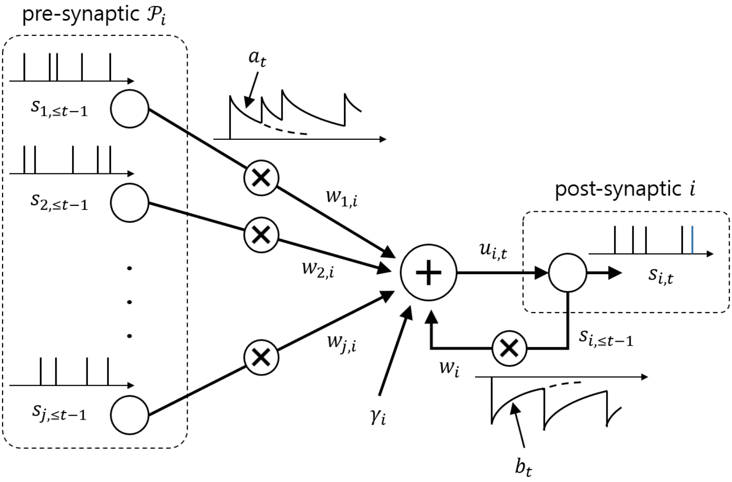
- Each neuron is characterized by an internal state known as membrane potential [Gerstner and Kistler '02].
- Generally, a higher membrane potentially entails a larger propensity for spiking.
- The membrane potential evolves over time as a function of the past behavior of pre-synaptic neurons and of the neuron itself.

Neuron Model

- Each neuron is characterized by an internal state known as membrane potential [Gerstner and Kistler '02].
- Generally, a higher membrane potentially entails a larger propensity for spiking.
- The membrane potential evolves over time as a function of the past behavior of pre-synaptic neurons and of the neuron itself.

Membrane Potential

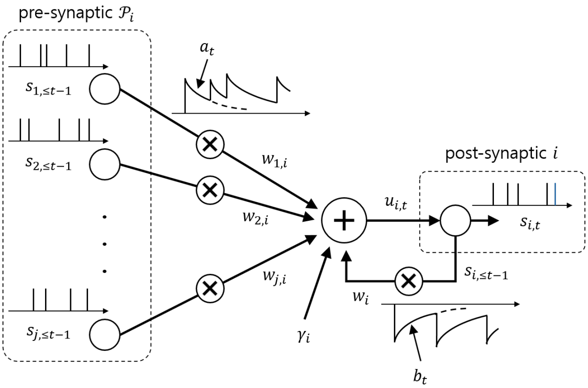
$$u_{i,t} = \sum_{j \in \mathcal{P}_i} w_{j,i} \left(a_t * s_{j,t} \right) + w_i \left(b_t * s_{i,t} \right) + \gamma_i$$



Membrane Potential

$$u_{i,t} = \sum_{j \in \mathcal{P}_i} w_{j,i} \left(a_t * s_{j,t} \right) + w_i \left(b_t * s_{i,t} \right) + \gamma_i$$

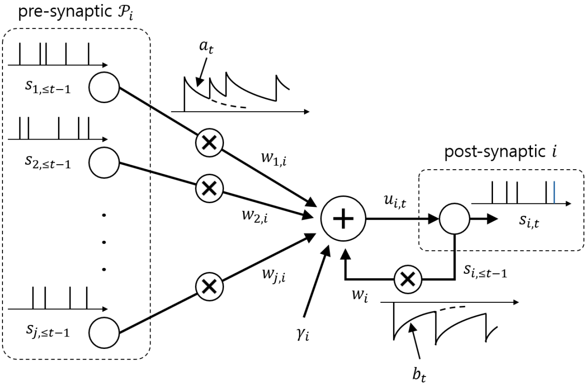
- Feedforward filter (kernel) a_t with learnable synaptic weight $w_{j,i}$



Membrane Potential

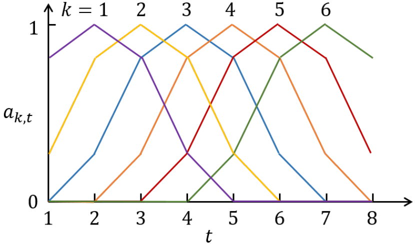
$$u_{i,t} = \sum_{j \in \mathcal{P}_i} w_{j,i}\left(a_t * s_{j,t}\right) + w_i\left(b_t * s_{i,t}\right) + \gamma_i$$

- Feedback filter (kernel) b_t with learnable w_i (e.g., refractory period)
- Bias (threshold) γ_i



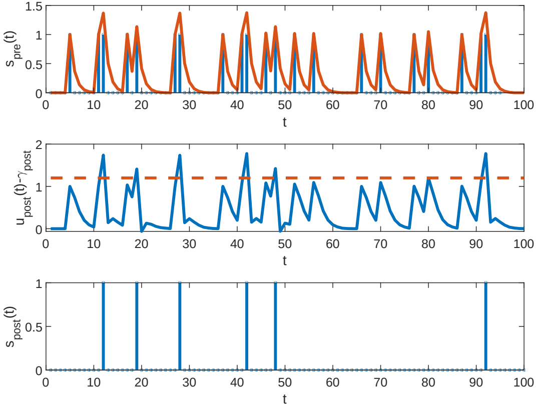
Membrane Potential

- Kernels can more generally be parameterized via multiple basis functions and learnable weights [Pillow et al '08].
- This allows learning of the temporal "receptive fields" of the neurons, e.g., by adapting synaptic delays.



Deterministic SNN Models

- The most common model is leaky integrate-and-fire (LIF) [Gerstner and Kistler '02]: Spike when membrane potential is positive



Deterministic SNN Models

- LIF-based SNNs can approximate operation of feedforward and recurrent ANNs: spiking rates in SNN $\approx$ neuron outputs in ANN
- Often used for inference by converting a pre-trained ANN [Rueckauer et al '17]
- Direct training is required to leverage temporal information processing and learning
- This is made difficult by output non-differentiability with respect to model parameters
- Heuristic training algorithms based on approximations, such as surrogate gradient [Neftci '18] [Anwani and Rajendran '18]
- As for ANNs, these require backpropagation

Deterministic SNN Models

- LIF-based SNNs can approximate operation of feedforward and recurrent ANNs: spiking rates in SNN $\approx$ neuron outputs in ANN
- Often used for inference by converting a pre-trained ANN [Rueckauer et al '17]
- Direct training is required to leverage temporal information processing and learning
- This is made difficult by output non-differentiability with respect to model parameters
- Heuristic training algorithms based on approximations, such as surrogate gradient [Neftci '18] [Anwani and Rajendran '18]
- As for ANNs, these require backpropagation

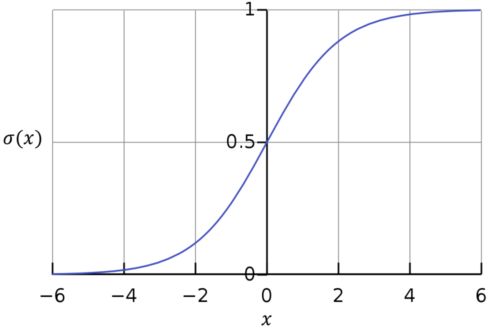
Deterministic SNN Models

- LIF-based SNNs can approximate operation of feedforward and recurrent ANNs: spiking rates in SNN $\approx$ neuron outputs in ANN
- Often used for inference by converting a pre-trained ANN [Rueckauer et al '17]
- Direct training is required to leverage temporal information processing and learning
- This is made difficult by output non-differentiability with respect to model parameters
- Heuristic training algorithms based on approximations, such as surrogate gradient [Neftci '18] [Anwani and Rajendran '18]
- As for ANNs, these require backpropagation

Probabilistic SNN Models

- Probabilistic Generalized Linear Model (GLM): Conditional spiking probability [Pillow et al '08]

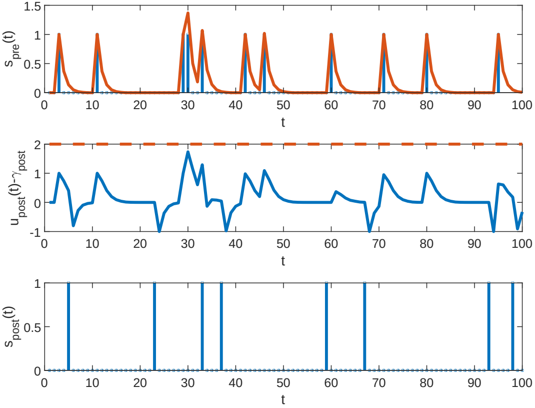
$$p(s_{i,t} = 1 | \mathbf{s}_{\leq t-1}) = \sigma(u_{i,t})$$



Probabilistic SNN Models

- Probabilistic Generalized Linear Model (GLM): Conditional spiking probability [Pillow et al '08]

$$p(s_{i,t} = 1 | \mathbf{s}_{\leq t-1}) = \sigma(u_{i,t})$$



Probabilistic SNN Models

- GLM SNN models are dynamic generalization of belief networks [Neal '92].
- They enable direct training of spike-based statistical criteria.

Overview

- Motivation and Introduction
- Applications
- Models
- Learning Algorithms
- Examples

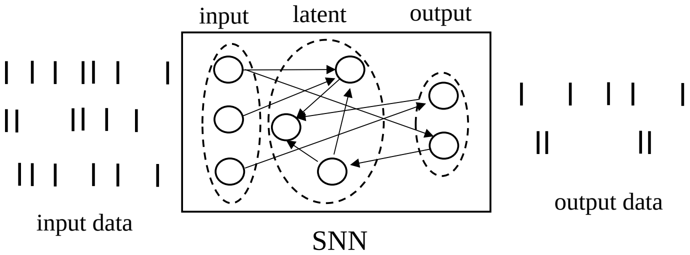
Training SNNs

- SNNs can be trained using supervised, unsupervised, and reinforcement learning.
- To fix the ideas, we focus here on supervised learning:

$$\text{training data} = \{(\text{input}, \text{output})\} \rightarrow \text{generalization}$$

Training SNNs

- Visible neurons, clamped to input/ output data
- Latent neurons, free



Learning Rules

- Probabilistic SNNs can be directly trained using standard statistical criteria, such as maximum likelihood, maximum mutual information, or maximum average return.
- Maximum likelihood for supervised (and unsupervised) learning:

$$\max \mathrm{E}_{\mathrm{data}}[\ln p(\mathrm{visible})] = \mathrm{E}_{\mathrm{data}}[\ln \mathrm{E}_{\mathrm{latent}} p(\mathrm{visible}, \mathrm{latent})]$$

- Regularization terms are typically added, e.g., to limit spiking rate (and hence energy consumption) – bounded rationality [Leibfried and Braun '15]

Learning Rules

- Probabilistic SNNs can be directly trained using standard statistical criteria, such as maximum likelihood, maximum mutual information, or maximum average return.
- Maximum likelihood for supervised (and unsupervised) learning:

$$\max \mathrm{E}_{\text{data}}[\ln p(\text{visible})] = \mathrm{E}_{\text{data}}[\ln \mathrm{E}_{\text{latent}} p(\text{visible},\text{latent})]$$

- Regularization terms are typically added, e.g., to limit spiking rate (and hence energy consumption) – bounded rationality [Leibfried and Braun '15]

Learning Rules

- Probabilistic SNNs can be directly trained using standard statistical criteria, such as maximum likelihood, maximum mutual information, or maximum average return.
- Maximum likelihood for supervised (and unsupervised) learning:

$$\max \mathrm{E}_{\text{data}}[\ln p(\text{visible})] = \mathrm{E}_{\text{data}}[\ln \mathrm{E}_{\text{latent}} p(\text{visible}, \text{latent})]$$

- Regularization terms are typically added, e.g., to limit spiking rate (and hence energy consumption) – bounded rationality [Leibfried and Braun '15]

Learning Rules

- Using Stochastic Gradient Descent (SGD) along with...
- ... variational inference to address marginalization over latent neurons yields the on-line learning rule [Rezende et al '11] [Osogami '17] [Jang et al '18]

$$w_{j,i} \leftarrow w_{j,i} + \eta \times \ell \times \text{pre-syn}_j \times \text{post-syn}_i$$

- The learning signal ℓ is a global feedback signal (akin to neuromodulator in neuroscience).
- Pre-synaptic and post-synaptic terms are local to each synapse.

Learning Rules

- Using Stochastic Gradient Descent (SGD) along with...
- ... variational inference to address marginalization over latent neurons yields the on-line learning rule [Rezende et al '11] [Osogami '17] [Jang et al '18]

$$w_{j,i} \leftarrow w_{j,i} + \eta \times \ell \times \text{pre-syn}_j \times \text{post-syn}_i$$

- The learning signal ℓ is a global feedback signal (akin to neuromodulator in neuroscience).
- Pre-synaptic and post-synaptic terms are local to each synapse.

Learning Rules

- Using Stochastic Gradient Descent (SGD) along with...
- ... variational inference to address marginalization over latent neurons yields the on-line learning rule [Rezende et al '11] [Osogami '17] [Jang et al '18]

$$w_{j,i} \leftarrow w_{j,i} + \eta \times \ell \times \text{pre-syn}_j \times \text{post-syn}_i$$

- The learning signal ℓ is a global feedback signal (akin to neuromodulator in neuroscience).
- Pre-synaptic and post-synaptic terms are local to each synapse.

Learning Rules

- Using Stochastic Gradient Descent (SGD) along with...
- ... variational inference to address marginalization over latent neurons yields the on-line learning rule [Rezende et al '11] [Osogami '17] [Jang et al '18]

$$w_{j,i} \leftarrow w_{j,i} + \eta \times \ell \times \text{pre-syn}_j \times \text{post-syn}_i$$

- The learning signal ℓ is a global feedback signal (akin to neuromodulator in neuroscience).
- Pre-synaptic and post-synaptic terms are local to each synapse.

Learning Rules

- Stochastic Gradient Descent (SGD) updates for synaptic weights:

$$w_{j,i} \leftarrow w_{j,i} + \eta \times \ell \times \text{pre-syn}_j \times \text{post-syn}_i$$

- The learning signal ℓ is derived using variational inference [Rezende et al '11] [Osogami '17] [Jang et al '18]:

$$\ell_t = 1 \text{ for visible (input and output) neurons,}$$

$$\ell_t = \sum_{i \in \text{visible}} \log p(s_{i,t}|u_{i,t}) \text{ for latent neurons}$$

- Positive feedback to hidden neurons if desired behavior has large probability

Learning Rules

- Stochastic Gradient Descent (SGD) updates for synaptic weights:

$$w_{j,i} \leftarrow w_{j,i} + \eta \times \ell \times \text{pre-syn}_j \times \text{post-syn}_i$$

- $\text{pre-syn}_j = a_t * s_{j,t} =$ pre-synaptic trace
- Large if previous behavior of pre-synaptic neuron is consistent with synaptic receptive field (e.g., if recent spiking)

Learning Rules

- Stochastic Gradient Descent (SGD) updates for synaptic weights:

$$w_{j,i} \leftarrow w_{j,i} + \eta \times \ell \times \text{pre-syn}_j \times \text{post-syn}_i$$

- $\text{post-syn}_i = s_{i,t} - \sigma\big(u_{i,t}\big) =$ post-synaptic error
- Post-synaptic error = desired/ observed behavior - model averaged behavior [Bienenstock et al '82]
- No backpropagation

Learning Rules

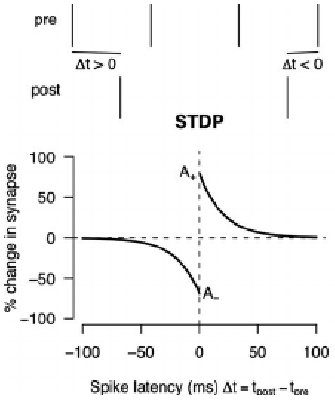
- Stochastic Gradient Descent (SGD) updates for synaptic weights:

$$w_{j,i} \leftarrow w_{j,i} + \eta \times \ell \times \text{pre-syn}_j \times \text{post-syn}_i$$

- $\text{post-syn}_i = s_{i,t} - \sigma\big(u_{i,t}\big) = \text{post-synaptic error}$
- Post-synaptic error = desired/ observed behavior - model averaged behavior [Bienenstock et al '82]
- No backpropagation

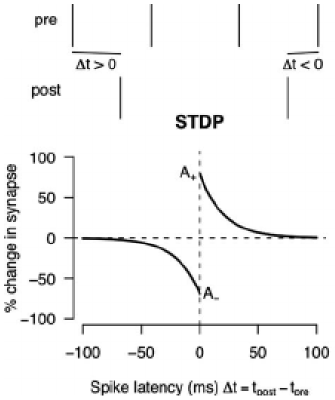
Learning Rules

- The learning rules simplifies to Spike-Timing-Dependent Plasticity (STDP) [Markram '97].
- Further simplifications yield Hebbian learning: "Neurons that fire together, wire together" [Hebb '49].



Learning Rules

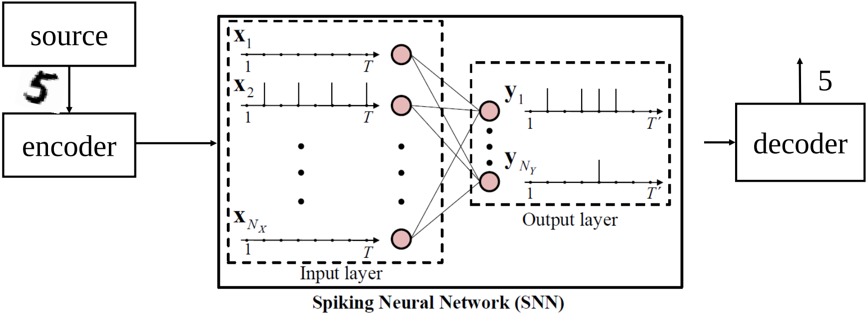
- The learning rules simplifies to Spike-Timing-Dependent Plasticity (STDP) [Markram '97].
- Further simplifications yield Hebbian learning: "Neurons that fire together, wire together" [Hebb '49].



Overview

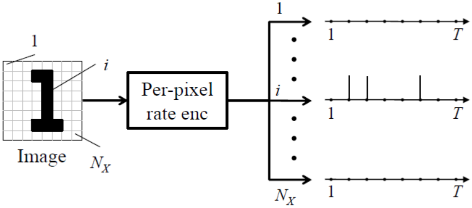
- Motivation and Introduction
- Applications
- Models
- Learning Algorithms
- Examples

Batch Training



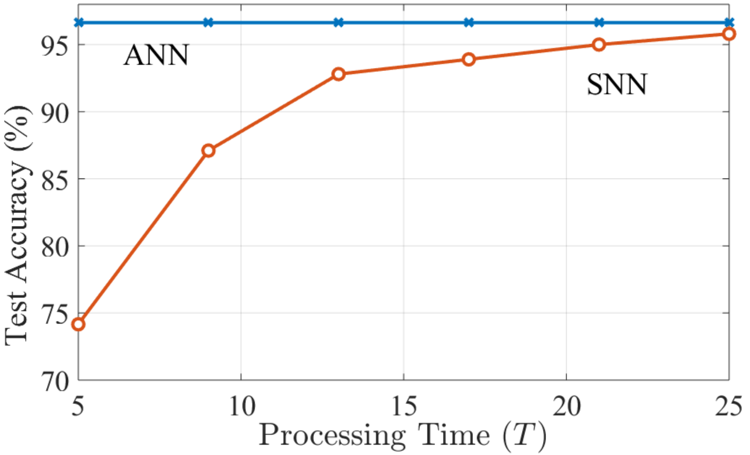
Batch Training

- Rate encoding/ decoding



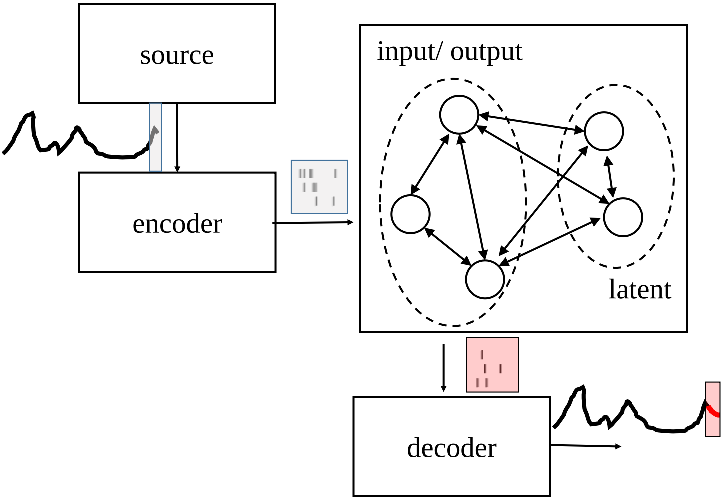
Batch Training

- Graceful trade-off complexity/ delay vs accuracy [Jang et al '18-2]



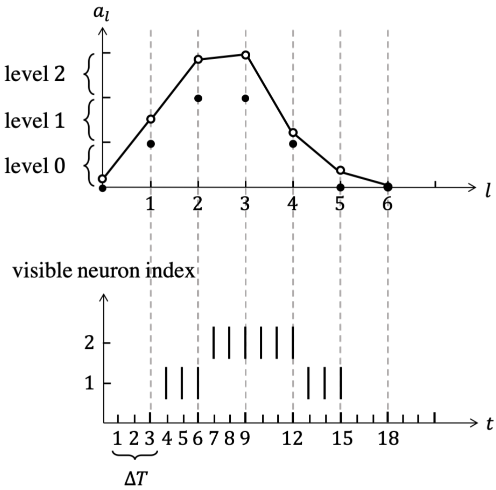
Online Training

- Online training for prediction
- Fully connected (recurrent) SNN topology with hidden neurons



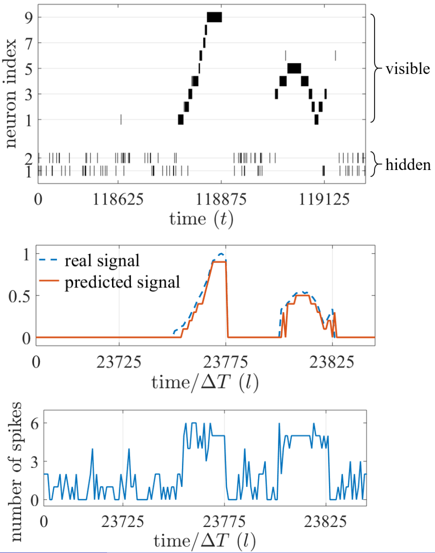
Online Training

- Rate encoding



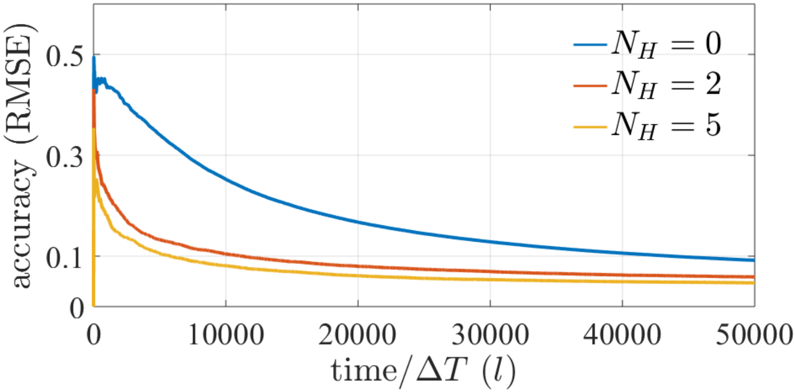
Online Training

- Rate encoding



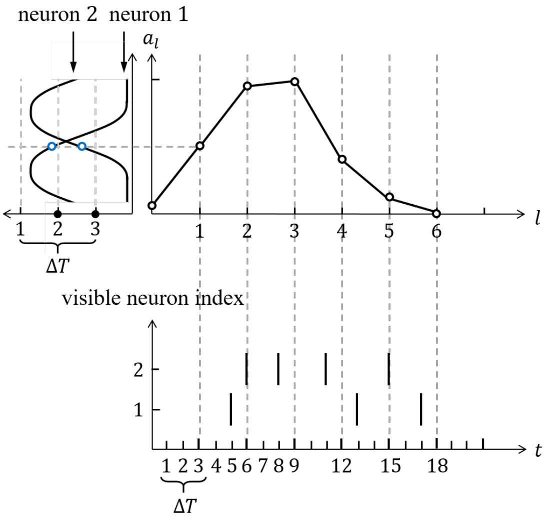
Online Training

- Rate encoding



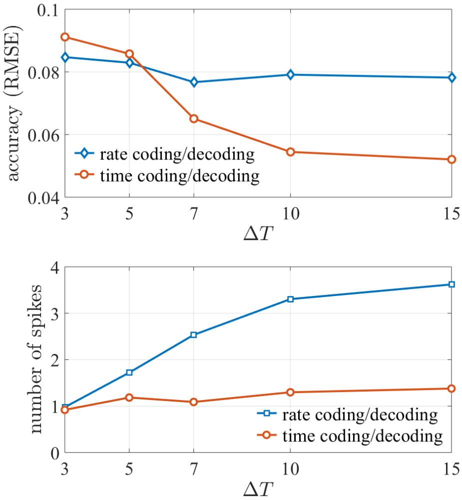
Online Training

- Time encoding



Online Training

- Rate vs time encoding



Concluding Remarks

- Statistical signal processing review of neuromorphic computing via Spiking Neural Networks
- Additional topics:
  - more general energy-based probabilistic SNN models [Osogami '17] [Jang '19]
  - recurrent SNNs for long-term memory [Maas '11]
  - neural sampling: information encoded in steady-state behavior [Buesing et al '11]
  - Bayesian learning via Langevin dynamics [Pecevski et al '11] [Kappel et al '15]
- Some open problems:
  - meta-learning, life-long learning, transfer learning [Bellec et al '18]
  - training I/O interfaces [Lazar and Toth '03]
  - integration of ANNs and SNNs [Pei et al '19]

For More...

- H. Jang, O. Simeone, B. Gardner and A. Gruning, "An Introduction to Probabilistic Spiking Neural Networks," to appear on IEEE Signal Processing Magazine (available on arxiv).

Acknowledgements

This work has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. 725731) and from the US National Science Foundation (NSF) under grant ECCS 1710009.

References

[Gerstner and Kistler '02] W. Gerstner and W. M. Kistler, Spiking neuron models: Single neurons, populations, plasticity. Cambridge University Press, 2002.

[Pillow et al '08] J.W. Pillow, J. Shlens, L. Paninski, A. Sher, A. M. Litke, E. Chichilnisky, and E. P. Simoncelli, "Spatio-temporal correlations and visual signalling in a complete neuronal population," Nature, vol. 454, no. 7207, p. 995, 2008.

[Osogami '17] T. Osogami, "Boltzmann machines for time-series," arXiv preprint arXiv:1708.06004, 2017.

[Ibnkahla '00] Ibnkahla M. Applications of neural networks to digital communications–a survey. Signal processing. 2000 Jul 1;80(7):1185-215.

[Koller and Friedman '09] Koller D, Friedman N, Bach F. Probabilistic graphical models: principles and techniques. MIT press; 2009.

References

[Fremaux et al '08] N. Fremaux and W. Gerstner, "Neuromodulated spike-timing-dependent plasticity, and theory of three-factor learning rules," Frontiers in neural circuits, vol. 9, p. 85, 2016.
[Jang et al '18] H. Jang, O. Simeone, B. Gardnerm and A. Gruning, "Spiking neural networks: A stochastic signal processing perspective," arXiv:1812.03929. ...
[Rezende et al '11] Rezende DJ, Wierstra D, Gerstner W. "Variational learning for recurrent spiking networks", In Advances in Neural Information Processing Systems, 2011 (pp. 136-144).
[Brea et al '13] J. Brea, W. Senn, and J.-P. Pfister, "Matching recall and storage in sequence learning with spiking neural networks," Journal of Neuroscience, vol. 33, no. 23, pp. 9565–9575, 2013.
[Hebb '49] D. Hebb, The Organization of Behavior. New York: Wiley and Sons, Nov. 1949.

References

[Bienenstock et al '82] E. L. Bienenstock, L. N. Cooper, and P. W. Munro, "Theory for the development of neuron selectivity: orientation specificity and binocular interaction in visual cortex," Journal of Neuroscience, vol. 2, no. 1, pp. 32–48, 1982.

[Pecevski et al '11] D. Pecevski, L. Buesing, and W. Maass, "Probabilistic inference in general graphical models through sampling in stochastic networks of spiking neurons," PLOS Computational Biology, vol. 7, no. 12, pp. 1–25, 12 2011.

[Rosenfeld et al '18] Rosenfeld B, Simeone O, Rajendran B. Learning First-to-Spike Policies for Neuromorphic Control Using Policy Gradients. arXiv preprint arXiv:1810.09977. 2018 Oct 23.

[Jang et al '19] Jang H, Simeone O. Training Dynamic Exponential Family Models with Examples and Lateral Dependencies for Generalized Neuromorphic Computing", in Proc. IEEE ICASSP 2019.

References

[Bellec et al '18] G. Bellec, D. Salaj, A. Subramoney, R. Legenstein, and W. Maass, "Long short-term memory and learning-to-learn in networks of spiking neurons," arXiv preprint arXiv:1803.09574, 2018.

[Kappel et al '15] Kappel D, Habenschuss S, Legenstein R, Maass W. Synaptic sampling: a Bayesian approach to neural network plasticity and rewiring. InAdvances in Neural Information Processing Systems 2015 (pp. 370-378).

[Buesing et al '11] Buesing L, Bill J, Nessler B, Maass W. Neural dynamics as sampling: a model for stochastic computation in recurrent networks of spiking neurons. PLoS computational biology. 2011 Nov 3;7(11):e1002211.

[Lazar and Toth '03] Lazar, Aurel A., and László T. Tóth. "Time encoding and perfect recovery of bandlimited signals." ICASSP (6). 2003.

[Maas '11] Maass W. Liquid state machines: motivation, theory, and applications. InComputability in context: computation and logic in the real world 2011 (pp. 275-296).

References

[Neftci '18] Neftci EO. Data and power efficient intelligence with neuromorphic learning machines. iScience. 2018 Jul 27;5:52.

[Anwani and Rajendran '18] N. Anwani and B. Rajendran, "Training Multilayer Spiking Neural Networks using NormAD based Spatio-Temporal Error Backpropagation," arXiv:1811.10678.

[Blouw et al '18] Peter Blouw, Xuan Choo, Eric Hunsberger, Chris Eliasmith, "Benchmarking Keyword Spotting Efficiency on Neuromorphic Hardware," arXiv:1812.01739.

[Binas et al '17] J Binas, D Neil, SC Liu, T Delbruck, "DDD17: End-to-end DAVIS driving dataset," arXiv:1711.01458, 2017.

[Rajendran et al '19] B. Rajendran, et al, "Low-Power Neuromorphic Hardware for Signal Processing Applications", arXiv:1901.03690.

References

[Welling '18] M. Welling, "Intelligence per kilowatt-hour", plenary talk, ICML 2018, https://youtu.be/7QhkvG4MUbk.

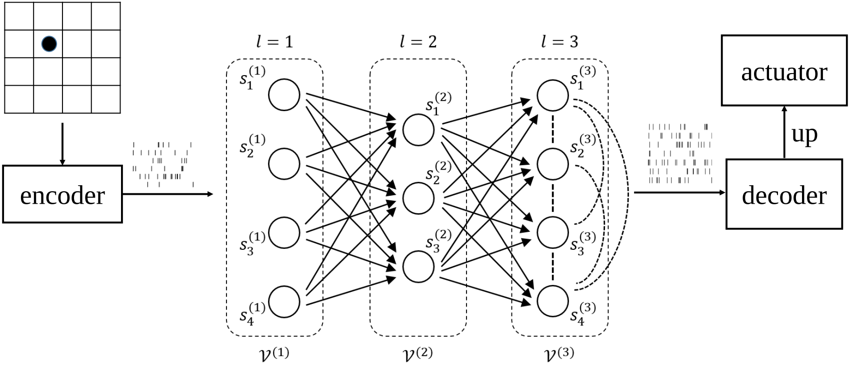
[Rueckauer '17] Rueckauer B, Lungu IA, Hu Y, Pfeiffer M, Liu SC. Conversion of continuous-valued deep networks to efficient event-driven networks for image classification. Frontiers in neuroscience. 2017 Dec 7;11:682.

[Neal '92] Neal RM. Connectionist learning of belief networks. Artificial intelligence. 1992 Jul 1;56(1):71-113.

[Leibfried and Braun '15] Leibfried, Felix, and Daniel A. Braun. "A reward-maximizing spiking neuron as a bounded rational decision maker." Neural computation 27.8 (2015): 1686-1720.

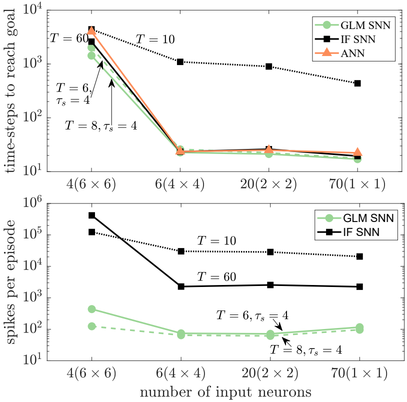
[Pei et al '19] J. Pei et al, "Towards artificial general intelligence with hybrid Tianjic chip architecture", Nature, Aug. 2019.

Reinforcement Learning



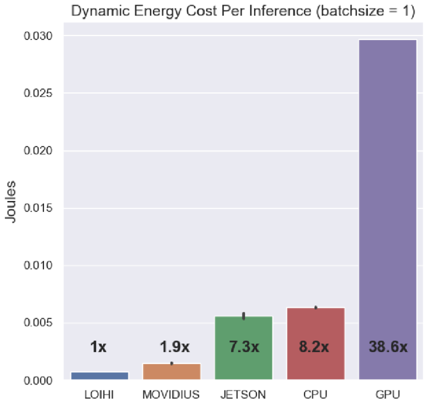
Reinforcement Learning

- From [Rosenfeld et al '18]



Applications

- Example: Keyword spotting based on audio streaming; accuracy comparable to ANN.



[Blouw et al '18]

Variational Inference (VI)

- With latent variables z, Maximum Likelihood requires the maximization of

$$\max_{\theta} \ \ln p(x|\theta) = \ln\left(\sum_{z} p(x, z|\theta)\right)$$
$$= \ln\left(\mathrm{E}_{\mathrm{z}\sim p(z|\theta)}[p(\mathrm{z}|x, \theta)]\right)$$

- Key issue: Need to marginalize over latent variables, whose distribution is to be learned.

Variational Inference (VI)

- VI tackles this problem by substituting the expectation for an optimization over a predictive, or variational, distribution $q(z|x, \varphi)$
- Optimization over both model parameters θ and variational parameters φ is done by maximizing the Evidence Lower BOund (ELBO) or (negative) free energy

$$\mathcal{L}(\theta, \varphi) = \mathrm{E}_{\mathrm{z} \sim q(z|x,\varphi)}[\underbrace{\ln p(x, \mathrm{z}|\theta) - \ln q(\mathrm{z}|x, \varphi)}_{\text{learning signal } \ell(x,z|\theta,\varphi)}]$$

Variational Inference (VI)

- VI tackles this problem by substituting the expectation for an optimization over a predictive, or variational, distribution $q(z|x,\varphi)$
- Optimization over both model parameters θ and variational parameters φ is done by maximizing the Evidence Lower BOund (ELBO) or (negative) free energy

$$\mathcal{L}(\theta,\varphi) = \mathrm{E}_{\mathrm{z}\sim q(z|x,\varphi)}[\underbrace{\ln p(x,\mathrm{z}|\theta) - \ln q(\mathrm{z}|x,\varphi)}_{\text{learning signal } \ell(x,z|\theta,\varphi)}]$$

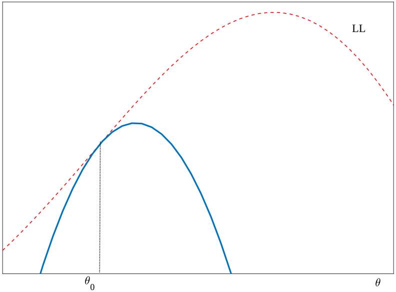
Variational Inference (VI)

- The ELBO is a global lower bound on the log-likelihood (LL) function

$$\ln p(x|\theta) \ \geq \mathcal{L}(\theta, \varphi),$$

- Equality holds at a value θ_0 if and only if the distribution $q(z|x, \varphi)$ is the posterior of z given x (i.e., optimal Bayesian estimate)

$$q(z|x, \varphi) = p(z|x, \theta_0).$$



Variational Inference (VI)

- VI approximates the EM algorithm by using SGD over both variables with gradients

$$\nabla_\theta L(\theta, \varphi) \approx \nabla_\theta \log p(x, z),$$
$$\nabla_\varphi L(\theta, \varphi) \approx \ell(x, z|\theta, \varphi) \cdot \nabla_\varphi \log q(z|x, \varphi)$$

with $z \sim q(z|x, \varphi)$

Variational Inference (VI) for SNNs

- In order to simplify the evaluation of the learning signal and the learning rule, a typical choice for GLM SNNs is

$$q(z_{i,t}|x_{\leq T}, z_{\leq t-1}) = p(z_{i,t}|u_{i,t}, \theta)$$

- This leads to a simplified learning signal

$$\ell_t = \sum_i \log p(x_{i,t}|u_{i,t})$$

  where the sum is over the observed spike trains.
- The learning signal measure the likelihood of the observed spike trains

Pseudocode for Online Learning

- For each time t
  - Sampling: Each hidden neuron i emits a spike with probability $\sigma(u_{i,t})$
  - Global feedback (1): A central processor updates the learning signal

$$\ell_t = \kappa \ell_{t-1} + (1-\kappa) \sum_i \log p(x_{i,t}|u_{i,t}),$$

    where the sum is over the observed neurons, which is fed back to all latent neurons
  - Global feedback (2): The central processor feeds back to all neurons any reward signal r_t (if reinforcement learning)
  - Local parameter update: Three-factor rule with $M_t = \ell_t \times r_t$ for hidden neurons and $M_t = r_t$ for the observed neurons