Classification of radio signals on a neuromorphic processor in space

Massimo Bortone

May 25, 2019

Institute of Neuroinformatics

Overview

1. Introduction
2. Modulation Recognition
3. Dataset
4. Delta Modulator
5. Reservoir computing
6. Results

Introduction

Introduction

What does a neuromorphic processor have to do in space?

Introduction

Space offers many opportunities for edge computing applications:

- autonomous robotic exploration
- scientific instruments
- radio communications

Introduction

Janette C. Briones, principal investigator in the NASA's cognitive communications project:

> *"Modern space communications systems use complex software to support science and exploration missions. By applying artificial intelligence and machine learning, satellites control these systems seamlessly, making real-time decisions without awaiting instruction."*

Introduction

Neuromorphic processors provide:

- low power consumption
- real-time online learning
- higher fault tolerance

Modulation Recognition

Modulation Recognition

What is a modulation process?

Modulation:
the process of encoding information onto a carrier signal by varying its properties

Carrier signal:
periodic waveform

$$c(t) = A(t)\sin\left(2\pi f_c t + \phi(t)\right)$$

with frequency f_c, amplitude A and phase ϕ

Modulation Recognition

Analog modulation:
transmit a continuous time signal $m(t)$ over an analog communication channel

Digital modulation (*keying*):
transmit a stream of bits $d[k]$ over an analog communication channel

IQ-representation:
a modulated signal $s(t)$ can be decomposed as a linear combination of sinusoidal basis functions

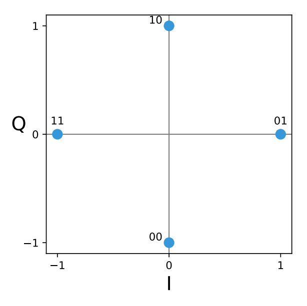
$$s(t) = I(t)\cos(2\pi f_c t) - Q(t)\sin(2\pi f_c t),$$

where the amplitudes are referred to as:

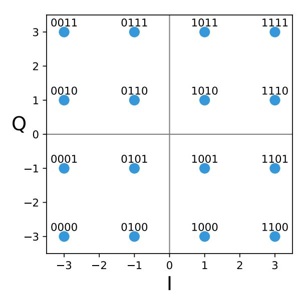
- in-phase component: $I(t) = \Re[s(t)]$
- quadrature component: $Q(t) = \Im[s(t)]$.

Modulation Recognition

Constellation diagrams for two digital modulations:



(a) QPSK



(b) QAM16

Figure 1: Blue dots represent baseband symbols used to encode the sequences of bits. QPSK can encode 2 bits per symbol, while QAM16 can do 4.

Modulation Recognition

Example of a modulated signal obtained with QPSK digital modulation:

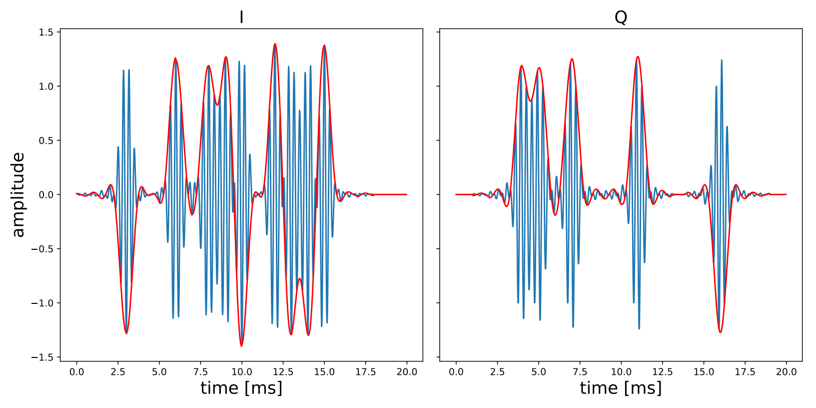


Figure 2: Carrier signal is shown in blue, while I and Q components are displayed in red.

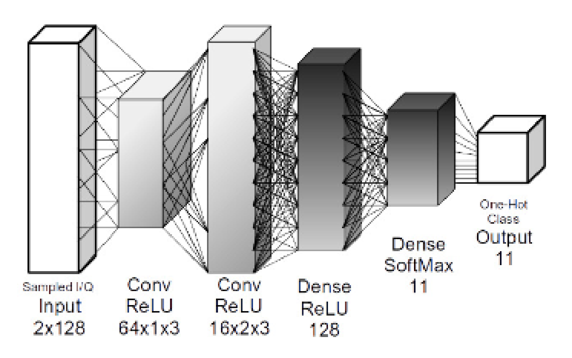
Modulation Recognition

Objective

Implement an automatic modulation recognition system on a neuromorphic processor

State of the art:

CNN developed by Tim O'Shea[3] reaches 87.4% accuracy across different SNRs



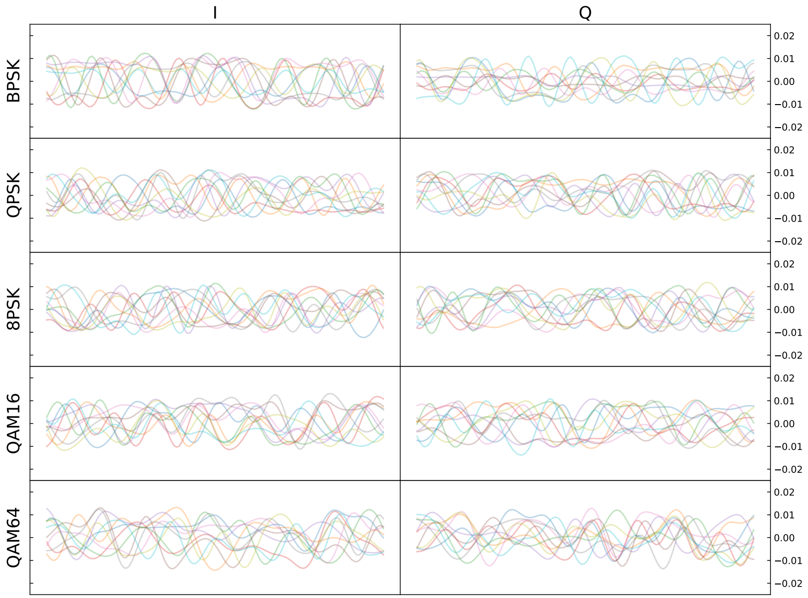
Dataset

Dataset

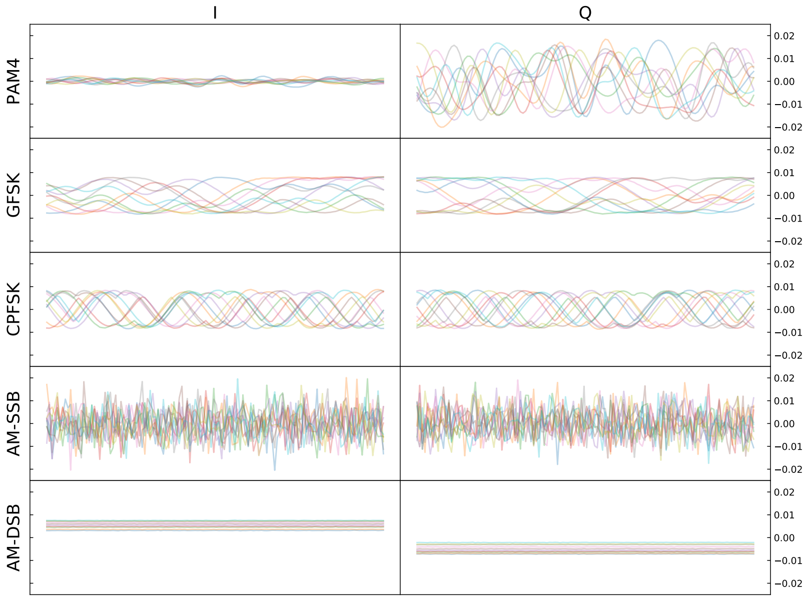
RadioML v2016.10a:

- synthetic radio signals with:
    - 11 modulations: 8 digital and 3 analog
    - 20 SNR levels: from -20dB to +20dB
    - 1000 samples per `(mod, snr)` tuple
    - sampled IQ-data at 1MHz for 128μs
- simulated channel effects:
    - random processes for central frequency offset
    - sample rate offset
    - additive white Gaussian noise
    - multi-path and selective fading

Dataset



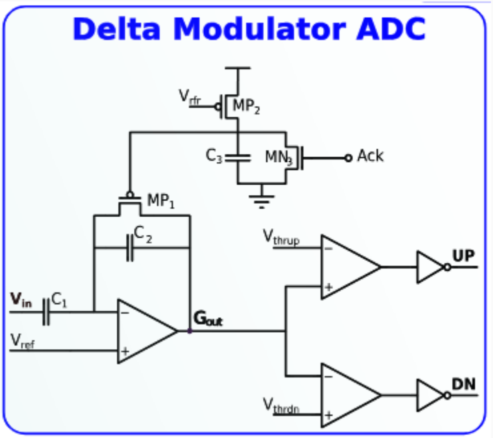
Dataset



Delta Modulator

Delta Modulator

Convert each component in the sampled IQ-data into spike trains using an approach similar to the Delta Modulator developed by Corradi et al. 2015



Delta Modulator

Steps in the conversion algorithm:

1. interpolate and resample IQ-data at higher frequency f_R
2. compare signal V_{in} with trailing thresholds V_{thrup} and V_{thrdn} at each time step
3. generate spikes in the UP and DN channel if the respective threshold is crossed

Parameters:

f_R, V_{thrup} and V_{thrdn}

Delta Modulator

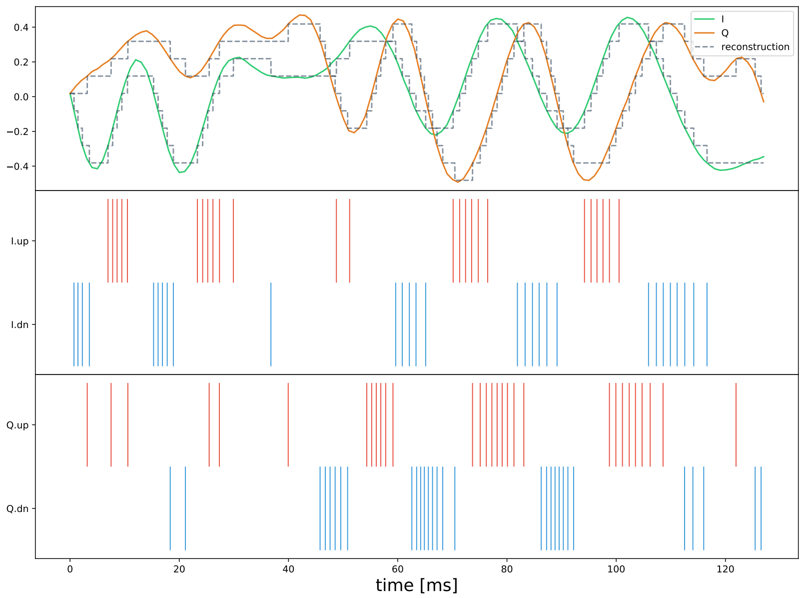
Optimize conversion parameters:

1. compute reconstructed signal V_{rec} from spike trains
2. minimize reconstruction error

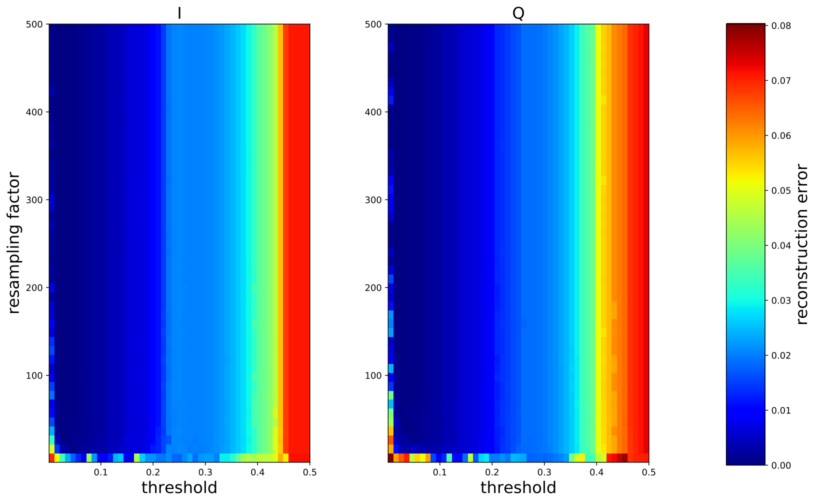
$$\epsilon_{rec} = \frac{1}{T} \sum_{i=1}^{T} (V_{in} - V_{rec})^2$$

where T is the number of time steps

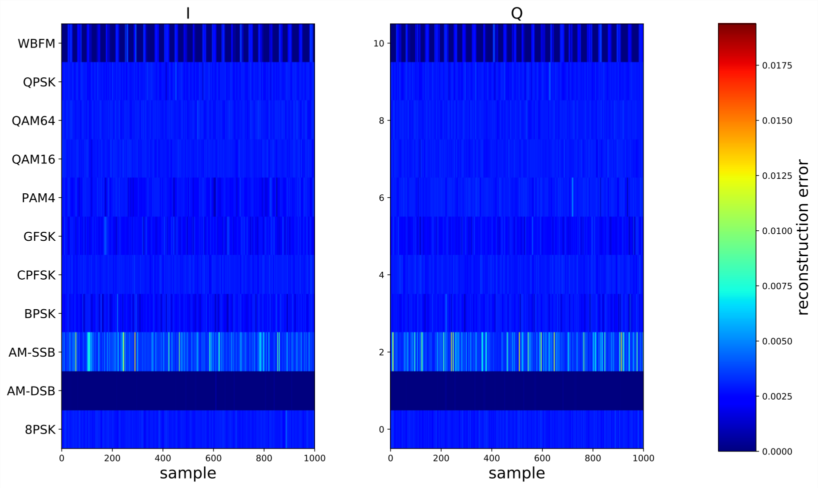
Delta Modulator



Delta Modulator



Delta Modulator



Reservoir computing

Reservoir computing

Requirements for the spiking neural network:

1. generate specific activation patterns for every baseband symbol
2. maintain a working memory of previous activation patterns

Reservoir computing

Reservoir computing can offer a solution to both requirements

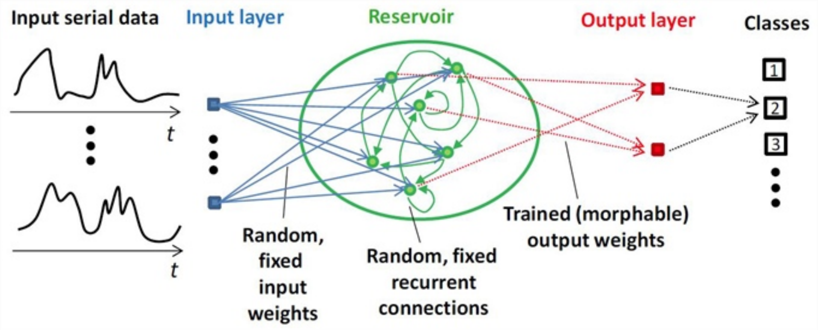


Figure 4: Reservoir computing architecture[1]

Reservoir computing

Input layer:
projects input $u(t)$ to the random units in the reservoir

Reservoir:
maps input $u(t)$ to a high-dimensional state $x(t)$

$$x(t) = f(W_{in}u(t) + W_{res}x(t - dt))$$

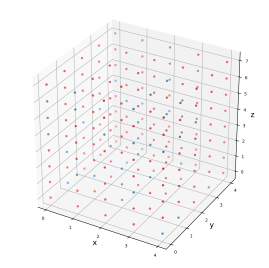
Readout layer:
learning occurs by adjusting the weights W_{out} of the readout layer

$$y(t) = W_{out}x(t)$$

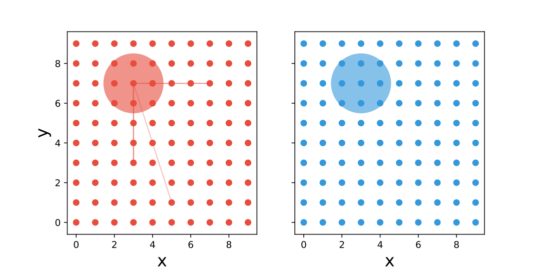
Neuromorphic implementation of a reservoir

- units: adaptive exponential integrate-and-fire neurons[2]
- dynamical system: network of $(1 - f) \cdot N$ excitatory neurons and $f \cdot N$ connected through synapses
- mismatch due to fabrication imperfections

Two connectivity models



(a) Schliebs



(b) Hennequin

Figure 5: Red dots represent excitatory neurons, whereas blue ones are inhibitory.

Reservoir computing

Schliebs model:

- probability of connection given by

$$p(A, B) \propto \Gamma(A, B) \exp\left(-\frac{d(A, B)^2}{2\lambda^2}\right),$$

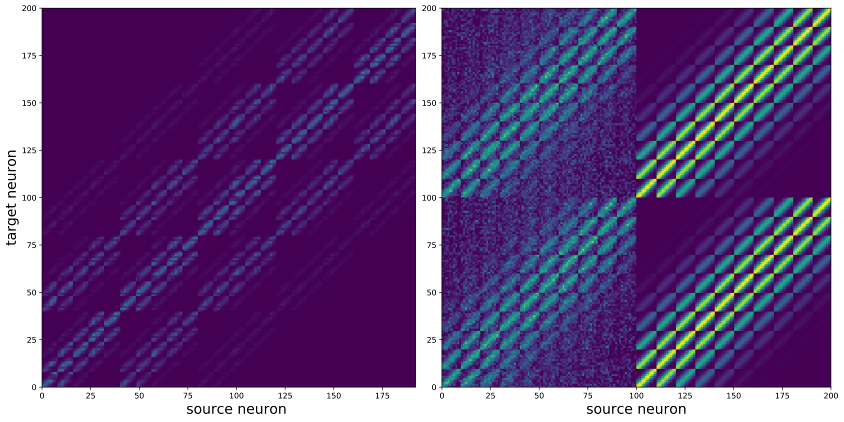
- probability amplitude $\Gamma(A, B)$ depends on type of the neurons involved

Hennequin model:

- probability of connection given by

$$p(A,B) \propto \; p_A^{local} \exp\left(-\frac{d(A,B)^2}{2\lambda^2}\right) + (1 - p_A^{local})\frac{1}{K}\sum_{i=1}^{K} \exp\left(-\frac{d(A,L_i)^2}{2\lambda^2}\right)$$

- patchy connectivity: $\begin{cases} \mathrm{p}_A^{local} = 0.5, \text{ A is excitatory} \\ \mathrm{p}_A^{local} = 1.0, \text{ A is inhibitory} \end{cases}$



(a) Schliebs

(b) Hennequin

Figure 6: Connection probability matrices for two different connectivity models of a reservoir with $N = 200$ neurons.

Reservoir computing

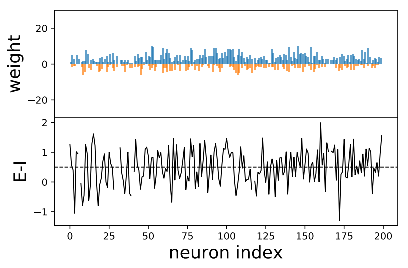
Weights:

$$W_{ij}^{E} \sim \mathcal{N}_E(\mu_E, \sigma_E)$$
$$W_{ij}^{E} \sim \mathcal{N}_I(\mu_I, \sigma_I)$$

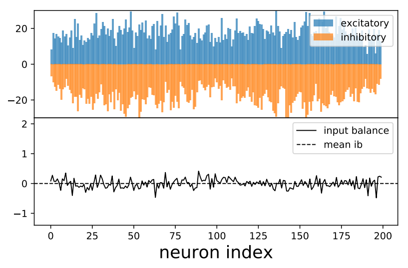
E-I balance:

$$\sum_j W_{ij}^{E} \approx \sum_j W_{ij}^{I} \forall i$$

- found in theoretical models and experimental data[6][4]
- underlies efficient coding of information
- achieved through excitatory and inhibitory synaptic plasticity



(a) Schliebs



(b) Hennequin

Figure 7: E-I balance for the two connectivity models.

Reservoir computing

Mismatch:

- caused by variance in transistor properties
- has been shown to improve energy-efficiency of ELMs[5]
- parametrized by variance η^2 of a Gaussian distribution centered on the bias value (e.g. I_τ)

Reservoir computing

Parameter tuning:

1. adapt neuron and synapse time constants to the temporal patterns in the input
2. adjust spike threshold and input weights until reservoir activity is detected
3. fine tune connectivity parameters (e.g. Γ, λ, ...) and reservoir weights distributions (e.g. μ_E, σ_E, ...)

Reservoir computing

Readout:

1. extract state vectors $x(t_k)$ from reservoir activity
2. train a linear classifier at each time step t_k
3. measure classification accuracy and feature importance

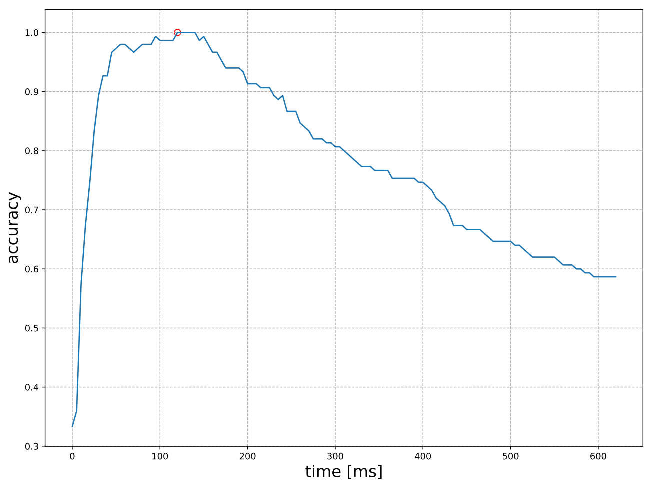
Results

Issues encountered:

- failed to tune the reservoir with Schliebs connectivity
- long duration of simulations even when compiled in C++:
  - $\sim 1h$ for 3 classes and 20 samples per class
- high variability of activity between samples of the same class
- too many parameters

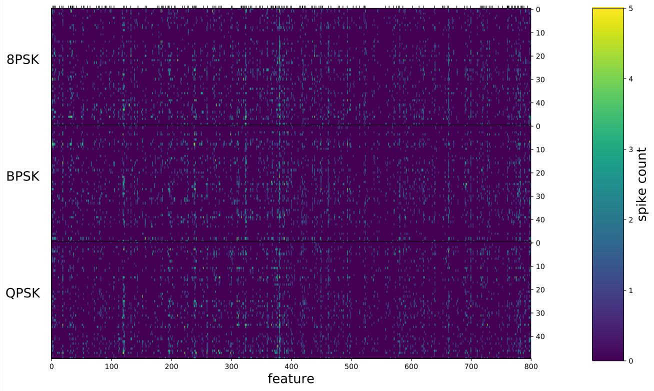
Results

Classification of 3 PSK modulations at an SNR of 18dB with 50 samples per class



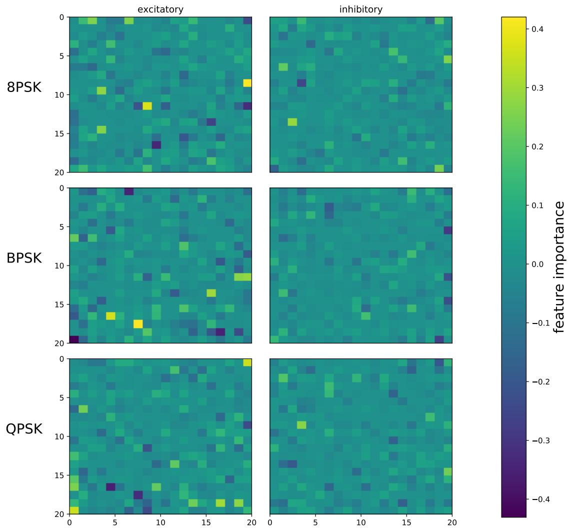
Results

Features at time of max. accuracy for all samples



Results

Feature importance at time of max. accuracy



Conclusions:

- RadioML is a difficult dataset for pattern recognition
- reservoir architecture needs further improvement
  - plasticity mechanisms
  - time delays
  - deep layers (?)

Thank you!

References i

D. J. Gauthier.
Reservoir computing: Harnessing a universal dynamical system, 2018.

G. Indiveri, B. Linares-Barranco, T. J. Hamilton, A. Van Schaik, R. Etienne-Cummings, T. Delbruck, S.-C. Liu, P. Dudek, P. Häfliger, S. Renaud, et al.
Neuromorphic silicon neuron circuits.
Frontiers in neuroscience, 5:73, 2011.

T. J. O'Shea, J. Corgan, and T. C. Clancy.
Convolutional radio modulation recognition networks.
In *International conference on engineering applications of neural networks*, pages 213–226. Springer, 2016.

References ii

Y. Shu, A. Hasenstaub, and D. A. McCormick.
Turning on and off recurrent balanced cortical activity.
Nature, 423(6937):288, 2003.

E. Yao, S. Hussain, A. Basu, and G.-B. Huang.
Computation using mismatch: Neuromorphic extreme learning machines.
In *2013 IEEE Biomedical Circuits and Systems Conference (BioCAS)*, pages 294–297. IEEE, 2013.

S. Zhou and Y. Yu.
Synaptic EI balance underlies efficient neural coding.
Frontiers in Neuroscience, 12:46, 2018.