BrainScaleS Workshop

4th HBP School

Korbinian Schreiber & Sebastian Billaudelle

June 13, 2017

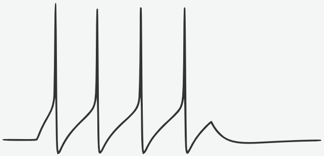
Kirchhoff-Institute for Physics, Heidelberg University

First Section

Analog Neuromorphic Hardware

1

observations



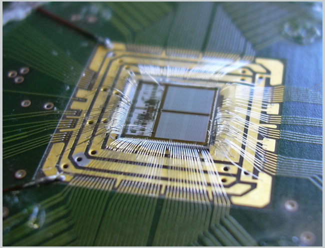
2

mathematical model

$$C\frac{\mathrm{d}V}{\mathrm{d}t} = -g_\mathrm{L}(V - E_\mathrm{L}) + I_\mathrm{syn}(t)$$

3

hardware realization



Roadmap

2004

Spikey

- single chip system
- 384 LIF neurons

2010

HICANN

- 180 nm CMOS
- 512 AdEx neurons

2015

20 Wafer System

- 4 million neurons
- 0.9 billion synapses

2017

HICANN DLS

- 65 nm CMOS
- PPU: integrated processing unit for advanced plasticity

2022

500 Wafer System

- 500 million neurons
- 130 billion synapses

PyNN API documentation

`https://neuralensemble.org/docs/PyNN/0.7/api/api-0.7.html`

Look out for:

- `pynn.Population`
- `pynn.Projection`
- `pynn.*Connector`

Creating (groups of) neurons

Create *populations* of neurons:

```
params = {
    "v_thresh": -60.0
    }
neurons = pynn.Population(42, pynn.IF_facets_hardware1, cellparams=params)
```

Get a list of default neuron parameters:

```
print pynn.IF_facets_hardware1.default_parameters
```

Generating stimuli

Create a stimulus from a spike train:

```
spike_train = np.arange(10.0, 101.0, 10.0)
stimulus = pynn.Population(1, pynn.SpikeSourceArray, {"spike_times": spike_train})
```

There is also a Poisson spike source:

```
poisson_params = {
    "start": 10.0,
    "duration": 100.0,
    "rate": 5.0
    }
stimulus = pynn.Population(1, pynn.SpikeSourcePoisson, poisson_params)
```

Synaptic connections

Recording observables

Spike times:

```
neurons.record()
...
spikes = neurons.getSpikes()
```

Analog membrane traces:

```
pynn.record_v(neurons[0], "")
```

- only *one* analog-to-digital converter (ADC)

→ one can record a single neuron at a time

Tasks

Task 1: a single neuron

Task 2: a simple chain