A Database for Neural Simulators

Pros & Cons

PRO

- Run Faster
- More Features
- Less Code

CON

- Incompatible with current software

Past & Present

- Object Oriented Programming
- Arrays of Structures (AoS)
  - Not Structures of Arrays (SoA)

The Future is Structures of Arrays

- Run Faster
- Enable Graphics Cards

Problems with SoA

- Existing software is AoS
  - You will need to rewrite your software to switch from AoS to SoA
- SoA is hard
  - Good tools will make the job easy

The Database

- Programmer describes their data structures
- Database implements them in an optimal way
- Database hides behind Object Oriented API
- Programmer can access raw data for speed

OOP Example

```
>>> class Neuron:
>>>     def __init__(self):
>>>         self.voltage = -70.
>>>
>>> my_neuron = Neuron()
>>> my_neuron.voltage
-70.
```

SoA Example

```
>>> db = Database()
>>> Neuron = db.make_class(“Neuron”)
>>> Neuron.add_attribute(“voltage”, initial_value=-70.)
>>>
>>> my_neuron = Neuron()
>>> my_neuron.voltage
-70.
```

Behind The Scenes

- my_neuron is a pair of:
  - Pointer to the database
  - Index of this neuron within the database

```
>>> array = db.get_array(“Neuron.voltage”)
>>> index = my_neuron.get_index()
>>> array[index]
-70.
```

Specialized Data Structures

- Sparse Matrix (Lists)
- Spatial Partitioning Tree (Nearest Neighbors)
- Grids of Objects
- Linear DiffEq (Diffusion & Passive Electrics)

Extra Features

- Automatic Error Checking
  - Check for NaN, NULL, valid range
- Documentation & Physical Units
- Save/Load from file
- Graphics Cards
- And more...