Reproducible neural network simulations: model validation on the level of network activity data

Robin Gutzen, Michael von Papen, Guido Trensch, Pietro Quaglio, Sonja Grün, Michael Denker

1 Institute of Neuroscience and Medicine (INM-6), Institute for Advanced Simulation (IAS-6), JARA-Institute Brain Structure-Function Relationship (INM-10), Jülich Research Centre, Germany
2 Simulation Lab Neuroscience, Institute for Advanced Simulation, JARA, Jülich Research Centre, Jülich, Germany
3 Theoretical Systems Neurobiology, RWTH Aachen University, Aachen, Germany

Concept

A model has the purpose to describe and predict its system of interest, which is a well defined entity selected for analysis. The model can be separated into two parts. The mathematical model is an abstract description formed by analysis and observation. The executable model is the implementation of the mathematical model which can perform simulations and thus generate testable predictions. [1][2]

Confirmation: Assesses the plausibility of modeling choices and premises.

Verification: Ensures that the implementation is a correct representation of the mathematical model, concerning the code and the calculations.

Validation: Establishes confidence in the model by testing whether its prediction accuracy is within an acceptable agreement to the system of interest.

Substantiation: Defined here as the evaluation and quantifying the level of agreement between two executable models.

In practice the substantiation performs equivalent tests as a validation and differs only in its interpretation.

Validation approach: We validate by evaluating the simulation outcome on the level of the network activity, as opposed to the complementary approach of validating on a single-cell level.

Model & Implementations

Mathematical Model

Polychronization network model. (Izhikevich [1])

800 exc. / 200 inh. spiking neurons
random connectivity, out-degree 100, random input,
integer delays, trained with STDP, measured without STDP

Custom C code: original publication [3], refactored,
adjusted to make it transferable to SpiNNaker,
1ms time steps

Validation results can guide model/implementation development.

Substantiation by Iterative Validation Test Cycles (I, II, III)

I) (i) uses an ESR ODE-solver, (ii) adapts Izhikevich’s neural dynamics algorithm, (iii) uses a more exact fixed step-size forward Euler ODE-solver.

→ Agreement of complex measures does not entail agreement of simpler measures.

II) uses finer integration steps and more precise detection of threshold crossing.

→ Multiple measures are needed for a reasonable and comprehensive validation.

III) improves the threshold crossing detection algorithm on SpiNNaker.

→ The evaluation of the level of agreement depends on the intended application. Here, despite good agreement of other measures, complex measures such as the pattern density (detected with SPADe [9]) is not yet consistent.

References


Python Validation Framework

developed at github.com/INM-6/NetworkUnit

I) We developed the module NetworkUnit as a formalized Validation test framework based on Elephant [7] and SciUnit [8] for validation and substantiation in experiment and simulation.

→ Example:

```python
from NetworkUnit import

# import networkunit esm
import scid

# class networkunit.model
class networkunit.model:

# input.parameters: ProduceInputParameterSet()

def __init__(self):
    # create simulation outcome
    ...

# class C_model(networkunit.model):
file_path = "validation_model/c_model"

# class SpiNNaker_model(networkunit.model):
file_path = "validation_model/siynnaker_model"

C = C_model()
S = SpiNNaker_model()

# class rate_test(scid.statistical_test, scid.metrics.rate_test)

def __init__(self):
    # initiate the test with a score

# def rate_test(score)

FR = rate_test()
FR.next_shape(S.C.I)
# performs the validation test

# Effect Size
detected = 0.800

Effect Size = 0.294
CI = (0.290, 0.463)
```

Acknowledgments:

This project has received funding from the European Union’s Horizon 2020 Framework Programme for Research and Innovation under Specific Grant Agreements No. 720270 (Human Brain Project SGA1) and 781907 (Human Brain Project SGA2).