Overcoming model limitations via empirically-tuned parameters

Alexandre René, André Longtin, Jakob H. Macke

Motivation

• Even the best neuron population models are still only approximations of biological processes.
• Statistical inference is already used to fit phenomenological models to data.
• We can compensate for approximations by extending statistical inference to mechanistic models.
• This allows us to combine the interpretability of mechanistic models with the predictive power of data-driven models.

Approach

Choose data and model with known mismatch:
• Synthetic data (microscopic)
• Derived model (mesoscopic)

Microscopic model

Excitation / Inhibition
Adaptation
Escape noise

Mesoscopic model

E1
L1
E2
L2

Simulate

Simulate

Infer

Use statistical inference to recover parameter point estimates...

...or their posterior distribution.

Inference can compensate for homogeneous approximation

Microscopic model

Infer all parameters

Average parameters

Mesoscopic model

Not all approximations can be compensated by inference

• Extreme inputs can break the quasi-renewal renewal approximation
• Inference can only work within model constraints
• More flexible models → more compensation possibilities

We can infer dozens of parameters simultaneously

16 connectivity parameters
16 membrane parameters
4 adaptation parameters

Method

Mesoscopic generalized integrate-and-fire model

Maximum likelihood inference

Likelihood function

- parameters: \( \theta \)
- current state: \( S(t) \)
- data \( \{y(t)\} \)
- model: \( P(y(t) | S(t-1), \theta) \)
- likelihood: \( L(\theta) = \prod P(y(t) | S(t-1), \theta) \)

Optimization via stochastic gradient ascent

Why use inference to determine parameters?

• Systematic: no hand-tuning of parameters
• Principled: recovered parameters maximize the likelihood
• Scalable: can infer dozens of parameters
• Flexible: model equations do not change
• Ready: code library available on github

Computational tools

Gradient ascent fits performed with ADAM\(^3\) algorithm, using TensorFlow for automatic differentiation and optimization. Posterior obtained via Hamiltonian Markov Chain Monte Carlo, using PyMC3\(^4\).

References


\(^1\) Max Planck Research Group Neural Systems Analysis, Center of Advanced European Studies and Research (ceaers), Bonn, Germany
\(^2\) Department of Physics, University of Ottawa, Canada;
\(^3\) Brain and Mind Research Institute, University of Ottawa, Canada
\(^4\) Computational Neuroengineering, Department of Electrical and Computer Engineering, Technical University Munich, Germany

Email: a.rene@uottawa.ca, a.longtin@uottawa.ca, macke@tum.de

\[^{4}\] M. Polson et al. Preprint (2019)