

Solving ODEs in Chemical Kinetics Using Artificial Neural Networks

Gabriel S. Gusmão[†], gusmaogabriels@gatech.edu

Zhenzi Yu[†], zyu331@gatech.edu

Nicole (Yuge) Hu[†], yugehu@gatech.edu

[†]Department of Chemical and Biomolecular Engineering

June 9th, 2019

Introduction: Ordinary differential equations (ODEs) and partial differential equations define numerous questions in the field of chemical engineering spanning from kinetics to transport phenomena. In this project, we would like to explore the possibility of using artificial neural networks (ANNs) to approach ODEs in chemical kinetics. The universal approximation theorem for ANNs¹ has been shown to extend to ODE solutions² and may be explored in the context of stiffness and scattered data.

Kinetics are defined as the rate of reactants converted to products based on stoichiometry. For a general 1st order reaction $A \xrightarrow{k_1} B$ where reactant A goes to product B, the rate of reaction can be expressed in an ODE: $\frac{dC_A}{dt} = -k_1 C_A$. More generally, a reaction network can be expressed as a linear combination of elementary reaction rates, as in eqn 1:

$$\frac{dC_i}{dt} = \sum_j \nu_{i,j} k_j \prod_{m \in S_j} C_m^{\nu_{j,m}} \quad (1)$$

where C_i is the concentration of species i , $\nu_{i,j}$ is stoichiometric coefficient of species i in reaction j , k_j is reaction j coefficient, m refers to other species involved in reaction.

Methods: We plan to use forward solutions of arbitrary chemical kinetic networks, as in eqn 1, as the input-output dataset to the proof-of-concept problem. Different from solving the ODEs, we are here interested in model selection, i.e. given a model candidate, use ANNs to find the parameter set of the model candidate that minimizes error metrics subjected to the underlying ODE. Once this framework is well-defined, the ensuing problem is to devise model generation and classification algorithms that span over a large set of potential model candidates and select those of higher likelihood in terms of a trade-off between a complexity metric and cost function minimization. The overall algorithm can be summarized as:

1. Generate arbitrary models according to eqn. 1.
2. Solve the forward ODE and add white noise to solution points.
3. Generate a set of potential model candidates.
4. Solve the reverse problem, i.e. solve the optimization problem $\min_{\theta_{ANN}, \theta_{model}} L(c(t), f(t, c(t), \theta_{ANN}, \theta_{model}))$.
Where L is some cost function, f is a model candidate, $\theta_{model} = \{k_j\}$ are the model parameters, θ_{ANN} comprise weights and biases of the ANN, $c(t)$ are observed values (solutions to the ODE) at time t . L may potentially include a regularization term, $R(\theta_{model})$, to account for non-bijection/uniqueness.
5. Classify models by generating a Pareto-front of efficient solutions.

Expected Results & Discussion: The greatest problem that we envision is to find expressions for the derivatives of L with respect to θ_{model} , which might be cumbersome to derive analytically given any arbitrary model. Therefore, we might opt for using automatic differentiation packages, such as `autograd`³.

We would like to use 1st order reactions as a proof of concept to validate our ANN framework, and then extend the application to higher order reactions to show that this approach is readily applicable to multiple types of chemical kinetic problems. Furthermore, we intend to explore the interplay between ANN and proposed model complexity (number of layers and nodes) with respect to overfitting and generalization.

¹Csáji, Balázs Csanád. "Approximation with artificial neural networks." Faculty of Sciences, Eötvös Loránd University, Hungary 24 (2001): 48.

²Kitchin Research Lab. "Solving ODEs with a neural network and autograd." <http://kitchingroup.cheme.cmu.edu/>

³Chen, Tian Qi, et al. "Neural ordinary differential equations." Advances in Neural Information Processing Systems. 2018.