

Parameter Calibration with Consensus-Based Optimization for Interaction Dynamics Driven by Neural Networks



Claudia Totzeck and Simone Göttlich

Abstract We calibrate parameters of neural networks that model forces in interaction dynamics with the help of the Consensus-based global optimization method (CBO). We state the general framework of interaction particle systems driven by neural networks and test the proposed method with a real dataset from the ESIMAS traffic experiment. The resulting forces are compared to well-known physical interaction forces. Moreover, we compare the performance of the proposed calibration process to the one in Göttlich and C. Totzeck (Optimal control for interaction particle systems driven by neural networks. arXiv:2101.12657, 2021) which uses a stochastic gradient descent algorithm.

1 Introduction

Modelling interacting particle dynamics such as traffic, crowd dynamics, schools of fish and flocks of birds has attracted the attention of many research groups in the recent decades. Most models use physically-inspired interaction forces resulting from potentials to capture the observed behaviour. In fact, the gradient of the potential is used as driving force for interacting particle systems formulated with the help of ordinary differential equation (ODE). These models are able to represent the main features of the dynamics, but as for all models we cannot be sure that they deliver the whole truth. The idea in [1] was therefore to replace the physical-inspired models by neural networks, train the networks with real data and compare the resulting forces.

C. Totzeck (✉)

School of Mathematics and Natural Sciences, University of Wuppertal, Wuppertal, Germany
e-mail: totzeck@uni-wuppertal.de

S. Göttlich

School of Business Informatics and Mathematics, University of Mannheim, Mannheim, Germany
e-mail: goettlich@uni-mannheim.de

In the recent years it became obvious that neural networks are able to represent a lot of details from the dataset. It may be possible that there are details captured that are not even noticed by humans and therefore do not appear in physical models which are built to reproduce observations of the modeller.

In the following we recall the general dynamic of interaction particle systems driven by neural networks as proposed in [1]. Then we shortly describe the global optimization method ‘Consensus-based optimization’ that we use for the real-data based calibration the network. Finally, we present the numerical results obtained by the calibration process and compare them to the ones resulting from the calibration with the stochastic gradient descent method reported in [1].

2 Interacting Particle Systems Driven by Neural Networks

We consider interacting particle dynamics described by ODE systems of the form

$$\frac{d}{dt}y_i = \sum_{j=1}^N W_{\theta}^{i,j}(y_j - y_i), \quad y_i(0) = z_0^i, \quad i = 1, \dots, N, \quad (1)$$

where $W_{\theta}^{i,j}$ represents the interaction force resulting for y_i in its interaction with y_j . The initial condition of the particles is given by real dataset $z_0 = z(0)$. In order to compare the results to the ones in [1] we restrict the class of neural networks to feed-forward networks. However, note that the approach discussed here allows for general neural networks while the discussion in [1] considers feed-forward networks and can only be generalized to neural networks allowing for back propagation.

2.1 Feed-Forward Neural Networks

In the following we consider feed-forward artificial neural networks of the form

Definition 1 A *feed-forward artificial neural network (NN)* is characterized by

– Input layer:

$$a_1^{(1)} = 1, \quad a_k^{(1)} = x_{k-1}, \quad \text{for } k \in \{2, \dots, n(1) + 1\},$$

where $x \in \mathbb{R}^{n(1)}$ is the input (feature) in (1) and $n(1)$ is the number of neurons without the bias unit a_1 .

– Hidden layers: for $\ell \in \{2, \dots, L - 1\}$, $k \in \{2, \dots, n^{(\ell)} + 1\}$

$$a_1^{(\ell)} = 1, \quad a_k^{(\ell)} = g^{(\ell)} \left(\sum_{j=1}^{n^{(\ell-1)}+1} \theta_{j,k}^{(\ell-1)} a_j^{(\ell-1)} \right).$$

– Output layer: $a_k^{(L)} = g^{(L)} \left(\sum_{j=1}^{n^{(L-1)}+1} \theta_{j,k}^{(L-1)} a_j^{(L-1)} \right)$ for $k \in \{1, \dots, n^{(L)}\}$

Note that the output layer has no bias unit. The entry $\theta_{j,k}^\ell$ of the weight matrix $\theta^{(\ell)} \in \mathbb{R}^{n^{(\ell-1)} \times n^{(\ell)}}$ describes the weight from neuron $a_j^{(\ell-1)}$ to the neuron $a_k^{(\ell)}$. For notational convenience, we assemble all entries $\theta_{j,k}^{(\ell)}$ in a vector \mathbb{R}^K with

$$K := n^{(1)} \cdot n^{(2)} + n^{(2)} \cdot n^{(3)} + \dots + n^{(L-1)} \cdot n^{(L)}.$$

For the numerical experiment we use $g^{(\ell)} = \log(1 + e^x)$ for $\ell = 2, \dots, N - 1$ and $g^{(L)}(x) = x$. For an illustration of the NN structure we refer to [1]. In the numerical section we consider an NN with $L = 3$, one input and 5 units in the hidden layer.

3 Parameter Calibration

We formulate the task of the parameter calibration as an optimization problem. Let $u \in \mathbb{R}^d$ denote the vector of parameters to be calibrated. This could be the weights of the neural network θ and some other parameters, as for example the average length L and the maximal speed v_{\max} of the cars which we will consider in the application. As we want the network to recover the forces hidden in the real data dynamics, we define the cost function for the parameter calibration as

$$J(y, u) = \frac{1}{2} \int_0^T \|y(t) - z(t)\|^2 dt + \frac{\delta}{2} |u - u_{\text{ref}}|^2, \quad (2)$$

where z denotes the trajectories of the cars obtained by the traffic experiment, and u_{ref} are reference values for the parameters. The parameter δ allows to balance the two terms in the cost functional. In case no reference values of the parameters are available, we set $\delta = 0$ in the numerical section.

3.1 Consensus-Based Optimization (CBO)

We solve the parameter calibration problem with the help of a Consensus-based optimization method [4] and choose the variant introduced in [5] which is tailored for high-dimensional problems involving the calibration of neural networks. The CBO dynamics is itself a stochastic interacting particle system with N_{CBO} agents given by stochastic differential equations (SDEs). The evolution of the agents is influenced by two terms. On the one hand, there is a deterministic term that aims to confine the positions of the agents at a weighted mean. On the other hand, there is a stochastic term that allows for exploration of the state space. The details are

$$du_t^i = -\lambda(u_t^i - v_f)dt + \sigma \text{diag}(u_t^i - v_f)dB_t^i, \quad i = 1, \dots, N_{\text{CBO}} \quad (3)$$

with drift and diffusion parameters $\lambda, \sigma > 0$, independent d -dimensional Brownian motions B_t^i and initial conditions u_0^i drawn uniformly from the parameter set of interest. A main role plays the weighed mean

$$v_f = \frac{1}{\sum_{i=1}^{N_{\text{CBO}}} e^{-J(u_i)}} \sum_{i=1}^{N_{\text{CBO}}} u_i e^{-\alpha J(u_i)}.$$

By its construction, agents with lower cost have more weight in the mean as the ones with higher cost. The parameter α allows to adjust this difference of the weights. For more information on the CBO method and its proof of convergence on the mean-field level we refer the interested reader to [2] and the references therein. As indicated by the notation above, the agents used in the CBO method are different realizations of parameter vectors that we consider for the calibration. For the numerical results NN4 we consider a neural network with 13 weights, i.e., $\theta \in \mathbb{R}^{13}$. Moreover, we assume the maximal speed v_{\max} as additional parameter. Hence, for fixed t we have for the i -th CBO agent $u_t^i \in \mathbb{R}^{14}$.

4 Numerical Results and Conclusion

For the calibration of the parameters we consider real data from the project ESIMAS [3]. As we want to compare the results to the well-known follow-the-leader model for traffic flow (LWR) we recall its details

$$\frac{d}{dt}y_i(t) = f\left(\frac{y_{i+1}(t) - y_i(t)}{L}\right), \quad i = 1, \dots, N-1, \quad (4a)$$

$$\frac{d}{dt}y_N(t) = v_{\max}. \quad (4b)$$

Here $f(\cdot)$ is either $v_{\max} \log(\cdot)$ or $v_{\max}(1 - 1/\cdot)$. To be prepared for a reasonable comparison, we consider for the neural network dynamics

$$\frac{d}{dt}y_i(t) = W_{\theta}^{i,i+1}(y_{i+1}(t) - y_j(t)), \quad i = 1, \dots, N-1, \quad (5a)$$

$$\frac{d}{dt}y_N(t) = v_{\max} \quad (5b)$$

supplemented with initial data $y(0) = z_0$. This leads to $u = (v_{\max}, \theta)$. To evaluate the models and compute the corresponding cost we solve all ODEs with an explicit Euler scheme. For details we refer to [1]. The number in the notation NN2, NN4 and NN10 corresponds to the number of nonbias neurons in the hidden layer.

4.1 Data Processing and Numerical Schemes

The data collection of the ESIMAS project contains vehicle data from 5 cameras that were placed in a 1km tunnel section on the German motorway A3 nearby Frankfurt/Main [3]. The data is processed in the exact same way as in [1]. Files with the processed data can be found online.¹

The SDE which represents the CBO scheme is solved with the scheme proposed in [5]. In particular, we set $dt = 0.05$, $\sigma_0 = 1$, $\lambda = 1$ and the maximal number of time steps to 100. The mini-batch size of the CBO scheme is 50 and we have 100 CBO agents in total. In each time step we update one randomly chosen mini-batch. The initial values are chosen as follows

$$v_{\max} \sim U([20, 40]), \quad L \sim U([0, 10]) \text{ and } \theta \sim U([-0.5, 0.5]^K)$$

4.2 Resulting Forces and Comparison

Figure 1 (left) shows the velocities resulting from the parameter calibration process. We find that the estimated velocities for the NN approaches are higher than the velocities of the LWR based models. The difference is most significant in data set 10. The plot on the right shows the average of the resulting forces for the different models. The forces of the NN approaches resemble linear approximations of the forces corresponding to the LWR models. The car length (L) appears only in the LWR models. Its optimized values for the different data sets are given in Table 1. We see that the lengths for the linear model are smaller than the ones in the logarithmic model. This is in agreement with the results obtained with stochastic gradient descent and shown in [1]. Finally, we summarize the cost values after parameter calibration in Table 2. The least values of every column are highlighted

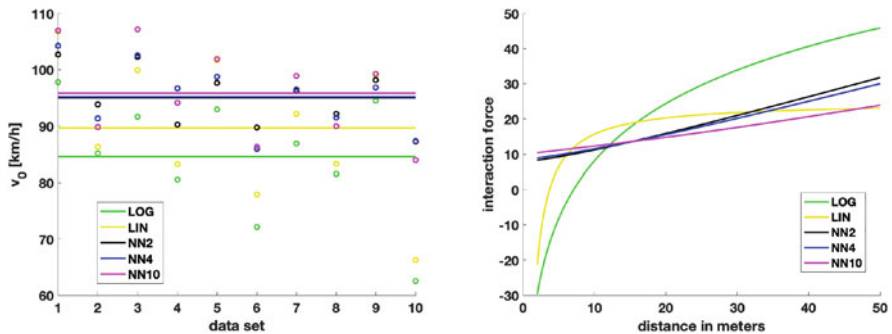


Fig. 1 Average velocities and forces resulting from the parameter calibration and learning process

¹ <https://github.com/ctotzeck/NN-interaction>.

Table 1 Car lengths (in m) estimated with the algorithm for the 10 data sets with the LWR-model with linear and logarithmic velocity

	1	2	3	4	5	6	7	8	9	10	average
Lin	3.5969	3.76	4.17	2.19	3.02	2.81	5.92	5.86	2.14	3.65	3.71
Log	7.15	7.21	8.05	8.17	6.19	5.00	8.10	8.46	5.63	6.91	7.09

Table 2 Values of the cost functional estimated with the algorithm for the 10 data sets with the LWR-model with linear and logarithmic velocity and the three different neural network approaches

	1	2	3	4	5	6	7	8	9	10	average
NN2	47.95	46.49	98.07	44.97	23.69	29.72	40.69	55.75	11.50	68.91	46.77
NN4	47.82	46.09	97.01	51.84	23.33	26.71	41.60	55.29	11.16	67.60	46.84
NN10	47.90	45.78	99.20	42.50	22.16	24.40	41.18	56.68	10.01	66.01	45.58
Lin	44.41	41.29	93.73	30.86	19.00	37.98	38.00	56.40	8.18	46.24	41.61
Log	53.53	50.31	109.36	65.24	26.50	52.93	38.09	58.22	14.54	52.75	52.15

in bold. It is obvious that the LWR model with linear force outperforms the other models. The results of the NN approaches are better than the ones of the LWR model with logarithmic force.

4.2.1 Comparison to Calibration with Stochastic Gradient Descent

In comparison to the parameter calibration based on the stochastic gradient descent method reported in [1], we find that the CBO approach finds better parameters for both LWR models. In fact, the resulting cost values are significantly smaller after the calibration with CBO. For the NN approaches the results are in good agreement. A clear decision in favour of the LWR approach or the NN ansatz was not possible based on the results of [1]. After the training with CBO the LWR with linear force seems to outperform all other approaches. We used NN with very simple structure here, it may be worth to test more sophisticated network structures in future work.

References

1. S. Göttlich and C. Totzeck, *Parameter calibration with stochastic gradient descent for interacting particle systems driven by neural networks*, Mathematics of Control, Signals, and Systems, online first, 2021.
2. C. Totzeck, *Trends in consensus-based optimization*, Active Particles, Volume 3, Springer International Publishing, 2022.
3. E. Kallo, A. Fazekas, S. Lamberty, and M. Oeser, *Microscopic traffic data obtained from videos recorded on a German motorway*, Mendeley Data, 2019. <https://doi.org/10.17632/tzckcsrpn6.1>
4. R. Pinnau, C. Totzeck, O. Tse, and S. Martin, *A consensus-based model for global optimization and its mean-field limit*, Math. Mod. Meth. Appl. Sci. **27**(1) (2017), 183–204
5. J. A. Carrillo, S. Jin, L. Li, and Y. Zhu, *A consensus-based global optimization method for high dimensional machine learning problems*, ESAIM: COCV **27** (2021) S5