COMPARISON OF DATA MODELLING POSSIBILITIES IN HYBRID MODELS

Stefanie Winkler

E101 - Institute of Analysis and Scientific Computing

MOTIVATION AND INTRODUCTION

The accumulation of data in various applications, platforms and software enforces researchers to look for powerful methods to process big data structures which makes data modelling one of the leading research areas. Despite this trend there are several research fields where system behaviour is still described using physical laws and mathematical formulations. Psichogios and Ungar [1] give an example of a possible combination of both approaches. Complex structures often require different methods and models, e.g. a mixture of discrete and continuous descriptions called hybrid models. Considering a model containing discrete and continuous processes where a neuronal network is used to determine these parameters. For example in automotive industry this combination was implemented to optimize a combination of different LTI systems, see Lu et al. [2]. In the following we will present the basic structure of hybrid models and neural networks and the possibilities to combine these approaches implementing a basic example.

MODELLING APPROACH

Hybrid Modelling. In modelling and simulation the term hybrid describes a certain collection of modelling approaches to describe the system behaviour. A hybrid automaton is a possible illustration of such complex behaviours consisting of different nodes symbolising various states of the system. Speaking of a hybrid dynamical system the nodes characterise the discrete or continuous system behaviour in form of ODEs, DAEs, DEVS or similar. The lines connecting these nodes define the conditions to enable transitions from one node to the other.

Neural Networks. In general neural networks are based on the biological nerve structure of human brains. The basic structure consists of three components: the input, the hidden and the output layers. The most interesting one is the hidden layer which contains a specific activation function to process the incoming signal. The weights of the edges can only be specified using training data consisting of input and corresponding output data as references.

RESULTS AND DISCUSSION

In the following the bouncing ball, an academic example of hybrid systems, is discussed. Regarding the bouncing ball the bounce itself represents the discrete part of the model and only occurs for a single point in time where the ball changes its direction and decreases its total velocity. This behaviour can be described with physical basics defining the corresponding ODE system and the state space description, respectively. The hybrid model description then consists of the state space description, the jump and the jump condition. Due to the fact that there exists an analytical solution to this problem training data for the neural network can be generated.

In the current setting the results of the neural network can be used to predict the next event of the hybrid system. Additionally the peak of the current bounce is calculated. Compared to the numerical simulation in MATLAB, where the behaviour of the system is given for every point in time, the

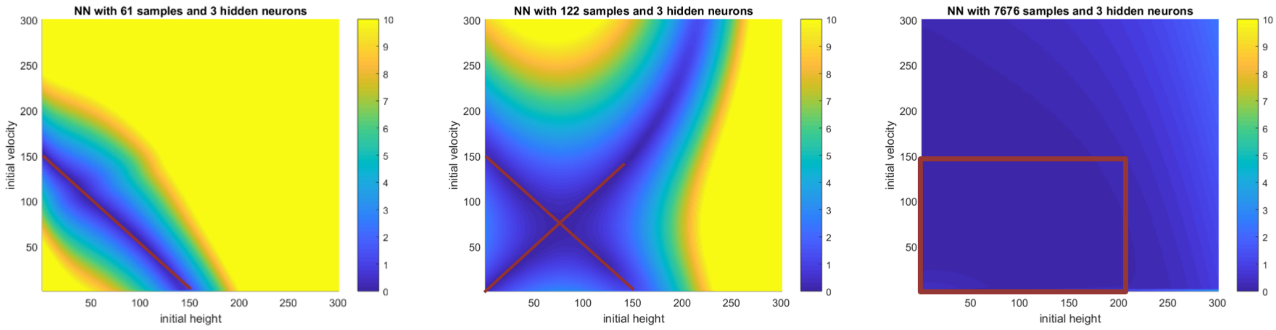

Figure 1: Depending on the number of data points in the training set the accuracy of the neural network is shown.

enabled analysis for the neural network is limited. Another possible implementation could predict only the height of the ball for the next time steps. This might be useful if such neural network would be embedded in a bigger real time system to calculate approximations to determine e.g. machine settings.

K. Hornik [3] claims that feed-forward networks are capable of arbitrarily accurate approximation to any real-valued continuous function over a compact set. Therefore it is not surprising that the accuracy of the neural network compared to the analytical solution is acceptable, as shown in Figure 1. The influence of the network structure and training set to the performance is here depicted. The results encourage the usage of neural networks for hybrid models.

OUTLOOK

This work was just a first step in answering the question if neural networks and hybrid modelling can be combined or even be used as replacement. The approach of Martinus and Lambert [4] could be useful to cope the challenging zeno effect typical for hybrid models. In order to explore the possibility to predict system behaviour, the training data has to be defined differently. Following this, the gathered knowledge will be applied to a more sophisticated example to analyse the system behaviour and provide possible answers to some of the following questions: What model structures can be replaced by neural networks? Which type of hybrid models benefit from embedded neural networks? What are the limits of this approach in hybrid modelling? The last question can never be fully answered but it is important to look for new enhancements and combinations of existing model approaches to keep pace with development and technology.

REFERENCES

[1] D. C. Psichogios and L. H. Ungar. *A Hybrid Neural Network-First Principles Approach to Process Modeling* AIChE Journal vol. 38, No. 10, pages 1499-1511, 1992.

[2] W. Lu, P. Zhu und S. Ferrari. *A Hybrid-Adaptive Dynamic Programming Approach for the Model-Free Control of Nonlinear Switched Systems.* IEEE Transactions on Automatic Control vol. 61, No. 10, pages 3203-3208, 2016.

[3] Hornik K., *Multilayer Feedforward Networks are Universal Approximators.* In: Neural Networks, vol 2, pp.359-366. 1989.

[4] Martinus G., Lampert C.H., *Extrapolation and learning equations.* 2016. arXiv preprint https://arxiv.org/abs/1610.02995.