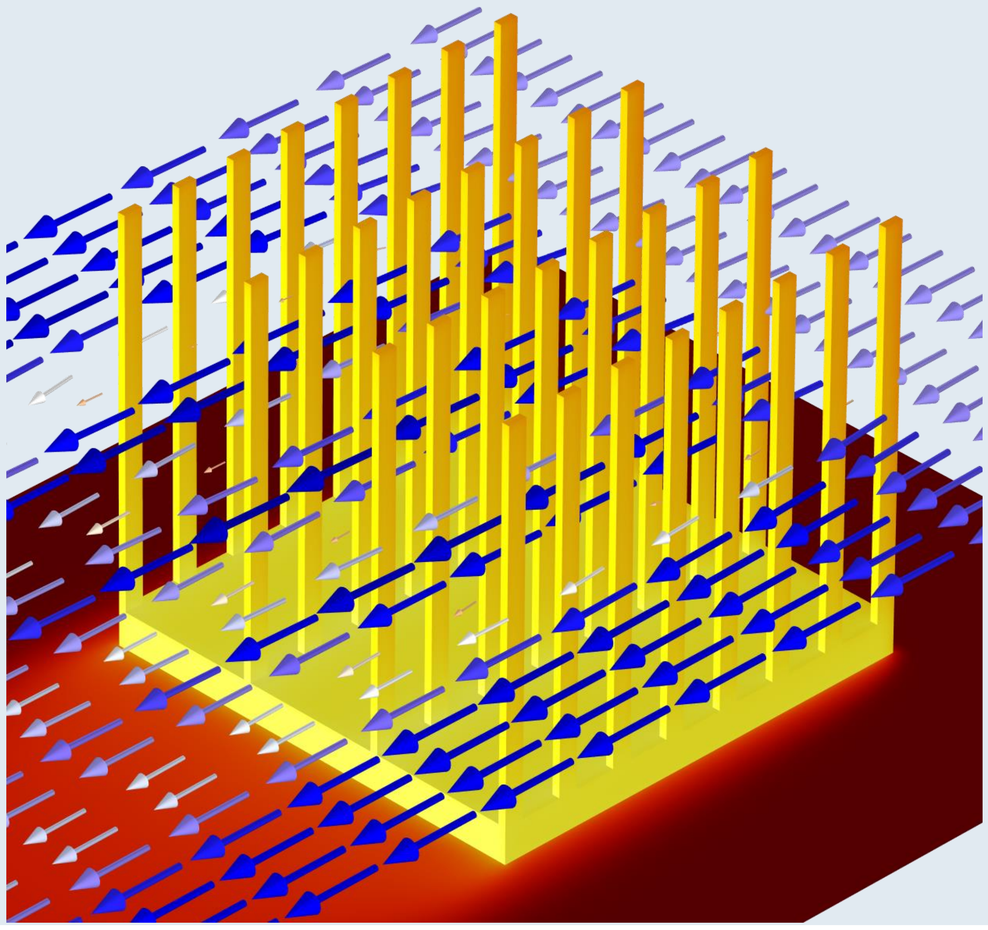

3D Study of TEG-Based Energy Harvesters with Physics-Informed Neural Networks

Two PINNs are compared with respect to their prediction accuracy and computing time advantage over FEM.

E. Vambolt[1], N. Pöpel[1], L. Lowe[1], L. Fromme[2], E. Wilczok[1], J. Lohbreier[1]

1. Faculty of Applied Mathematics, Physics and Humanities, Technische Hochschule Nürnberg Georg Simon Ohm, Nürnberg, Germany
2. Institute for Technical Energy Systems (ITES), Hochschule Bielefeld – University of Applied Sciences and Arts (HSBI), Bielefeld, Germany

Introduction

A fully-parameterized 3D model of a thermoelectric generator (TEG)-based energy harvester is built. With the use of a coupled CFD and heat transfer FEM simulation, the temperature field within the complete system (heat source, cooler, ambient air) is computed. Especially, the thermal resistance of the cooler is important to match the TEGs thermal resistance so that the harvestable energy is maximized. Such an extensive numerical simulation takes several minutes to hours until the desired data (e.g. the harvested power) is available for one set of input parameters (e.g. design of the cooler or properties of the fluid flow). To shorten this process, data sets are first generated with the FEM simulation and subsequently used to train deep neural networks (DNNs); they are able to process their input in real time. Finally, two implementations of this PINNs are compared with respect to their prediction accuracy.

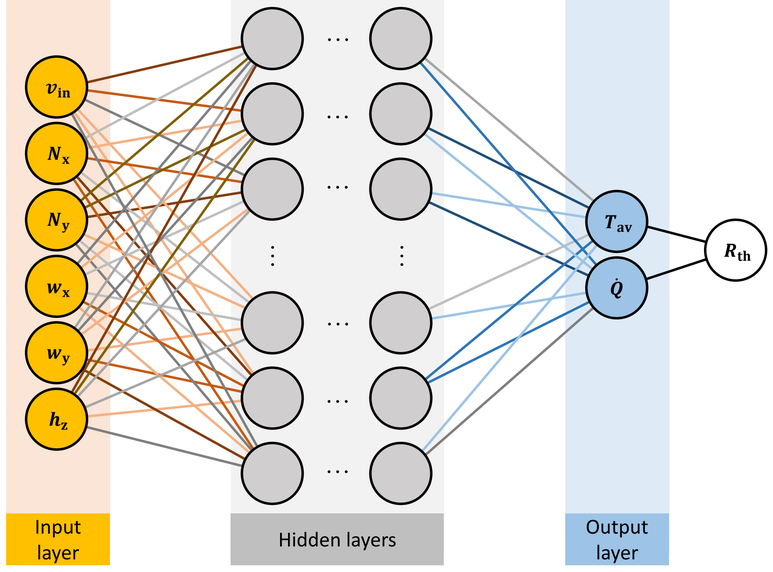

FIGURE 1. Architecture of the DNN. In this proof of concept the network has 3 hidden layers with 10 neurons each.

PINNs

A Latin hypercube algorithm is employed to generate ideal sample data for the training of two PINNs: One neural network is directly implemented in COMSOL and a second one is programmed in Python with the PyTorch framework. Both DNNs feature the same properties displayed in Figure 1: Six input parameters (inflow velocity, number of cooling fins in x- and y-direction, width of the cooling fins in x- and y-direction and the height of the fins in z-direction) are varied for the training data set. The output parameters are the average surface temperature (in K) of the cooler and the integrated heat flux (in W). With these two quantities, the thermal resistance of the specific configuration can be computed. The data used for the training as well as the hyperparameters like batch size, learning rate, number of hidden layers, neurons per hidden layers and the activation function are identical.

Results & Outlook

Using comparably little data (< 1000 data sets) both PINNs are able to predict the relevant output parameters with acceptable accuracy (indicated by the validation loss in Figure 2). While the training losses of the two DNNs are comparable (light and dark green curves), the validation loss of the Python network (light blue) is lower. This means that both PINNs are able to memorize known data but the Python DNN is better at responding to unseen data with correct predictions. Of course, larger training data would help to further increase the accuracy and thus the applicability of the PINNs. Then, the full-size FEM studies could be replaced by the much faster neural networks within a certain regime of parameters. In the future, a hyperparameter optimization will be used to investigate whether further increases in DNN performance are possible with the same amount of training data.

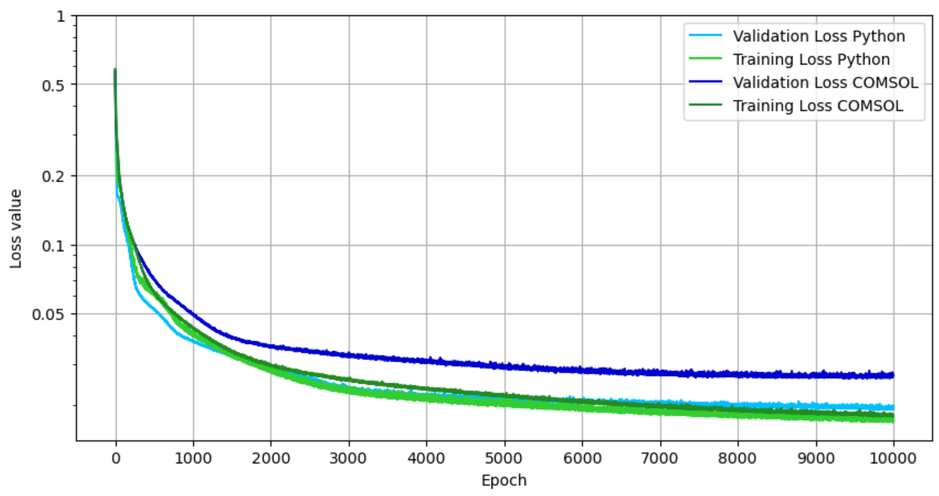

FIGURE 2. Loss value comparison of the two PINNs: One can see the slightly better performance of the PyTorch-based DNN.

ACKNOWLEDGMENTS

This work is supported by the LoLiPoP-IoT Project (www.lolipop-iot.eu), which is jointly funded by the European Commission and national funding agencies under the KDT joint undertaking.

Technische Hochschule Nürnberg

Hochschule Bielefeld
University of Applied Sciences and Arts