Deep Residual Learning applied to real-gas thermodynamics

Yipeng Ge[1,a)], Maximilian Hansinger[1], Christoph Traxinger[1] and Michael Pfitzner[1]

[1]*Institute for Thermodynamics, Bundeswehr University Munich, Werner-Heisenberg-Weg 39, 85577 Neubiberg, Germany*

[a)]Corresponding author: edison.ge@unibw.de

Abstract. Deep learning has greatly changed the landscape of computer vision and machine learning. It is generally known as supervised learning with back propagation, which is very efficient for regression and classification problems. In this work, deep learning with residual mapping has been introduced into the modeling of thermodynamic properties in real–gas flows. Using a data set generated by solving detailed physics for a single component supercritical jet, the modeling capability of deep residual learning for thermodynamic properties has been demonstrated. Furthermore, its application in a CFD simulation showed a promising 166x speed up compared to the reference model.

Introduction

In many propulsion and energy systems, e.g. gas turbines and liquid rocket engines (LREs), injection, mixing and combustion takes places at elevated pressure often exceeding the critical value of the pure components, i.e. $p > p_\mathrm{c}$. As a result, real-gas effects are a prominent feature in such processes and therefore fluid properties, e.g. specific heat c_p, strongly depend on both temperature as well as pressure and the ideal gas law is not valid. Especially in LREs the real-gas effects are strongly pronounced because at least one of the propellants is injected at cryogenic temperatures. During the last decades different groups have investigated the injection and subsequent combustion process in LREs experimentally, see, e.g., [1] and [2], as well as numerically, see, e.g., [3] and [4]. The numerical investigation is computationally expensive. It is state-of-the-art to apply a thermodynamic framework based on the cubic equation of state (EoS) or to use a-priori tabulation of the relevant thermophysical quantities. The approach based on the EoS is often preferred as it is a very good compromise between accuracy and efficiency. In contrast, tabulation is an efficient tool, however, depending on the problem studied, an accurate tabulation library can take a large amount of storage and high dimensional input tables are intractable in simulations.

A promising candidate to overcome the limitations in computational cost, accuracy and dimensions are artificial neural networks (ANNs). This approach is inspired by the biological neural network that constitutes the brain of beings. ANNs are already used in different economic and scientific fields ranging from speech recognition to stock market prediction or early stage cancer detection. In the field of thermodynamics, where this publication is aiming on, different groups, see, e.g., [5, 6, 7], have applied ANNs to predict vapour liquid equilibria (VLEs) of binary systems. All these groups are using a multi-layer perceptron (MLP) ANN with one hidden layer in combination with the back propagation algorithm to properly train the network. The groups report good agreement between experimental data and the estimations of the trained network. Additionally, Mohanty [6] and Lashkarbolooki et al. [7] report a superior performance of the applied ANN compared to commonly used cubic EoS.

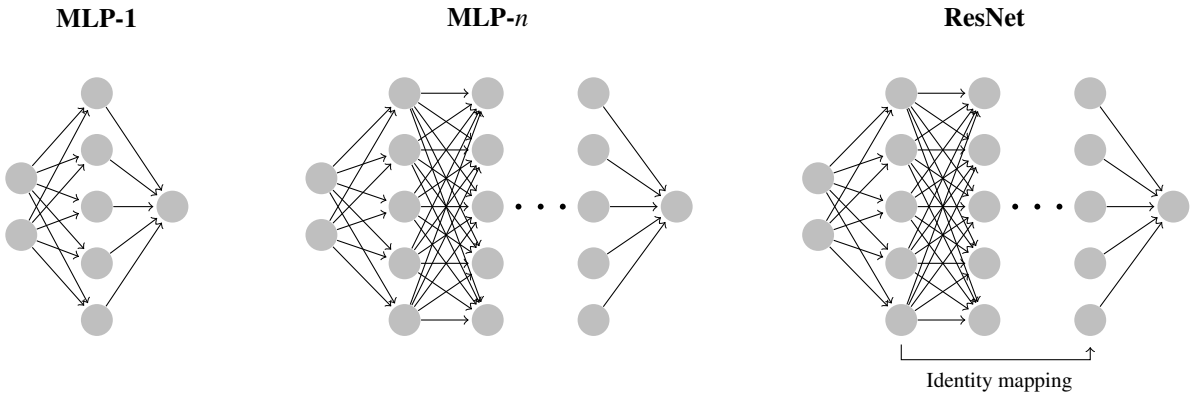
In the present study we are applying different ANNs to estimate real-gas thermodynamic properties necessary for the numerical CFD investigation of LREs. Our final goal is the speed-up of the numerical simulations with zero loss in accuracy and generality. In total, we will compare three different ANN concepts, namely a one layer MLP, a five layer MLP and the recently emerging residual network. In order to get some meaningful boundary conditions for the thermodynamic states typically occurring in LREs, we will use the widely studied Mayer test cases [8] where cryogenic nitrogen is injected into warm N_2 at supercritical pressure and quantify the time saving within the numerical simulation by applying an ANN instead of a cubic EoS.

Deep Residual Learning

In numerical simulations of supercritical injection, the update of the thermodynamic quantities, i.e. the calculation of the primitive variables from the conservative ones, is done at every iteration step. As we are using a pressure-based version of OpenFOAM, for more details see, e.g., [9], the solver provides a solution for the enthalpy h and the pressure p. In the single-phase state this information is sufficient to recalculate all other thermodynamic properties leading to the following general relation between primitive and conserved variables:

$$[T, \rho, c_p, a_s, \psi, \kappa, \mu]^T = \mathcal{F}[h, p]^T \ . \tag{1}$$

Here, T, ρ, c_p, a_s, ψ, κ and μ and are the temperature, the density, the specific heat, the speed of sound, the compressibility, the thermal conductivity and the viscosity, respectively. By applying an ANN this recalculation of the primitive variables could be done with a single and highly efficient interface. Basically, a number of different ANNs is available in the literature and it is always a trial and error procedure to find the best network for the particular purpose. Figure 1 gives an overview of the three different neural networks employed in the present study. Especially, the MLP-1 is a very well established ANN. For a more detailed overview on neural networks and residual networks see, e.g., [10, 11].



(a) MLP-1: One hidden layer between the in- and the output.

(b) MLP-n: Deep neural network with n stacked hidden layers between the in- and the output.

(c) ResNet: One block with two hidden layers and bypassed identity mapping between the in- and the output.

FIGURE 1: Overview of the different employed ANN topologies.

Since 2012, Deep Learning has gained great momentum and changed the landscape of machine learning [10]. It proves to be a powerful tool in solving regression and classification problems. By stacking multiple layers of 'neurons', deep neural network provide a universal function emulator. For recent advances in the field of deep neural–networks the reader is referred to, e.g., [12]. Taking Eq. (1) as the target, we propose a deep learning model based on residual learning [11], which is capable of accurately predicting all seven targets. Neural networks integrate all level of 'features' in an end to end fashion, and the stacked layers of the network (depth) enriches such integration. In image classification it has been shown that adding more (deep) layers to the network is beneficial for the prediction accuracy, when the network is applied to more complex data sets [13]. However, adding more layers can lead to higher training error and accuracy gets saturated and then degrades rapidly. To overcome this, we have adopted the deep residual network [11] in this work. In residual learning, the learning task has been reformulated into a residual representation. $H(x)$ is an underlying mapping to be fit by two stacked layers, with x denoting the inputs to the first of these layers, in our case h and p. Instead of learning $H(x)$, the stacked layers approximate a residual function $f(x) = H(x) - x$. The original mapping has been recast into $f(x) + x$. This formulation is realized by feed-forward neural network with 'identity mapping' which skips one or more layers. A building block is defined as:

$$y = f(x, W_i) + x \ . \tag{2}$$

Here, x and y are the input and output vectors of the layers considered. The function $f(x, W_i)$ represents the residual mapping to be learned.

Network Structure and Learning

To train our networks in the supervised learning process we use a data base generated with respect to the Mayer test cases [8] where cryogenic nitrogen is injected into itself at supercirtical pressure. For the generation of the data base there is basically no restriction in terms of EoS like it is usually the case when the thermodynamics is evaluated with EoSs during run-time. Therefore, we are not restricted to cubic EoS and could also use high fidelity Helmholtz EoS. For the present investigation, we decided to generate our data with a state-of-the-art framework based on the cubic EoS of Peng and Robinson [14] in order to give a meaningful comparison between previous investigations and the possible speed-up in the numerical simulation by applying a ANN. For each discrete pressure level between $p = 34$ bar and $p = 50$ bar we obtained 2000 data points to have a well resolved data base and a sufficient pressure and temperature range in order to use the trained ANN later in the numerical simulations. The data set generated contains the input $[h,\ p]^T$ and target values $[T, \rho, c_p, a_s, \psi, \kappa, \mu]^T$, resulting in a total of 32000 data points. As it is common practice in machine learning, each value in the data set has been normalized and scaled between 0 and 1, to improve the training process. The complete data set is then split up into two subsets, 70% is for the training set and 30% is withheld for the test set. The splitting is important, as the accuracy of the network has to be tested with data it has not 'seen' during the training process. In the training process input and target values from the training data set are presented to the network and it should 'learn' from the data. In this optimization problem the optimal weights and biases of the neurons are sought in order to minimize an objective function, also called loss function; in this case the loss is the sum of the mean square errors (MSE) between predicted target from the network (y_{pred}) and training set target values (y_{train}):

$$MSE = \sum_{i}^{N_s} \sum_{j}^{N} (y_{train_{i,j}} - y_{pred_{i,j}})^2 \tag{3}$$

with N_s as the number of test samples and N the number of targets. Optimization in the training process is achieved with the backward propagation algorithm in combination with the stochastic gradient descent (SGD) method [15]. For further evaluation of the model accuracy the correlation coefficient (R^2) is used:

$$R^2 = 1 - \frac{\sum_i^{N_s} \sum_j^{N} (y_{train_{i,j}} - y_{pred_{i,j}})^2}{\sum_i^{N_s} \sum_j^{N} (y_{train_{i,j}} - \overline{y}_{train_j})^2} \tag{4}$$

For a more comprehensive view on practical machine learning and ANNs the reader is referred to, e.g., [16, 17, 15].

Results

In the following the results from different network architectures and training sets are presented, namely the simple one layer MLP (MLP-1), a multi-layer deep neural network with five hidden layers (MLP-5), and the residual network (ResNet). All networks have been created and trained in python with the Keras [18] and CNTK [19] libraries. The networks' performance can be tuned via hyperparameters, such as the number of neurons in a given layer and the number of hidden layers or blocks in case of MLP-5 and ResNet, respectively. For the training process additionally the batch size - the number of samples that is going to be propagated through the network - and the percentage of training data withheld for cross-validation can be specified; the latter one is set to 10% throughout all training phases. The batch size was set to 5000 samples.
Table 1 gives an overview of the three different network architectures and a favourable set of hyperparameters, which have been chosen based on experience, to obtain a low MSE (or R^2 close to one) for the given architecture. Based

TABLE 1: Network architectures and test errors.

Network	Neurons	Layers	MSE	R^2
MLP-1	600	1	1.8e−4	0.9961
MLP-5	600	5	2.8e−5	0.9994
ResNet	600	5	1.5e−6	0.9999

on the overall R^2 value, see Tab. 1, all three networks achieve a quite high accuracy (> 99%) and differ only by some decimals. The trend is that a more complex network results in a higher accuracy. For the different thermodynamic properties this overall trend can also be observed, but the achievable R^2 value for each property significantly differs among the different networks. Figure 2 shows a direct comparison between the test data and the targets for the density ρ (1st row) and specific hear c_p (2nd row) as a function of the temperature T at p = 40 bar with the target specific R^2 values achieved in the training. All three networks are able to predict almost perfectly the density in the temperature range between 90 K and 350 K. In addition, the steep drop of density at the pseudo-boiling temperature T_pb is predicted with almost perfect accuracy in all three networks which is an important and challenging feature when simulating supercritical injection processes with cryogenic propellants. However, when it comes to predict the specific heat capacity c_p, see Fig. 2 2nd row, only the ResNet is able to predict the target values with reasonable accuracy. The two MLPs (MLP-1 and MLP-5) can basically represent the overall pattern, but the response of these two networks is not satisfactory and would lead to problems in the numerical simulation as the conservation of basic thermodynamic relations is important for accurate and stable computations in the real-gas state region. Again the comparison of the results of the different predictions shows that for our field of application a more complex and deeper neural network leads to more accurate predictions. An identical conclusion can be drawn when looking at the results for the other thermodynamic properties which are taken into account in the present regression problem, see Eq. (1).

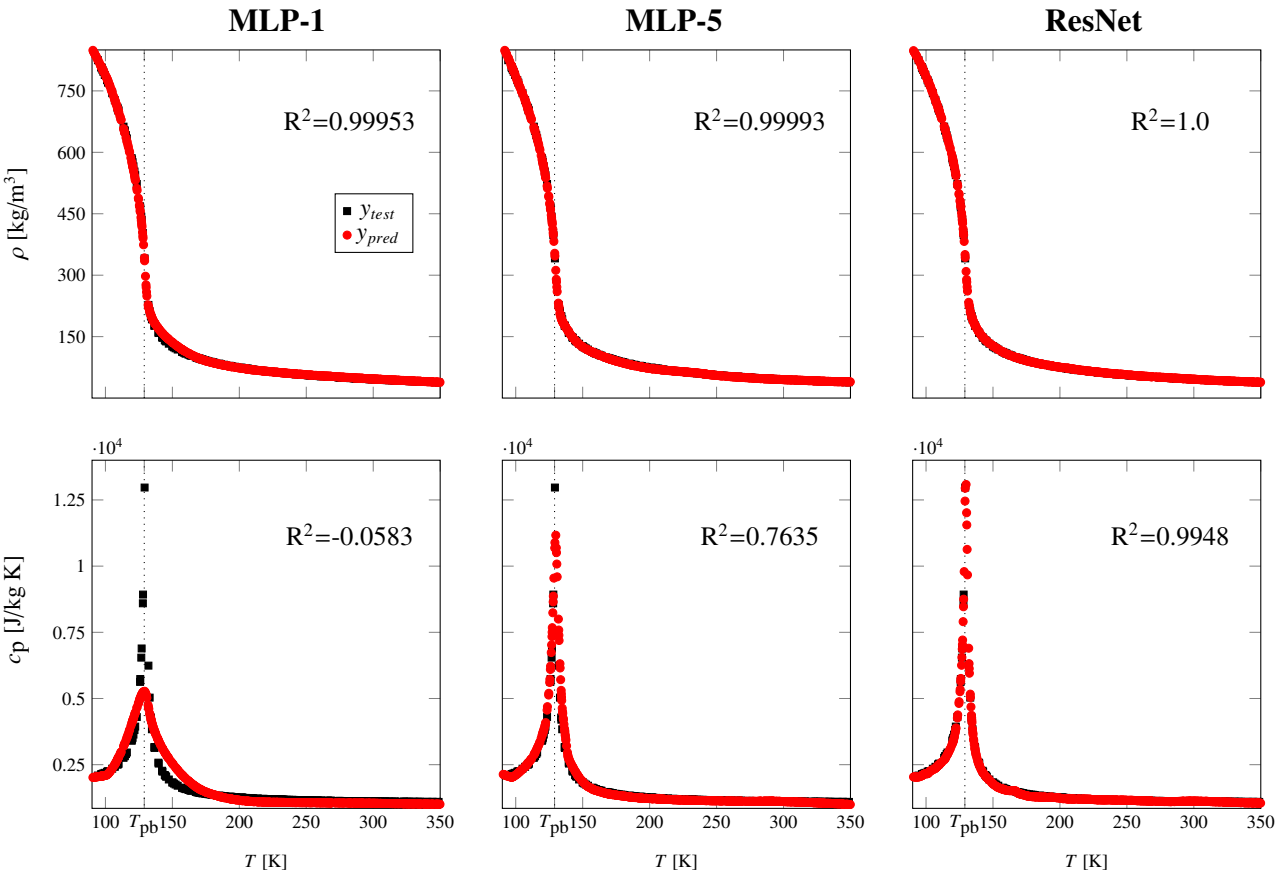


FIGURE 2: Comparison of test data and ANN predictions for ρ and c_p.

Based on the analysis of the three different networks (MLP-1, MLP-5 and ResNet) we selected the ResNet as our network of choice. In order to evaluate the performance and therefore the desired speed-gain within the numerical simulations we used a 2D-LES test case based on the Mayer test case number 3 where nitrogen is injected at a temperature of 126.9 K into it self at a pressure of approximately 40 bar. Table 2 gives an overview on the gained run-time simulation speed-up through the ANN regression. The benchmark was done at a time when the jet was already well developed, showing a wide range of different thermodynamic states. The computation times are averaged over

TABLE 2: Average computation time for real–gas EoS model(CPU) and ResNet inference (CPU and GPU).

Method	computational time [s]	speed-up factor
EoS(CPU)	0.0756	-
ResNet(CPU)	0.455e-3	166
ResNet(GPU)	0.450e-3	168

200 time steps in total. Three evaluation procedures are compared to each other, namely the state-of-the-art real-gas framework based on the EoS evaluated on the CPU and the ResNet inference evaluated on both CPU and GPU. The gained speed-ups are a factor of 166 for regression on the CPU and 168 for the GPU, respectively.

Conclusion and Outlook

Artificial neural networks (ANNs) are getting increasing attention and applications nowadays. In the field of thermodynamics different groups, see, e.g., [5, 6, 7], have applied simple multi-layer perceptron (MLP) with one hidden layer (MLP-1) to predict vapour liquid equilibria of binary systems. In the present investigation we are going beyond this scope with respect to the ANNs and compare three different ANN architectures ranging from MLP-1 to deep neural networks. In the analysis of the three different ANNs the residual neural network structure (ResNet) has been identified as the most promising candidate to perform the regression task without almost any loss in accuracy and other a wide range of different thermodynamic properties and states. The gained speed-up in a CFD simulation through ANN regression compared to state-of-the-art frameworks based on the cubic equation of state is found to be a factor of 166. In future work we are planning as a first step to more accurately investigate the conservation of basic thermodynamic relations when applying an ANN which is crucial at real-gas conditions. Next, we are aiming to use ANNs not only in single-component cases but also in multi-component and combustion cases.

Acknowledgements

Financial support has been provided by the German Research Foundation (Deutsche Forschungsgemeinschaft – DFG).

REFERENCES

[1] Oschwald, M., et al. *Combustion Science and Technology*, 178(1-3):49–100, 2006.
[2] Chehroudi, B. *International Journal of Aerospace Engineering*, 2012:1–31, 2012.
[3] Oefelein, J. C. and Yang, V. *Journal of Propulsion and Power*, 14(5):843–857, 1998.
[4] Schmitt, T., et al. *Comptes Rendus Mécanique*, 337(6-7):528–538, 2009.
[5] Sharma, R., et al. *Computers & chemical engineering*, 23(3):385–390, 1999.
[6] Mohanty, S. *Fluid Phase Equilibria*, 235(1):92–98, 2005.
[7] Lashkarbolooki, M., et al. *The Journal of Supercritical Fluids*, 75:144–151, 2013.
[8] Mayer, W., et al. *Heat and Mass Transfer*, 39(8-9):709–719, 2003.
[9] Müller, H., et al. *Physics of Fluids*, 28(1):015102–1–015102–28, 2016.
[10] Krizhevsky, A., et al. In *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.
[11] He, K., et al. *arXiv preprint arXiv:1512.03385*, 2015.
[12] LeCun, Y., et al. *Nature*, 521(7553):436–444.
[13] Simonyan, K. and Zisserman, A. *arXiv preprint arXiv:0902.0885*, 2014.
[14] Peng, D. and Robinson, D. B. *Ind. Eng. Chem. Fundam*, 15(1):59–64, 1976.
[15] Bishop, C. M. Springer, 2006.
[16] Gron, A. O'Reilly, 2017.
[17] Schmidhuber, J. *Neural Networks*, 61:85–117, 2015.
[18] URL `https://keras.io/`.
[19] URL `https://www.cntk.ai/pythondocs/layerref.html`.