Hyper Parameter Optimization

Nijat Rustamov

Introduction

An accurate understanding of transport mechanisms in nano-confined systems is critical for many applications relevant to energy and sustainability technologies. However, the intricate physics governing the gas flow in confined media challenges the scientific efforts to bridge the gap between continuum and free molecular flow scales. In an effort to mitigate this problem a highly efficient numerical simulation model powered by the lattice Boltzmann method was developed and optimized capable of undertaking extremely large and complex porous media with a wide range of Knudsen number. However, as for any other numerical method, simulation of large number of samples is restricted by computational time. In this course, I decided to start exploring and implementing what I have learned so far and apply it to the domain of my interest. In this exercise and several subsequent ones, I generate numerous artificial porous media, run the numerical simulations, and try to formulate the problem as a machine learning task to predict the flow behavior.

Dataset description

The dataset includes numerical simulation domains with the dimensions of 500 x 500 and the results of those simulations. Domain generation and processing was written in MATLAB and Figure 1 in Appendix section shows a few samples. The processing part included skeletonization of image, calculation of mid-axis and local pore size. Using local pore size information Knudsen number is then calculated. These parameters are used as input to the numerical simulation and the output is the x and y direction velocity distributions as well as density distribution. Numerical simulation is written in C++ from scratch using the theory of Lattice Boltzmann Method with specialized boundary conditions and equation of states to capture the physics of nano-confined gas flow. Several optimization and parallelization techniques have been applied to speed up the program and enable the utilization of HPC. For confidentiality reasons, I am unable to share the source code. Please refer to the publications in the references section for technical details [1, 2]. The Knudsen number distribution was used as input to the ML to match the x direction velocity distribution from the numerical simulation. Sample Knudsen number and velocity distributions are shown in Figure 2 in Appendix. In total, 300 samples have been generated and simulated over the period of two weeks. **To access the data used in this exercise, please follow the link to my OneDrive account, due to 25 MB limitation on GitHub I could not upload it there.** HPO_DATASET

Experimental Setup

All 300 input-output pairs are loaded and split into the training and test set. Each set was normalized by their maximum value. Using the CNN architecture shown in Figure 3 grid search was deployed to find the best settings. The search space included 4 activation functions, 5 loss functions and 4 optimizers. Due to long runtimes, only 5 epochs were run. Table 1 in Appendix shows all the parameters used in grid search. Overall, there are 80 trainable models in the search space and the best model was selected based on the lowest test error. In fact, both the training and the test loss pointed out to the same set of settings for the best model which was "relu", " mean squared logarithmic error" and "adam" for activation function, loss function and optimizer respectively. The training procedure was carried out via 5-fold cross validation to watch out for overfitting, and none was observed. Then the best model was trained again on a training set for 100 epochs. 80 % of the samples (i.e., 240 samples) were used in training and the model was tested on the remaining 60 samples.

Results and Discussion

Generally speaking, the results are not as good as I expected. Figure 4 shows the training curves where validation loss represents the 5-fold cross validation. It can be concluded from the downward trend of the curve that letting training run for more iterations would further decrease the loss. At the end of the training, final loss on training data was around 0.035 and for test data that figure was around 0.0086. Although, the loss is quite low, the pixel values of the samples are low as well, which leads to significant differences apparent from Figures 5 and 6. It looks like the most important thing the model has learned was to recognize the boundaries where the velocity is zero. However, it failed to capture the fundamental physical relationships between Knudsen number and velocity. Of course, there is a lot to be done in this project such as dimensionality reduction of images. However, due to the limited number of samples I could not manage to apply reduction techniques. Moreover, the architecture of CNN could be tuned for better performance, expanding the search space, and implementing different HPO algorithms could return better ML model with more accurate performance. This exercise serves as a preliminary demonstration of what it would look like to apply CNN on a problem like this. My future research plans include the implementation of ML for fast prediction of transport properties, hence I am interested in the subject and turned it into coursework.

Acknowledgements

I have used ChatGPT for debugging some parts of my Python code and in writing this report. Although it was useful in writing, debugging was not so successful, but it did point me into the correct direction for solving the small bugs in the code quickly.

Appendix

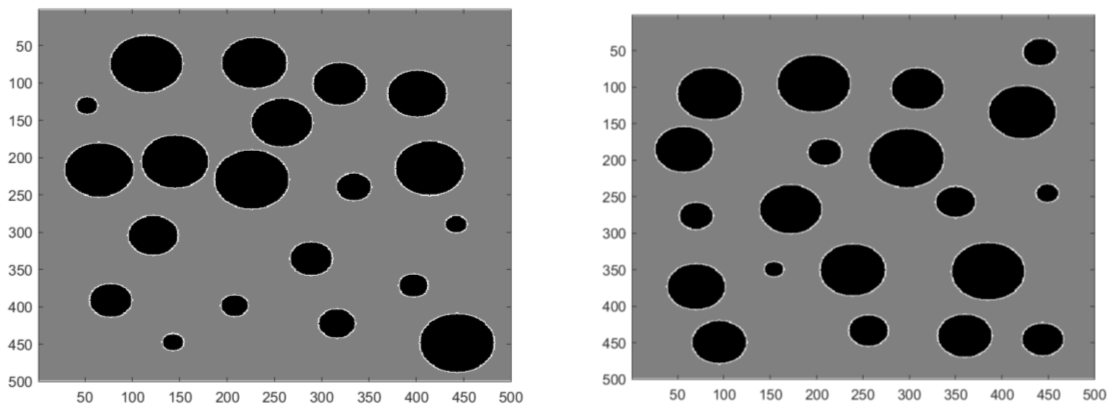

Figure 1. Randomly generated simulation domains. Gray areas indicate pores and black areas (circles) represent grains (boundaries) that fluid cannot penetrate.

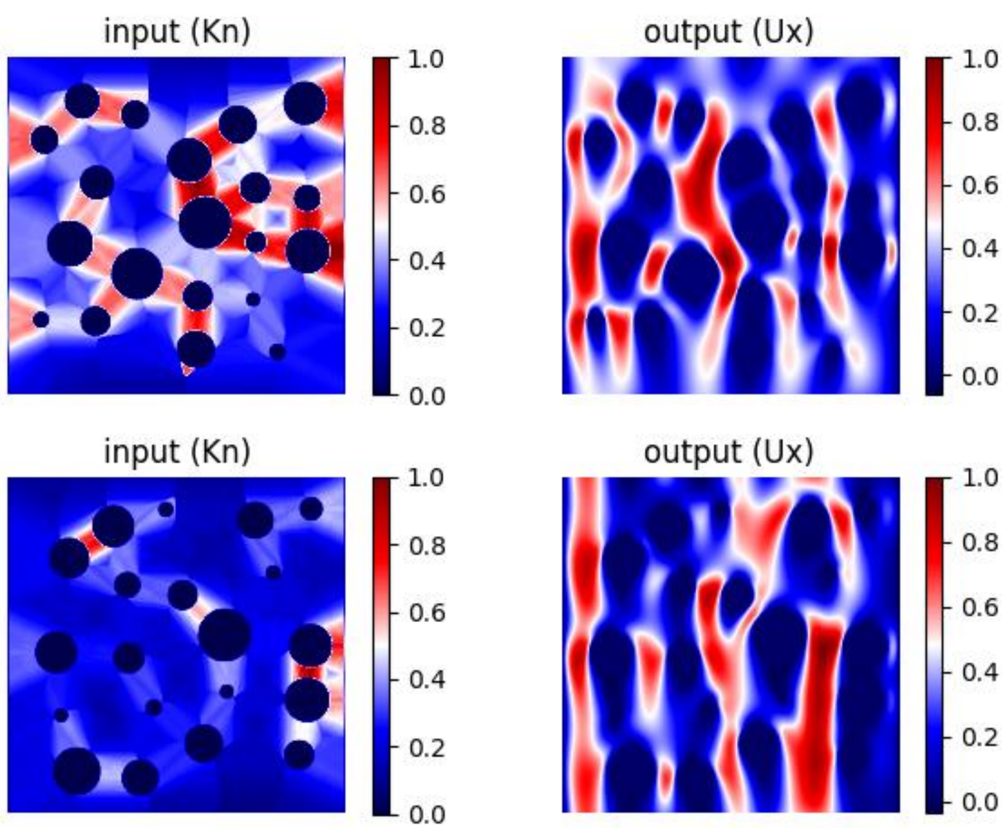

Figure 2. Sample input (left column) Knudsen number distributions and output (velocity) distribution. Data has been normalized by maximum value.

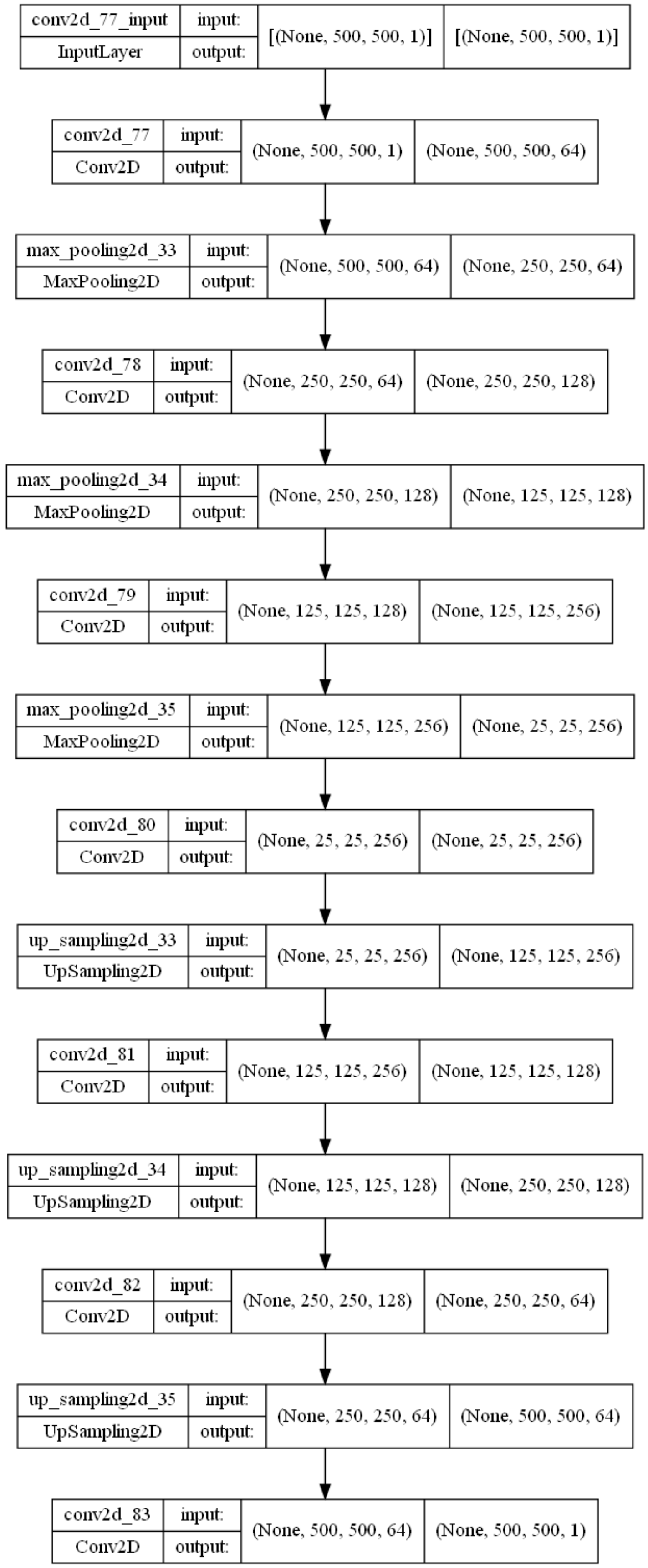

Figure 3. CNN Architecture

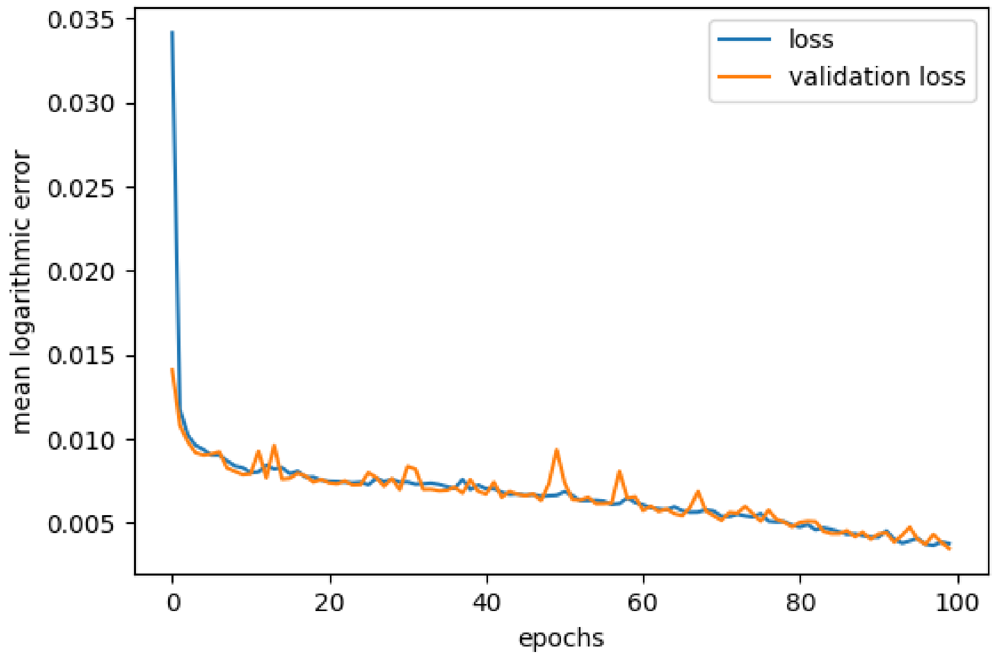

Figure 4. Training curve

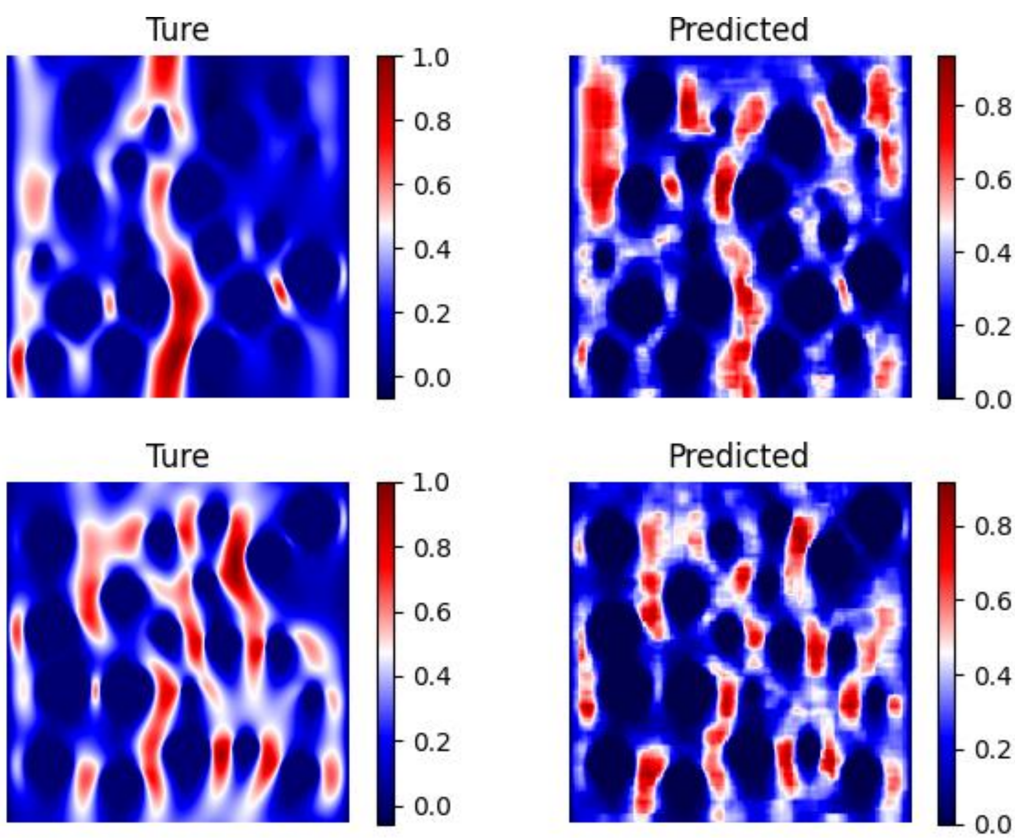

Figure 5. Sample predictions of training samples

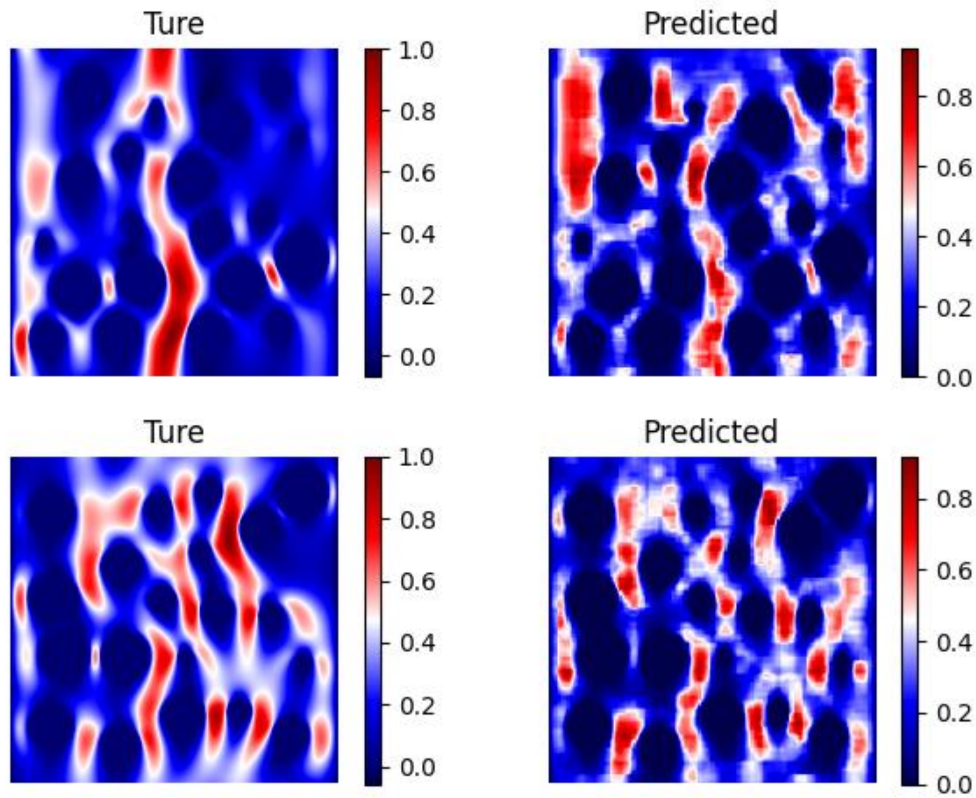

Figure 6. Sample predictions of test samples

Table 1. Search space for grid search

Loss functions	Activations	Optimizers
Mean Sq. Error	Relu	RMSProp
Mean Abs. Eror	Sigmoid	Adam
Mean Abs. Perc. Error	Tanh	Adamax
Mean Sq. Log Error	Softmax	AdaDelta
Cosine Similarity		

References

1. Rustamov, N., Douglas, C. C., & Aryana, S. A. (2022). Scalable simulation of pressure gradient-driven transport of rarefied gases in complex permeable media using lattice Boltzmann method. *Fluids*, *8*(1), 1. https://doi.org/10.3390/fluids8010001
2. Rustamov, N., Liu, L., & Aryana, S. A. (2023). Scalable simulation of coupled adsorption and transport of methane in confined complex porous media with density preconditioning. *Gas Science and Engineering*, *119*, 205131. https://doi.org/10.1016/j.jgsce.2023.205131