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.

The numerical simulations are based on the continuous Boltzmann equation given as

$$\frac{\partial f}{\partial t} + \vec{\xi} \cdot \vec{\nabla} f + \frac{\vec{F}}{m} \cdot \vec{\nabla} f = \Omega(f) \tag{1}$$

Where f represent particle distribution function in space and left-hand side describes the movement of particles in space and time and right-hand side describes the collision dynamics. The discretized multi-relaxation time Boltzmann equation is given by

$$f_\alpha(x + ce_\alpha \delta t, t + \delta t) = f_\alpha^{eq} + \widetilde{f_\alpha} - \sum_\beta (\boldsymbol{M}^{-1}\boldsymbol{SM})_{\alpha,\beta} \widetilde{f_\beta} + \delta t F_\alpha(x, t) \tag{2}$$

The details of what each term stands for can be found in [2]. With proper boundary conditions the method is capable of simulating flows in confined (nanoscale) media. In this work, Knudsen numbers are used to represent the scale. Knudsen number is the ratio of mean free path of the fluid to the representative pore diameter. In this work, fluid is methane gas flowing through the pores as shown in Figure 1 in Appendix. At high Knudsen numbers the flow goes into slip, transitional and free molecular flow regimes where the slip velocity cannot be neglected.

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 minimum and maximum values. PCA was implemented to reduce the dimensions of the images from 500 x 500 to 15 x 15 by keeping the 98 % of the variance. The output of PCA was normalized between 0 and 1 again. Using the CNN architecture shown in Figure 3 grid search was deployed to find the best settings. For hyper-parameter optimization two algorithms were considered: Bayesian optimization and grid search. The search space included 4 activation functions, 4 optimizers and 4 loss functions. Keras Tuner library was used to run the optimization. Bayesian optimization was run with 50 trial points 50 epochs for each run. The best model was found to have "sigmoid", "adam" and "mean squared logarithmic error" for activation function, optimizer, and loss function respectively. Grid Search was run for 64 trial points each with 50 epochs and the best model was found to have "tanh", "adamax" and "mean squared logarithmic error" for activation function, optimizer, and loss function respectively. Then the best model from each optimizer was trained on training data for 200 epochs. The learning curves are shown in Figure 4. The training procedure was carried out using 5-fold cross validation.

Results and Discussion

Generally speaking, the results are not as good as I expected. The final validation loss as a result of training after Bayesian optimization was around 0.0152 and for grid search this value was 6.6×10^{-4}. Which means that grid search found a better model to train than Bayesian search. This can also be seen from Figures 5-8. It looks like the best model found by Bayesian search did not learn anything and performed poorly on both training and test images. Grid search, however, had more accurate predictions on training samples but still struggled with test images. Of course, there is a lot to be done in this project such as trying to figure out why Bayesian search found worse model than gird search and enhancing search space to include number of kernels and kernel size, etc. However, due to the limited number of samples I could not manage to apply reduction techniques. 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

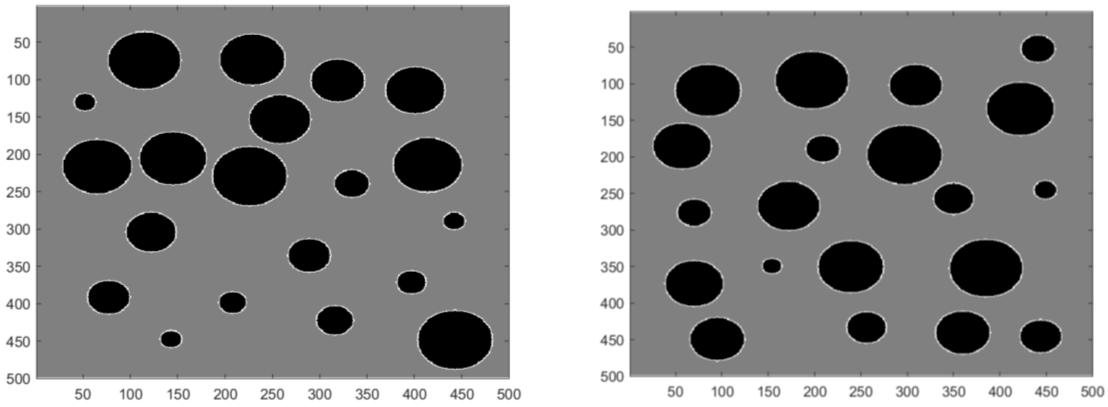

Figure 1. Randomly generated simulation domains. Gray areas indicate pores and black areas (circles) represent grains (boundaries) that fluid cannot penetrate.

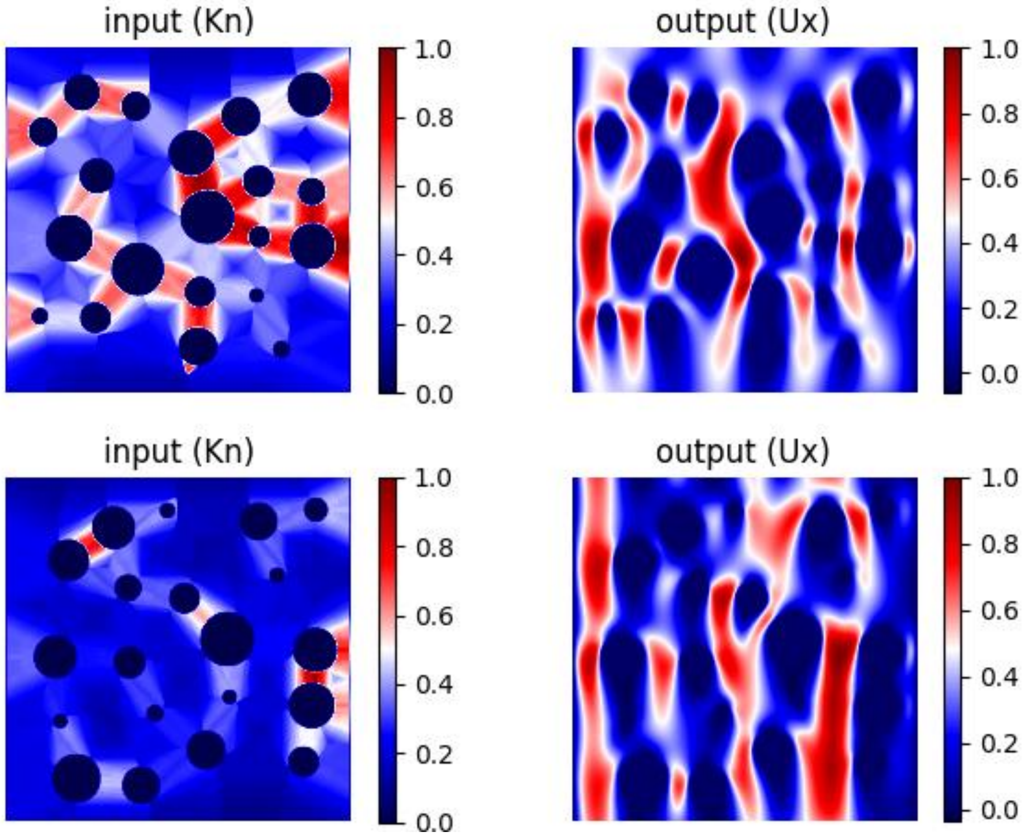

Figure 2. Sample input (left column) Knudsen number distributions and output (velocity) distribution. Data has been normalized between 0 and 1.

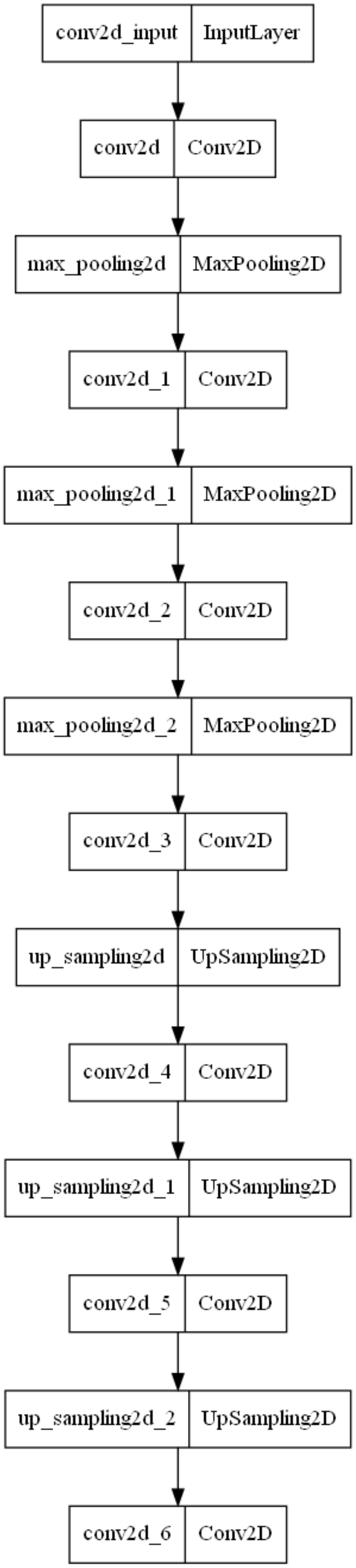

Figure 3. CNN Architecture

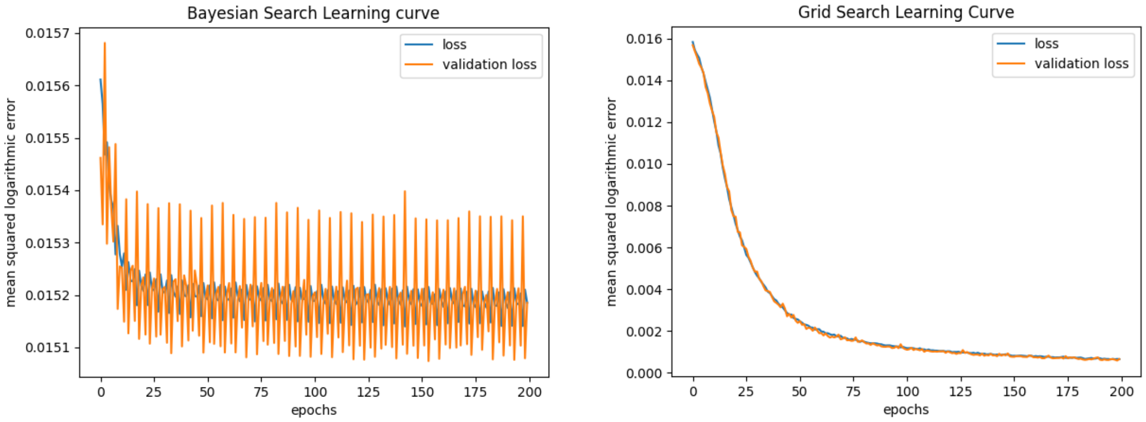

Figure 4. Training curve

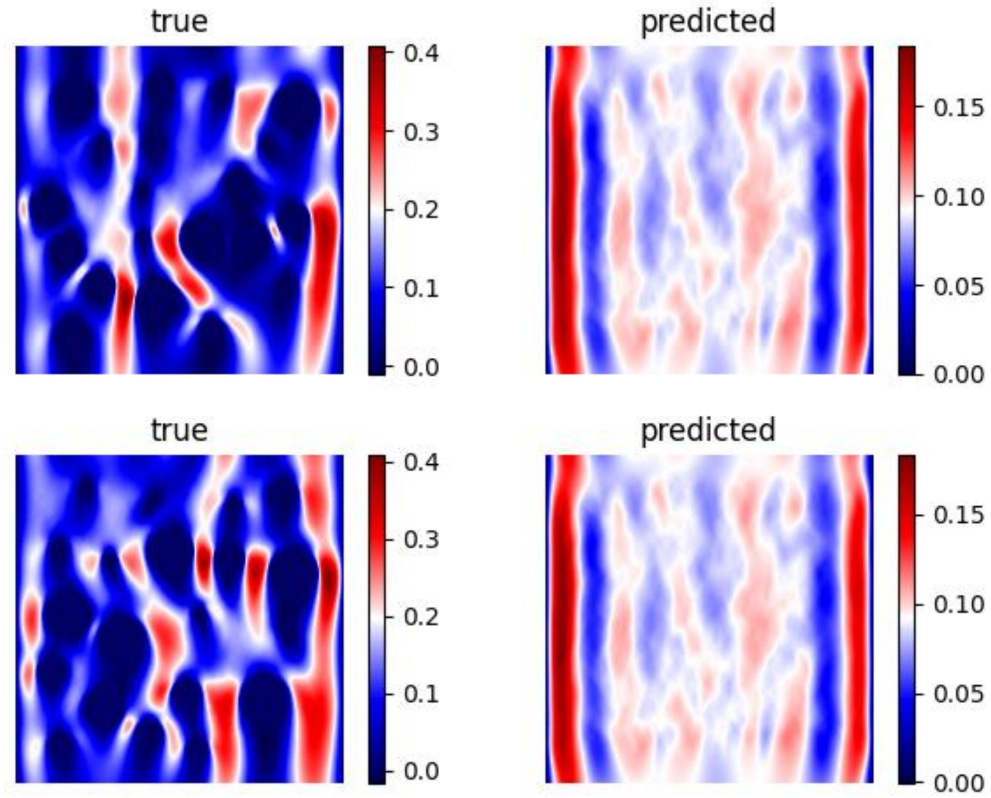

Figure 5. Sample predictions of training samples from Bayesian search

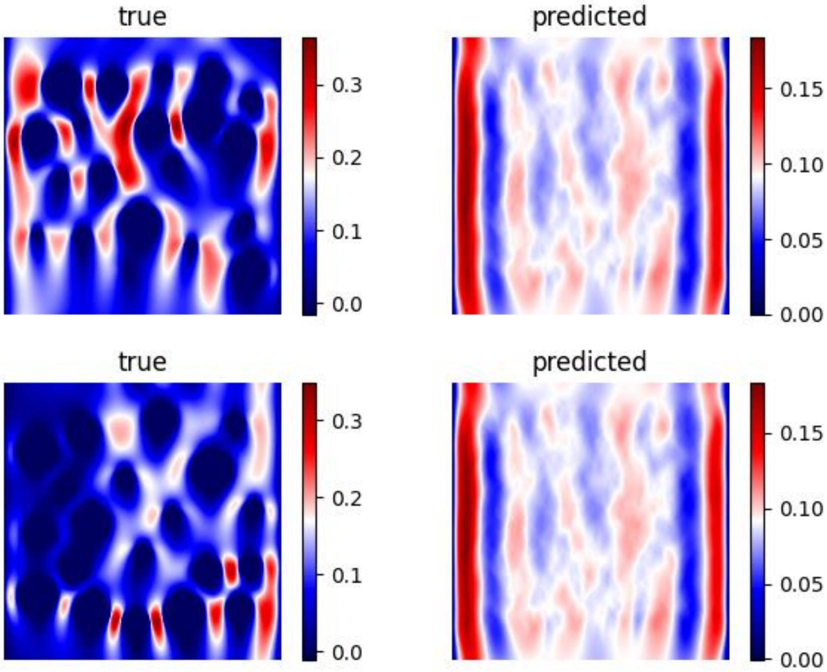

Figure 6. Sample predictions of test samples from Bayesian search

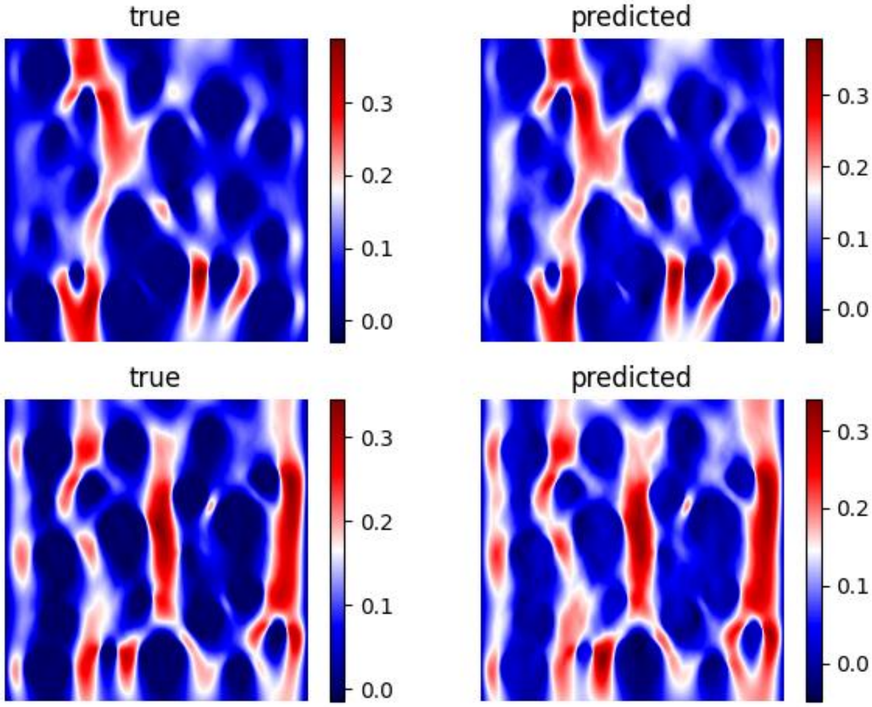

Figure 7. Sample predictions of training samples from grid search

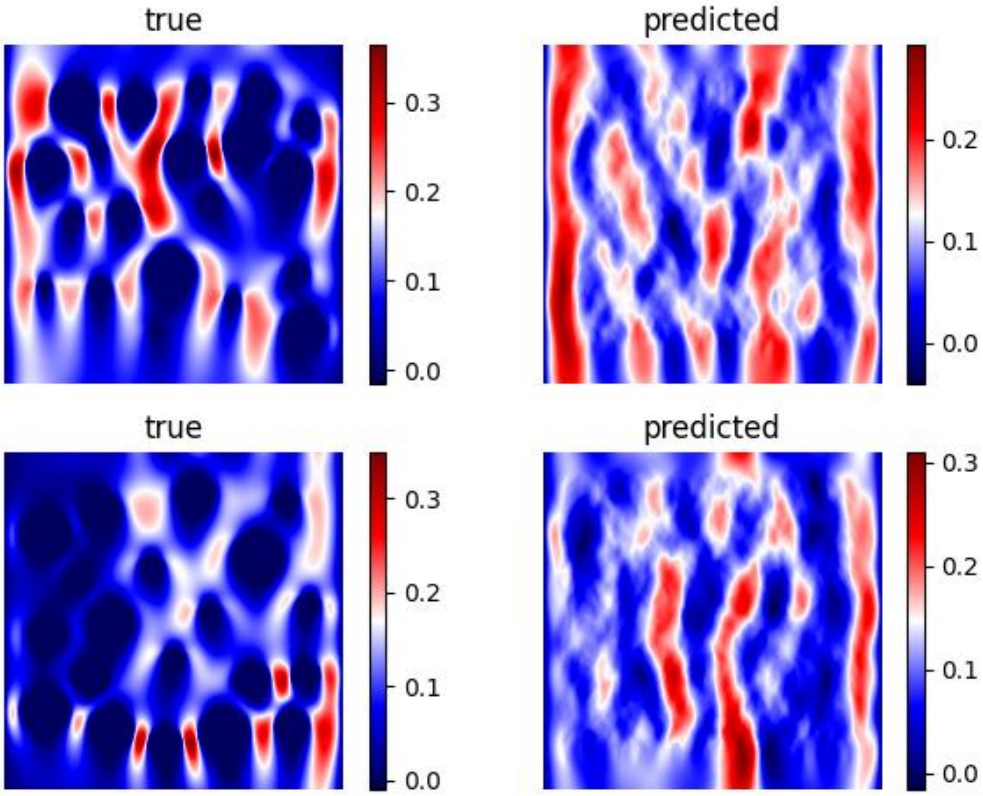

Figure 8. Sample predictions of test samples from grid search

Table 1. Search space for hyper-parameter optimization

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

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