Optimization and application of neural network models for accelerated predictive modelling of NSTX-U

Justin Kunimune, Vaish Gajaraj, Dr. Dan Boyer, and Dr. Michael Zarnstoff

Princeton Plasma Physics Laboratory

Background

Real-time predictive modeling is an important prerequisite to making fusion reactors reliable and cost-effective. However, physics-based first-principles models are far too slow for such purposes. Neural networks are much faster, and almost as accurate, but they have many parameters that must be tuned. Even if, after tediously adjusting all of the levers, you get a model that works, how do you know that it's the *best* model?

Objectives

- Develop a method of optimising neural network parameters
- Enable the tuning of weights between model speed and fidelity
- Apply it to the NUBEAM code to create an optimally fast and accurate neural network ensemble

Genetic Algorithm

1. A group of sets of parameters, or "topologies", is randomly created.
2. Each topology is evaluated on both speed and accuracy.
3. The best topologies are randomly bred to create new topologies.
4. Steps 2 and 3 are repeated a set number of times.
5. The best topology of the final network is returned.

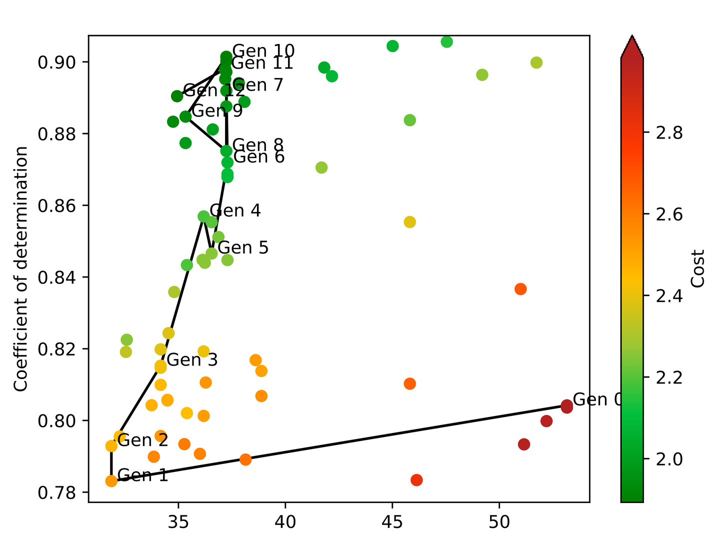

Figure 1: Progress of algorithm over time.

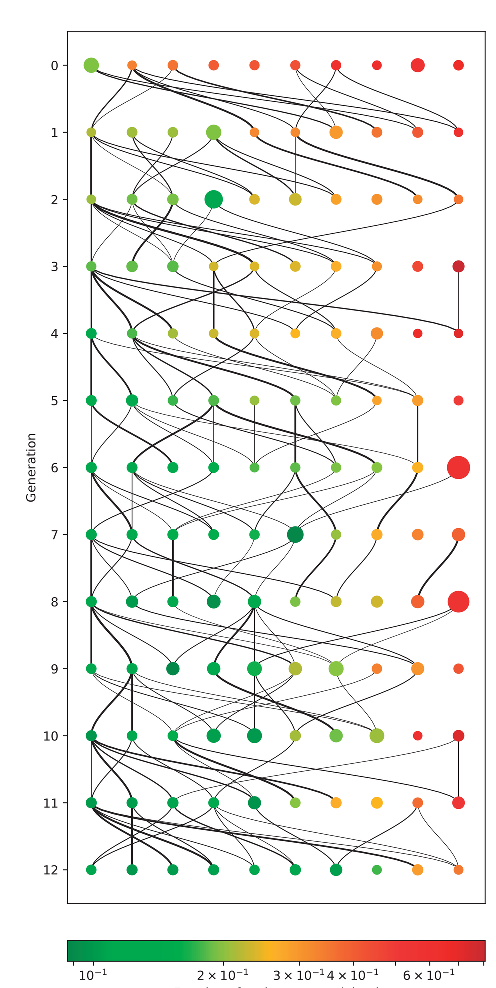

Figure 2: Passage of genes over 13 generations.

This algorithm mimics natural selection and produces optimal neural network parameters within 20 generations of 20 individuals.

Computation

Each topology can take over 30 minutes to evaluate. Running many large generations was therefore infeasible. With the aid of PPPL's Unix cluster, though, each topology of each generation could be evaluated in parallel, each submitted via Slurm to a different CPU. This allowed 24 generations of 20 topologies to be run in just 12 hours.

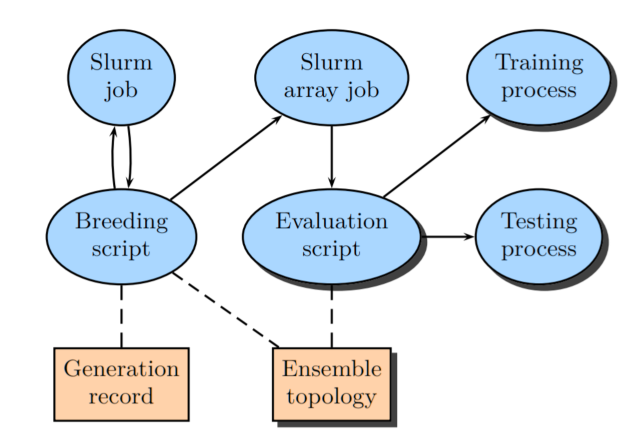

Figure 3: The different parts of the algorithm that enabled topologies to be evaluated in parallel. Code is blue; files are orange.

Results

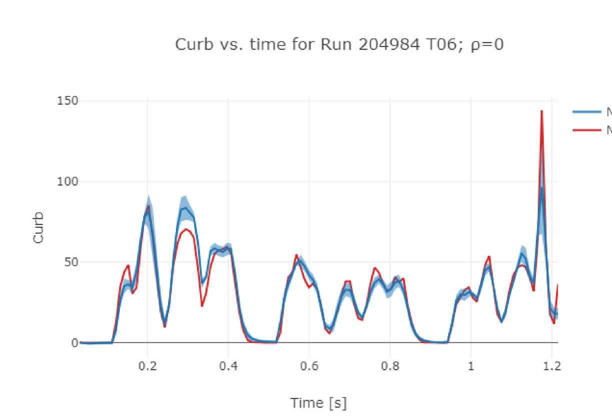

Figure 4: CURB time series comparison.

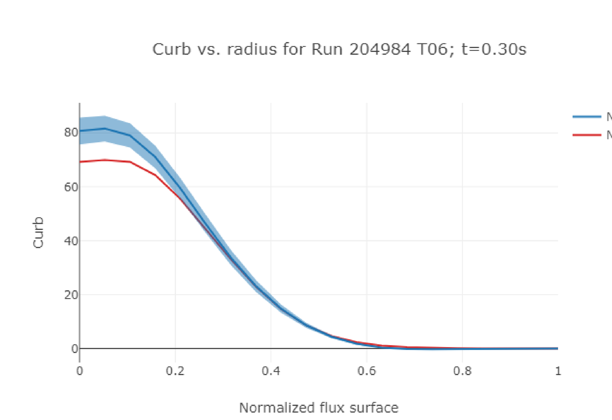

Figure 5: CURB profile comparison.

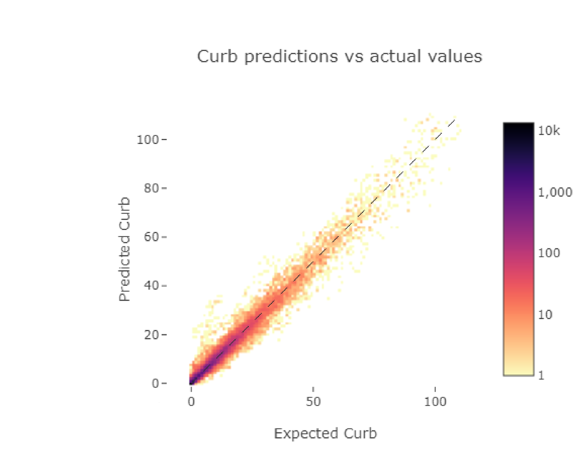

Figure 6: CURB predictions vs. values.

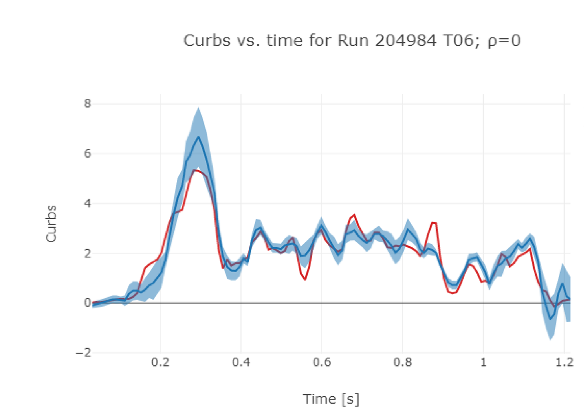

Figure 7: CURBS time series comparison.

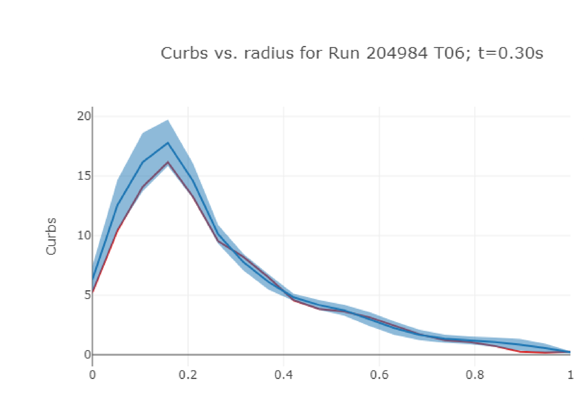

Figure 8: CURBS profile comparison.

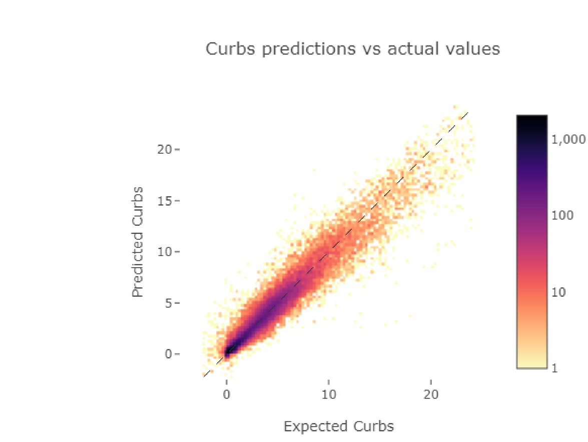

Figure 9: CURBS predictions vs. values.

The optimal model

The optimal model, which balances execution time and accuracy, has two hidden layers of 30 and 66 hidden nodes. It uses higher regularisation and fewer input variables than similar manually-created network models. It also uses smaller time-constants in its low-pass filters and different levels of detail in its profile inputs.

While my models are only marginally better than manually-created ones, they were generated automatically, without need for time-consuming fiddling. This allows physicists to focus on designing datasets and applications rather than tuning model parameters.

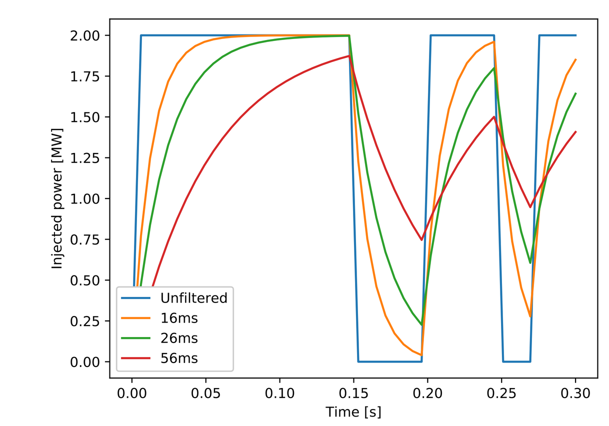

Figure 10: Low-pass filters used to represent time dependence.

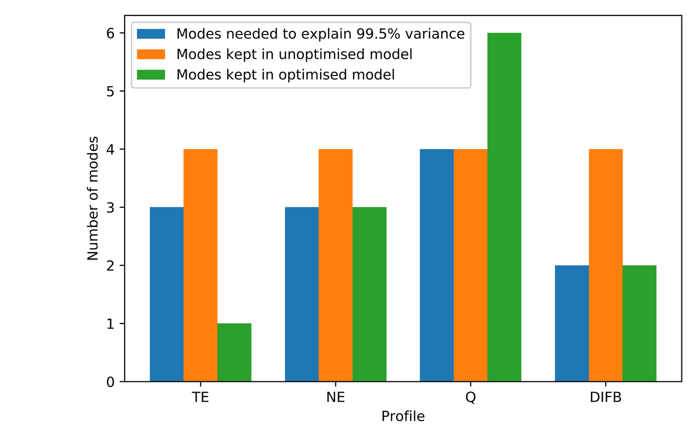

Figure 11: Number of spatial modes used to represent each profile.

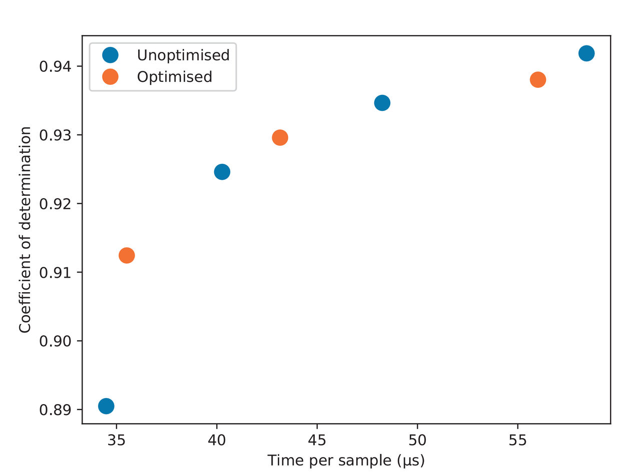

Figure 12: Comparison of unoptimised models with different numbers of nodes and optimised models with different weights.

Acknowledgements

This work was supported by the US Department of Energy Grant under contract number DE-AC02-09CH11466.

Next steps

- Apply genetic algorithm to other models and problems
- Develop feed-back and feed-forward methods for controlling plasmas
- Apply control methods to real fusion devices to manage reactions

References

[1] M. D. Boyer, S. Kaye, D. Liu, K. Erickson, O. Meneghini, and S. Sabbagh, "Real-time capable neural network approximation of nubeam for use in the nstx-u control system." Submitted to European Physical Society Plasma Physics Conference, 2018.

[2] O. Meneghini, C. J. Luna, S. P. Smith, and L. L. Lao, "Modeling of transport phenomena in tokamak plasmas with neural networks," *Physics of Plasmas*, vol. 21, 2014.

[3] O. Meneghini, S. P. Smith, P. B. Snyder, G. M. Staebler, J. Candy, E. Belli, L. Lao, M. Kostuk, T. Luce, T. Luda, J. M. Park, and F. Poli, "Self-consistent core-pedestal transport simulations with neural network accelerated models," *Nuclear Fusion*, vol. 57, 2017.

[4] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.