DESIGN OF A DEEP LEARNING SURROGATE MODEL FOR THE PREDICTION OF FHR DESIGN PARAMETERS

A.J. Whyte[1], Z. Xing[1], G.T. Parks[1] and E. Shwageraus[1]
[1]University of Cambridge
Department of Engineering, Trumpington Street, Cambridge, CB2 1PZ

ajw287@cam.ac.uk, zx227@cam.ac.uk, gtp10@cam.ac.uk, es607@cam.ac.uk

ABSTRACT

Following previous work by Xing and Shwageraus, a large corpus of data has been collected for simulated AGR-style fuel assembly design in FHRs. The results exhibit a non-linear system response, so a 'deep' multi-layer perceptron surrogate model is designed and tested for prediction of design parameters. This neuro-surrogate regression model could be useful for the fast optimization of the design parameters, for example in multi-objective optimization problems, due to the extremely fast evaluation time. Source code is made available for the audit and authentication of the scientific method.

KEYWORDS: FHR, deep learning, surrogate model, AGR, fuel design, neural network

1. INTRODUCTION

Advanced Gas-cooled Reactors (AGRs) have a number of highly desirable design attributes, such as high coolant outlet temperature and online refuelling capabilities [1]. These attributes lead to high thermal efficiency and long continuous operation times. There is potential to create designs that exploit these features for Fluoride-salt-cooled High-temperature Reactors (FHRs) [2]Although FHRs differ from AGRs in that they are salt-cooled, they have attributes in common, being graphite-moderated, high-temperature reactors with a prismatic block fuel design. Fig. 1(a) shows the essential elements of an AGR fuel element. A central tie bar is surrounded by 36 fuel pins, with a 0.4 mm steel cladding. Coolant flows around the pins and is contained within an inner graphite sleeve. It has been proposed by Forsberg *et al.* [3] that utilization of the knowledge gained from the design of AGR fuel assemblies might be advantageous in the design of FHRs. As well as being commercially proven, AGR fuel designs have the advantages of many years of operational experience, applied use of graphite at high temperature, and benefit from developed manufacture and spent fuel management techniques and infrastructure. Xing and Shwageraus [4] advanced this idea with an investigation of the design space for a number of parameters. In their investigation, as well as extended studies by Xing [5], [6], a significant amount of data was generated for fuel cell models. Single pin simulations were carried out within a homogenized moderator/coolant with reflected boundary conditions, as shown in Fig. 1(b). Each of three types of pin (solid, annular and plates) were simulated for three different coolant salts (NaF-ZrF_4, FLiBe, FLiNaK) at two coolant temperatures (900 K, 1000 K), over a range of enrichments and salt-to-moderator volume ratios. Furthermore, for each type of pin, specific parameters were varied: in solid pins, the pitch and

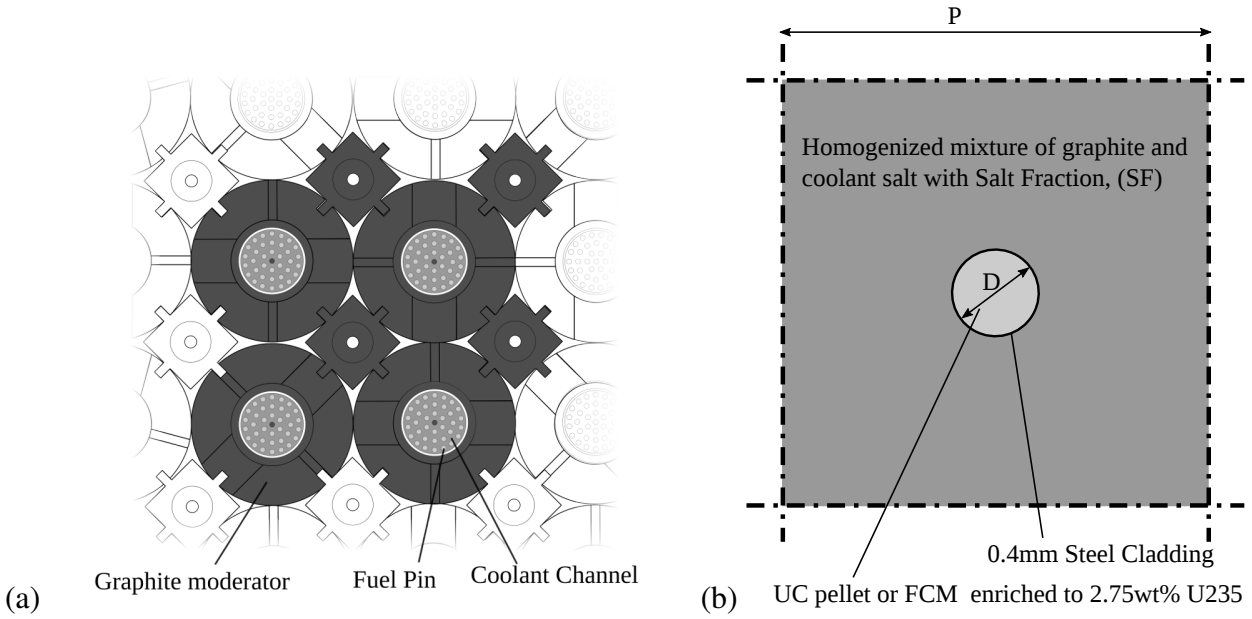

Figure 1. (a) AGR fuel elements are encased by graphite moderator blocks and keys (based on Nønbol [1]), and (b) An example solid pin model used in this paper.

diameter were simulated for a range of values; in annular pins, inner and outer diameters and pitch were varied; in plate-type fuel, a variety of thicknesses and pitch sizes were explored.

The design choices in previous work [4] were made by manually selecting design samples from a uniform grid in the input design space. At each stage of fine tuning of these design parameters, more neutronic simulations were required. This gave rise to the incentive to develop a fast computational tool that can replace computationally expensive neutronic solvers and speed up the design optimization process. The FHR design variables showed combinatorial complexity, and previous simulations saw highly nonlinear neutronic performance in the output space. Simple interpolation schemes struggle to deliver accurate results across the searched space. Deep Multi Layer Perceptrons (MLPs), which have been shown to efficiently represent complex problems [7], are used in this study to develop a surrogate model of the FHR system. The model simulates neutronic parameters of interest (k_∞, Coolant Temperature Coefficient (CTC), Doppler Coefficient (DC), discharge burnup (BU)) of the unit cell FHR models from input data such as geometric information (pin pitch, diameter), fuel type (uranium carbide (UC), Fully Ceramic Micro-encapsulated (FCM) fuel), and coolant salt. This allows the functionality of a fuel design to be evaluated in hundreds of microseconds, as opposed to minutes or hours, as expected in deterministic or Monte Carlo solutions of the neutron transport equation. The models developed in this study can be employed in future FHR preliminary design work to quickly narrow down the design spaces to optimized regions.

1.1. Simulation Experimental Set-up

In a previous study [5], eleven FHR families covering three fuel forms (solid, annular and plate-type fuel), two fuel materials (UC and FCM) and two salts (FLiBe and NaF-ZrF_4) were explored, out of which four were selected for further analysis. First, solid UC fuel with FLiBe coolant was identified as the best overall neutronic design. Second, the best FCM fuel model, the FLiBe-cooled

solid pin was chosen, because of the robustness of the fuel form and associated safety benefits. Third, the best annular fuel design, the annular FCM FHR cooled with FLiBe was selected, because it showed promising thermal-hydraulic performance and allows higher power uprate [8]. Finally, the best performing NaF-ZrF_4 design, the solid UC pin with NaF-ZrF_4 coolant is selected as a backup option to the mainstream FLiBe-cooled FHRs, because it does not produce tritium. The neutronic data collected for these four design streams have been used in this study to train the surrogate models. The numbers of data samples corresponding to each design stream are summarised in Table I. A total sample population of 33,119 is used in the Beginning Of Life (BOL) experiments, and a total of 270 samples were used in the depletion experiments.

For the first part of this study, the BOL neutronic data of the eleven design families were used to train surrogate models. After proving the concept on these results, a subset of the four selected design families, which had been found to demonstrate the best neutronic performance, were brought forward for depletion analysis, the results of which were used to train new models incorporating depletion information.

Table I. Simulation data summary totals.

Fuel, geometry	Coolant	Total samples	BU samples
UC, solid	NaF-ZrF_4	1260	45
	FLiBe	980	156
	FLiNaK	1260	
UC, annular	NaFZrF	2800	
	FLiBe	2799	
FCM, solid	NaF-ZrF_4	911	
	FLiBe	910	57
FCM, annular	$NaFZrF_4$	6075	
	FLiBe	6075	71
FCM, plate	NaF-ZrF_4	9743	
	FLiBe	306	

In the neutronic data, three geometric inputs were varied: (1) lattice pitch to fuel rod diameter ratio (P/D), assuming an AGR fuel diameter [1]; (2) Salt mass Fraction (SF) in the salt and carbon homogeneous mixture in the simplified unit cell model (shown in Fig. 1(b)); and (3) the enrichment of U235 in the fuel. P/D represents the geometrical envelope and heterogeneity of the model; it is varied from 1.2 to 8 in 0.4 increments to ensure both sufficient cooling of the fuel and reasonably high power densities. SF represents the measure of salt and carbon content in the core, and is varied from 20% to 100% with 20% increments. Fuel enrichment is varied from 2.5 wt% to 20 wt% with 2.5 wt% increments. Two temperature states (900 K and 1000 K) were covered for each design.

For the first part of this study, a new model was created and trained for each category shown in Table I. The data for the category was split by random selection into 80% training data and tested on the remaining 20% for its ability to accurately predict k_∞. The test set is not used in training, so it is considered an unbiased evaluation of the performance of the network. These models could be used by a system designer using the AGR framework for the design of FHR fuel elements, to

guide the designer to find optimal arrangements of pins, pin sizes and to investigate design options without iteratively carrying out computationally expensive simulations.

2. NEURO-SURROGATE MODEL DEVELOPMENT

Design simplicity was favoured in order to allow the efficient development of a functional surrogate model. However, considerable optimization is possible on neural network models that is beyond the scope of this study.

2.1. Neural Network Architectural Investigation

Neural networks have previously been used in nuclear engineering to predict core parameters, e.g. for burnup predictions by Noda *et al.* [9] and in optimization of core loading patterns by Faria and Pereira [10] and Kim *et al.* [11]. An Artificial Neural Network, such as the one shown in Fig. 2, is usually set up with low random initial weights between nodes so that differentials for the error values can be established. Then error is minimized by adjusting weights ($\mathbf{w}^1, \mathbf{w}^2, \mathbf{w}^3$).

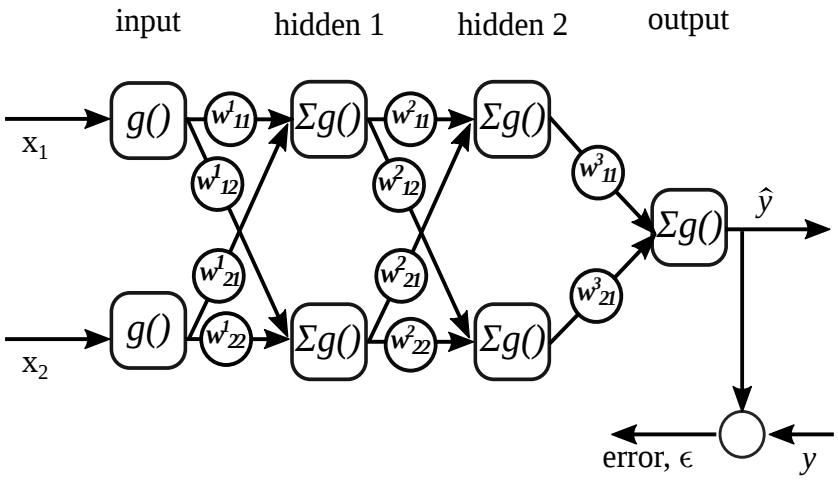

Figure 2. A simple feed-forward, MLP-type neural network.

Although myriad network topologies exist, relatively early work in neural networks showed that three-layer networks could act as universal function approximators [12], so work concentrated on shallow networks until the recent interest in deep learning began, summarised concisely by LeCun *et al.* [13]. Deep learning networks showed a marked improvement on shallow networks. Telgarsky [7] showed in 2015 that there are, in fact, functions that cannot be represented *efficiently* by shallow networks, and that a large set of functions are more easily represented in deep learning networks. A great deal of progress has been made in the last ten years, including novel topologies and efficient algorithm implementations [14, 15]. In 2016, Mhaskar *et al.* [16] described the kind of problems for which deep or shallow learning is advantageous. They show that, for complex problems, orders of magnitude of improvement in error performance is achieved with three or more hidden layers.

Even on only moderately complex problems, performance is shown to be better with deep learning networks.

When training networks, a pseudo-random number generator, the Mersenne Twister [17], is used to generate the initial weights connecting each node (neuron) prior to training. All models are trained for 250 epochs, each with a batch of 35 data samples. The neuron activation function used is one that has gained popularity recently, called Rectified linear units (Relu) [18, 19], mathematically represented as:

$$f(x) = \begin{cases} x, & \text{if } x \geq 0 \\ 0, & \text{otherwise} \end{cases}$$

Relu simplifies the computational task for each neuron while still performing the function of stopping neuron weights from tending to infinity, which can happen if two weights cancel each other out and the back-propagation algorithm modifies them in opposite directions. Compared to the traditional sigmoid function, Relu can often allow quicker learning of deep neural networks on large and complex sample spaces. The loss function uses root mean square (RMS) error to quantify the error of the network. Lastly, the 'Adam' optimizer [20] is used instead of, as is traditional, stochastic gradient descent to back-propagate the errors and update the weight of the neurons. Adam combines the advantages of two forms of gradient descent algorithm, namely the Adaptive Gradient Algorithm and Root Mean Square Propagation, to allow better performance on noisy data samples and large datasets with many input parameters [20].

Since a large corpus of data was already available for this study, it is relatively simple to use it for iteratively testing a variety of neural network topologies.

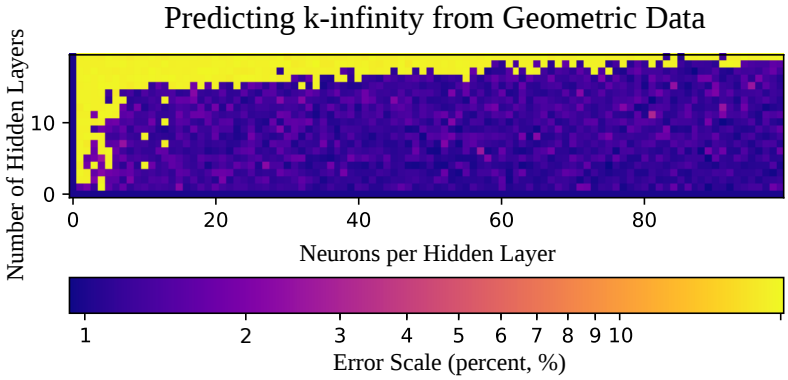

Figure 3. Error heatmap for different sized networks; this data guides the choice of network size used in Table II.

To establish a suitable network topology, tests were carried out on a range of networks. The performance of tested topologies is illustrated by the heatmap shown in Fig. 3, with RMS error characterised by the colour scheme transitioning from yellow to dark blue, with the darker shades representing better accuracy. Each pixel represents the prediction error of a dense, rectangular, feed-forward network, with an input layer of 50 neurons and between 1 and 20 hidden layers and

from 1 to 100 neurons per hidden layer in the network. The problem is 'embarrassingly parallelizable' [21, p 14] and can be run relatively quickly without requiring substantial computational resource. From the heatmap, it can be seen that with a small number of neurons per hidden layer and/or a larger number of ($\sim$20) hidden layers, the model performs poorly. With a small number of total nodes, the level of complexity of the problem cannot be effectively represented by the network. If the network has too many layers, the back-propagation of errors across many layers of weights becomes impossible.

The network is, at this point, trained using BOL data only. How the physics changes as a result of depletion has not yet been examined; without this, prediction of the reactor's economic and safety performance would be incomplete and unconvincing. The discharge burnup of the reactor, which represents the revenue from reactor operation, has not been incorporated into the database. Also, how reactivity feedbacks change with depletion is important for the multi-batch/on-load refuelling management scheme. To incorporate burnup characteristics into the surrogate model, depletion calculations are carried out for well-performing samples from the four selected design families. Exercising the network architectural investigation using the depletion data, the heatmap for the prediction error shown in Fig. 4 is produced. In this experiment, dense, feed-forward networks, with an input layer of 150 neurons and between 1 and 20 hidden layers and from 1 to 100 neurons per hidden layer in the network, are trained and tested. In this case, it appears that there are no longer sufficient samples to fully train the networks up to the limits of the back-propagation algorithm. Shallower networks have superior results, implying that there are not enough samples to propagate errors back through deeper networks.

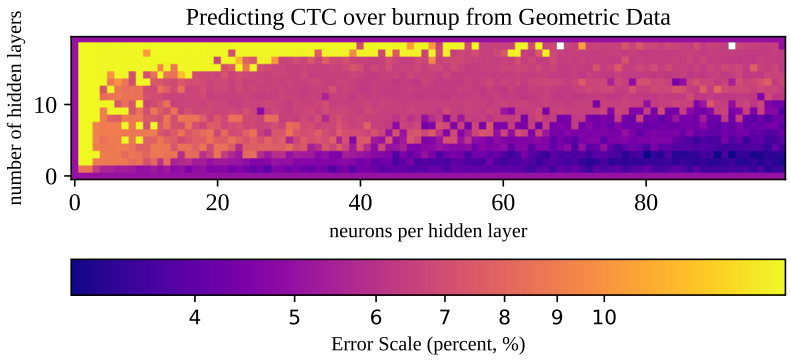

Figure 4. Error heatmap for depletion study of solid pin designs

2.2. Evaluation of Accuracy of Deep Learning Networks

In order to establish the accuracy of the deep learning networks at predicting k_∞ from the experimental input data, scripts were written that automatically tested the training of the networks over a range of random number generator seed values. Adjacent random seeds are used to study the variation of the results. The RMS error and standard deviation, σ, of the model on the test set were recorded. For stochastic algorithms, it is standard to test over 30 (or more) seed values. As before, 80% of the data was used for training and 20% for testing. This was repeated for each random seed – meaning that a new random subset of the hypercube data was used for each test set.

3. RESULTS

3.1. Beginning of Life k_∞ Prediction

This study shows that a deep MLP will accurately predict k_∞ for a variety of proposed fuel pin designs. Table II presents a summary of the results achieved using a sequential architecture deep learning network to predict k_∞ for a particular set of design parameters. Based on the results in Fig. 3, a network size of 10 layers of 50 neurons was selected for further experiments. This can be seen to be reasonably far from the high error regions for this problem. Le Cun *et al.* [13] suggest 5 to 20 hidden layers define a deep network, but it is now common to define networks with more than one hidden layer as deep, e.g. [22].

Table II. Prediction accuracy (averaged over 30 runs) of sequential networks trained and tested on fuel pin design data.

Fuel, geometry	coolant	RMS k_∞ error / pcm	k_∞ σ	RMS k_∞ error / pcm	k_∞ σ
UC, solid	FLiBe	655.31	223.73	482.08	187.05
	$NaF\text{-}ZrF_4$	823.74	298.08		
	FLiNaK	749.49	249.94		
UC, annular	$NaF\text{-}ZrF_4$	701.97	262.46	559.26	214.49
	FLiBe	711.77	344.43		
FCM, solid	$NaF\text{-}ZrF_4$	552.81	181.71	534.84	180.84
	FLiBe	725.81	187.07		
FCM, annular	$NaF\text{-}ZrF_4$	617.60	221.79	319.95	125.17
	FLiBe	887.85	354.39		
FCM, plate	$NaF\text{-}ZrF_4$	371.96	135.92	381.92	155.16
	FLiBe	624.63	208.72		

Columns 3 and 4 of Table II show a summary of the results achieved when a new network was instantiated for each type of fuel and salt. Columns 5 and 6 show the results for networks that used the salt as an input and were trained on a single set of data created by merging data for the same fuel type and different salts. Somewhat surprisingly these networks out-perform the more specialised networks. This is thought to be due to the extra learning opportunity gained by merging these datasets and allowing these networks to train on more samples.

Experiments show that these pre-trained feed-forward sequential networks evaluate input data to deliver an estimate of k_∞ in 320 microseconds on standard PC hardware. The speed and accuracy of these results make the approach significant for iterative optimization strategies and for initial exploration of the design space. For example, the test set of 270 simulations required $\sim$1.1 cpu hours per burnup step, totalling 2970 cpu hours, whereas the neural networks used in this section are able to approximate the results in < 1 second on a single core of a commercial laptop PC.

3.2. Prediction of Discharge Burnup and Reactivity Coefficients Over Depletion

After demonstrating that the surrogate models can predict BOL neutronic performance of FHRs with reasonable accuracy, depletion analysis is performed for the selected design space up to a bur-

nup of 80 MWd/kgHM. To incorporate fuel cycle economic considerations into the analysis, the discharge burnup to enrichment ratio (BU/e) is adopted as an output of the surrogate model to replace BOL k_∞ as an indicator. The ratio represents the amount of economic gain from the reactor's electricity production (proportional to burnup) per unit of fuel cost (approximately proportional to fuel enrichment) and should therefore be maximized. The discharge burnup of the reactor is calculated as twice that of a once-through fuel cycle, assuming the use of an on-load refuelling scheme for the FHR. The assembly (represented by unit cell models) CTC and DC are obtained at various stages during the depletion. Core-average reactivity coefficients are calculated assuming a three-batch refuelling scheme, to give an approximation of on-load refuelled core-average reactivity coefficients, in case they become less negative or even positive with burn up. Both reactivity coefficients, which are also treated as outputs of the surrogate models, should be kept below zero throughout the cycle to assure stability and favourable behaviour in transients. Using the depletion data, the surrogate models were able to generate performance levels as summarised in Table III.

Table III. Prediction percentage error ($\%$) to 3 significant figures of sequential networks trained and tested on depletion data for fuel pins.

Fuel, geometry	coolant	BU/e	CTC	DC	BU/e	CTC	DC
UC, solid	FLiBe	7.51	7.38	8.54	5.53	5.43	5.50
	$NaF\text{-}ZrF_4$	17.8	17.5	17.6			
UC, annular	FLiBe	12.5	16.1	17.7			
FCM, solid	FLiBe	7.52	7.38	8.55			

Greater errors are evident in this exercise compared with the previous models trained on BOL data. This is partly because significantly less depletion data was available to the networks during training and also because the uncertainty on the training data is higher in reactivity coefficients. It also explains the greater errors reported in Table III for the $NaF\text{-}ZrF_4$ and the annular fuel design families, which had much less data compared with the other cases, as shown in Table I. Again, by merging two UC solid pin neutronic datasets, the network produced more accurate predictions. Absolute RMS error from the surrogate model is, for example, 0.131 $pcmK^{-1}$ for core average CTC in UC solid pins, whereas statistical uncertainties from the Monte Carlo Serpent calculations were found to be around 0.1 $pcmK^{-1}$ for CTC.

The surrogate models' errors range from 5.4% to 17.8% in BU/e, giving absolute errors of the order of 1 MWd/kgHM. The original For the preliminary design purpose as a fuel cycle economics indicator, this is considered reasonably acceptable. The two reactivity coefficients must be kept negative and can be used as either optimization objectives or constraints. The errors from the surrogate models require an additional margin to be put on the two parameters if used as optimization constraints.

For the entire depletion design space covered in this study, BU/e values ranging from 0.07 to 27.13 MWd/kgHM were obtained, with an average value of 8.6 MWd/kgHM. Imposing a constraint of -0.2 pcm/K on both reactivity coefficients and an upper bound of 200 MWd/kgHM on reactor discharge burnup, the current data space produced a best design that can reach a BU/e of 15.4 MWd/kgHM. Compared to in-service LWRs, which have BU/e of $\sim$10 MWd/kgHM, the AGR-like FHR shows great potential in terms of reactor economics. For future study, the surrogate models

can be coupled to optimization algorithms to identify optimized design spaces. More depletion data can be accumulated to specifically target these design spaces, which can, in turn, be fed to the network to further improve its predictive power.

4. CONCLUSIONS

A neural network regression analysis has been applied to a collected body of FHR design data, showing that it is possible to interpolate the data and effectively predict neutronic performance parameters despite a highly nonlinear system response. Initial work shows that it has been possible to reach an expected prediction accuracy for k_∞ of $\sim$675 pcm and that a more generalised model incorporating more results within a single model performs better, achieving an expected accuracy for k_∞ of $\sim$456 pcm. The network thereafter demonstrated acceptable performance when trained on a much smaller sample space that incorporated depletion data. These results rely on the availability of a training set; therefore, this method is significant when iterative simulations are expected anyway. Although a neuro-surrogate model is unable to out-perform Monte Carlo or deterministic transport codes in terms of the accuracy, the sub-millisecond calculation speed of a pre-trained network compared to the relative expense of iteratively running neutronics simulations justifies the investment of seconds of cpu time to train an advanced regression method like a deep learning network, especially for initial studies, before a neutronics code is employed, and for design and optimization purposes, where the fast evaluation of approximate results is desirable. Further testing will show whether meta-parameters can be used to train a network, allowing the creation of a system that could be used to predict the performance of geometries that have not been explicitly simulated.

Code for the surrogate models is available under the permissive MIT license and can be cloned from the repository: *https://ajw287@bitbucket.org/ajw287/fhr-surrogate-mandc.git*

REFERENCES

[1] E. Nonbøl. "Description of the advanced gas cooled type of reactor (AGR)." Technical report, Nordisk Kernesikkerhedsforskning (1996).

[2] C. Forsberg, D. Wang, E. Shwageraus, B. Mays, G. Parks, C. Coyle, and M. Liu. "Fluoride-Salt-Cooled High-Temperature Reactor (FHR) Using British Advanced Gas-Cooled Reactor (AGR) Refueling Technology and Decay Heat Removal Systems That Prevent Salt Freezing." *Nuclear Technology*, pp. 1–16 (2019).

[3] C. Forsberg, J. Richard, J. Pounders, R. Kochendarfer, K. Stein, E. Shwageraus, and G. Parks. "Development of a fluoride-salt-cooled high-temperature reactor (FHR) using advanced gas-cooled reactor (AGR) technology." *Transactions of the American Nuclear Society*, **volume 112**, pp. 566–570 (2015).

[4] Z. Xing and E. Shwageraus. "Design space exploration studies of an FHR concept leveraging AGR technologies." In *ICAPP 2017* (2017).

[5] Z. Xing. *Design Space Exploration for Salt Cooled Reactor Systems*. Ph.D. thesis, University of Cambridge (2019).

[6] Z. Xing and E. Shwageraus. "Molten Salt Coolant Reactivity Feedback in Alternative FHR Designs." In *PHYSOR 2018* (2018).

[7] M. Telgarsky. "Representation benefits of deep feedforward networks." *arXiv preprint arXiv:150908101* (2015). URL https://arxiv.org/pdf/1509.08101.

[8] Z. Xing and E. Shwageraus. "Investigation into Reactivity Feedback of FHR Designs with Alternative Coolants." In *ICAPP 2018* (2018).

[9] H. Noda, A. Yamamoto, Y. Nagasawa, H. Murao, and S. Kitamura. "Core burnup calculations using neural networks." *Proc Topl Mtg Advances in Nuclear Fuel Management II*, pp. 20–39 (1997).

[10] E. F. Faria and C. Pereira. "Nuclear fuel loading pattern optimisation using a neural network." *Annals of Nuclear Energy*, **volume 30**(5), pp. 603–613 (2003).

[11] H. G. Kim, S. H. Chang, and B. H. Lee. "Pressurized water reactor core parameter prediction using an artificial neural network." *Nuclear Science and Engineering*, **volume 113**(1), pp. 70–76 (1993).

[12] G. Cybenko. "Approximation by superpositions of a sigmoidal function." *Mathematics of Control, Signals and Systems*, **volume 2**(4), pp. 303–314 (1989).

[13] Y. LeCun, Y. Bengio, and G. Hinton. "Deep Learning." *Nature*, **volume 521**, pp. 436–444 (2005).

[14] F. Chollet et al. "Keras." (2015). URL https://github.com/fchollet/keras.

[15] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng. "TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems." (2015). URL https://www.tensorflow.org/.

[16] H. Mhaskar, Q. Liao, and T. Poggio. "Learning functions: When is deep better than shallow." *arXiv preprint arXiv:160300988* (2016). URL https://arxiv.org/abs/1603.00988.

[17] M. Matsumoto and T. Nishimura. "Mersenne Twister: A 623-dimensionally Equidistributed Uniform Pseudo-random Number Generator." *ACM Transactions on Modeling and Computer Simulation*, **volume 8**(1), pp. 3–30 (1998).

[18] R. H. Hahnloser, R. Sarpeshkar, M. A. Mahowald, R. J. Douglas, and H. S. Seung. "Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit." *Nature*, **volume 405**(6789), p. 947 (2000).

[19] R. H. Hahnloser and H. S. Seung. "Permitted and forbidden sets in symmetric threshold-linear networks." In *Advances in Neural Information Processing Systems*, pp. 217–223 (2001).

[20] D. P. Kingma and J. Ba. "Adam: A Method for Stochastic Optimization." *CoRR*, **volume abs/1412.6980** (2014). URL http://arxiv.org/abs/1412.6980.

[21] M. Herlihy and N. Shavit. *The Art of Multiprocessor Programming, Revised Reprint*. Morgan Kaufmann (2012).

[22] O. Delalleau and Y. Bengio. "Shallow vs. Deep Sum-Product Networks." In *Advances in Neural Information Processing Systems 24*. Curran Associates, Inc. (2011).