doi.org/10.15392/2319-0612.2024.2541
2024, 12(4) | 01-16 | e2501
Submitted: 2024-06-30
Accepted: 2024-09-20

Multilayer perceptron neural network applied to the neutron transport equation for the single velocity neurocomputational approach to neutronic calculations

Ferreira-Savio[a*], A. G. N.; Franco-Rennan[a], A. G. N.; Conti-Thadeu[a]

[a] Institute for Energy and Nuclear Research, 05508-000, City, São Paulo, Brazil.

*Correspondence: rennanfrancoalmeida@usp.br

Abstract: The performance of neutronic calculations is a fundamental process for the analysis and design of nuclear reactors. However, due to the intrinsic complexity of these calculations, their solution is nearly impossible, whether through analytical or numerical methods. This work, through the application of a four-layer multilayer perceptron artificial neural network to the neutron transport equation, demonstrates the benefits of using neural computing for electronic calculations.

Keywords: Neurocomputational Methods, Neutron Transport Equation, Physical Analysis, Reactor Analysis.

doi.org/10.15392/2319-0612.2024.2541
2024, 12(4) | 01-16 | e2501
Submitted: 2024-06-30
Accepted: 2024-09-20

Rede neural de perceptron multicamadas aplicadas à equação de transporte de neutrons para o modelo de difusão de velocidade única de um reator nuclear com uma abordagem neurocomputacional para cálculos neutrônicos

Resumo: O desempenho dos cálculos neutrônicos é um processo fundamental para a análise e o projeto de reatores nucleares. No entanto, devido à complexidade intrínseca desses cálculos, sua solução é quase impossível, seja por métodos analíticos ou numéricos. Este trabalho, por meio da aplicação de uma rede neural artificial de perceptron multicamadas com quatro camadas à equação de transporte de nêutrons, demosntra os benefícios do uso da computação neural para cálculos neutrônicos.

Palavras-chave: Métodos Neurocomputacionais, Equação de Transporte de Nêutrons, Análises Físicas, Análise de Reatores.

1. INTRODUCTION

In the context of nuclear reactor analysis, Neutronic Calculations are an attempt to describe and model the behavior observed in neutrons within the space of a given reactor. Among the phenomena described by neutronic equations is the neutron transport equation, which aims to describe the distribution of neutrons inside the reactor at any given moment, a fundamental factor for determining the occurrence of various reactions in the reactor.

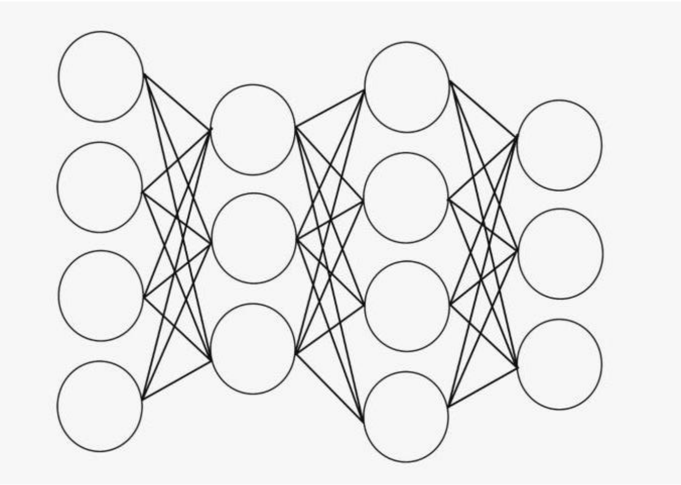
However, despite its evident importance, solving neutronic calculations is challenging and often unattainable, with hopes of exact solutions only feasible for the simplest cases. The difficulty in solving neutronic calculations is rooted not only in the intrinsic complexity of the equations, which are multivariate and nonlinear, making their resolution through analytical means difficult, but also in their high dimensionality, multi-group approximation, and strong coupling. While complexity hinders their resolution through analytical methods, the large number of interactions, variables, and iterative renders their solution impossible through numerical methods [Duderstadt, 1974].

Significant effort is invested in the development of algorithms and numerical methods to aid in solving these equations, thereby enhancing the credibility of analyses that depend on them. However, there is limited discussion in the literature regarding a new approach to solving these problems: neural computing.

Neural computing is an attempt to model and formulate a computational approach capable of mimicking some of the characteristics found in neuron processing and the human brain. Among the targeted features, it's worth noting neuroplasticity and lateralized processing, in which the brain doesn't need to repeat calculations many times, and it can perform various calculations concurrently. Both characteristics, precisely because they reduce the computational demand of processes, make neural computing optimal for solving problems that are insoluble by analytical and numerical methods [Bishop, 1994].

Among the possible models and architectures for solving problems, many of them are sophisticated and particularly powerful for the task. This work opted for the use of the simplest architecture, the multilayer perceptron. Specifically, a 4-layer perceptron, as shown in Image 1, with gradient descent learning, precisely due to its simplicity. The main purpose of this work is to demonstrate the feasibility of applying neurocomputational methods in solving neutron equations, and for this purpose, there is significant value in using the simplest possible example.

Figure 1: 4-layered-perceptron



It's worth noting that the processing of a neural network will be proportionally more costly as the number of weights used in a backpropagation network increases [Bishop, 1994]. Therefore, given that the topology of this neural network has a total of thirty six weights, the expectation is that its processing will be remarkably fast.

The neutron transport equation originates from the Boltzmann equation for the study of gas movement in 1800. It arises from the understanding of concepts such as neutron density, interaction frequency, interaction rate density, as well as angular counterparts, considering the direction of the neutron's trajectory, and angular current density.

These fundamental concepts become the building blocks for describing the processes of neutron gain and loss within a volume of interest inside the reactor. In the single velocity diffusion model, the transport equation is described by Equation 1,

$$\mathbf{\Omega} \cdot \nabla\psi(\mathbf{r}, \mathbf{\Omega}) + \Sigma_t(\mathbf{r})\psi(\mathbf{r}, \mathbf{\Omega}) = \frac{1}{4\pi}\int_{4\pi} \Sigma_s(\mathbf{r}, \mathbf{\Omega}' \to \mathbf{\Omega})\psi(\mathbf{r}, \mathbf{\Omega}')d\mathbf{\Omega}' + \frac{S(\mathbf{r})}{4\pi}$$

$$\frac{\partial N}{\partial t} + V\Omega * \nabla n + \sum_{t} n(r,\ E, \Omega, t\ ) =$$

$$= \int_{4\pi} d\Omega \int_{0}^{\infty} dE'V'\sum_{s} (E' \to E,\ \Omega \to \Omega)N(r,\ E',\ \Omega, t\ ) + S(r,\ E, \Omega, t\ )$$

eq 1

It's important to note that, unlike the Boltzmann equation, the neutron transport equation is linear. The methodology currently used in reactor design albeit being an attempt to solve the neutron transport equation oftentimes is using approximations obtained through analytical and often stochastic methods and can have inherent undesirable computational costs.

Through all the gain and loss functions described in Equation 1, it is observed that the inputs within the functions repeatedly include r, E, t, and Ω, some of them having an apostrophe. These variables are essential for describing the state of a given neutron and are, respectively, the position vector, energy, measurement time, and the direction of particle movement. The counterparts with an apostrophe represent either the energy that entered the relevant energy for solving the problem through the scattering process or the direction that has the same destination.

It is precisely these four variables that become the foundation for modeling the employed neural network. The network is iteratively fed with these four variables, suitably normalized, and trained so that its output corresponds to what can be obtained through stochastic methods and, ultimately, what can be obtained experimentally.

After modeling, the neural network is programmed in Python, using mathematical libraries such as NumPy, and the data to be fed into the network is properly prepared. Once this is done, the neural network is trained and tested. For the training and testing of this neural network, experimental data is used, as well as data obtained using stochastic methods already employed for solving this equation, with appropriate stochastic treatment. The training of this neural network involved introducing the four input variables of the neutron transport equation, time, input energy, orientation, and position, as an input vector in the zero layer and the final calculated or experimentally obtained position as the target output. Gradient descent learning was also used, as it is not only simpler but also particularly suitable for linear equations like this.

In this work, an artificial neural network was employed to overcome the computational obstacles that exist for simulating the neutron transport equation for the single velocity diffusion model, to demonstrate the possibility of using neurocomputational methods to solve neutron equations. Nevertheless, it was hoped that the results would demonstrate that in the worst case, the neural network can describe neutron diffusion with the same fidelity as current stochastic methods, requiring only a fraction of the computation time.

However, it's worth noting that the application of neurocomputational methods should not be done lightly. Artificial neural networks have at their core an unsolved problem, the black box problem. In a given convolutional neural network, regardless of its depth, it becomes impossible to determine how decisions are being made, effectively hiding decision-making within a "black box" [Rawashadeh, 2023]. Thus, it is strongly discouraged to apply neural methods to any problems whose modeling has not yet been unraveled. Although challenging, neutron equations satisfactorily describe the observed behavior within a nuclear reactor and, therefore, serve to avoid this problem.

1.1. Objective

The central objective of this work is to assess the benefits of using artificial neural networks for neutronic calculations by applying a 4-dimensional multilayer perceptron neural network to the calculation of neutron transport in the single velocity diffusion model of a nuclear reactor.

2. MATERIALS AND METHODS

The modeling of the problem is remarkably straightforward. Firstly, we know the reactor source, and given its model, we know the position it occupies. We also know, with a certain degree of accuracy and precision, thanks to experimental data and stochastic data, the final position of the neutrons. This gives us a version of neutron density within the volume determined by the reactor's sensors.

The task of the neural network is simple: to estimate a value within the allowable error range for the neutron density in each of the volumes or detectors, given the source data.

The geometry of the problem is primarily determined by the experimental data used. In particular, the ReGal project uses radial geometry, as illustrated in Image 2.

2.1 Network topology modeling

The modeling of the neural network starts with an understanding of the inputs in Equation 1, the neutron transport equation, with the goal of incorporating them into the neural network as the input vector. The input values in the equation are:

$$\begin{aligned}\frac{\partial n}{\partial t} &+ v\hat{\mathbf{\Omega}}\cdot\nabla n + v\Sigma_t n(\mathbf{r},E,\hat{\mathbf{\Omega}},t)\\ &= \int_{4\pi} d\hat{\mathbf{\Omega}}' \int_0^{\infty} dE'\, v'\Sigma_s(E'\rightarrow E,\hat{\mathbf{\Omega}}'\rightarrow\hat{\mathbf{\Omega}})\, n(\mathbf{r},E',\hat{\mathbf{\Omega}}',t) + s(\mathbf{r},E,\hat{\mathbf{\Omega}},t).\end{aligned}$$

The values are:

Position Vector r: $\mathrm{r} = [\mathrm{r}, \varphi, \phi]$.

Orientation: $\Omega' = [\alpha, \varphi, \phi]$

Time: $\mathrm{t} = [0, \mathrm{t_f})$

2. Energy $\mathrm{E}' = \mathrm{E_n} \in \mathrm{dE}$

However, since the neural network simply uses the source as an input, we need to use the position and energy. The position is zero, as the space is being understood radially. As for energy, it mainly depends on the type of fuel in the rod and the nature of emissions. For each fuel rod, a discrete set of possible energies must be assumed, from which neutrons ultimately choose one. It should be noted that despite the observed mean free path of neutrons, there is a noticeable zig-zag trajectory due to collisions, which can significantly affect the final distance [Duderstadt, 1970].

The output vector consists of ten sections along the main diagonal inside the reactor, with each output node corresponding to a segment. For each node, a neutron flux value is determined.

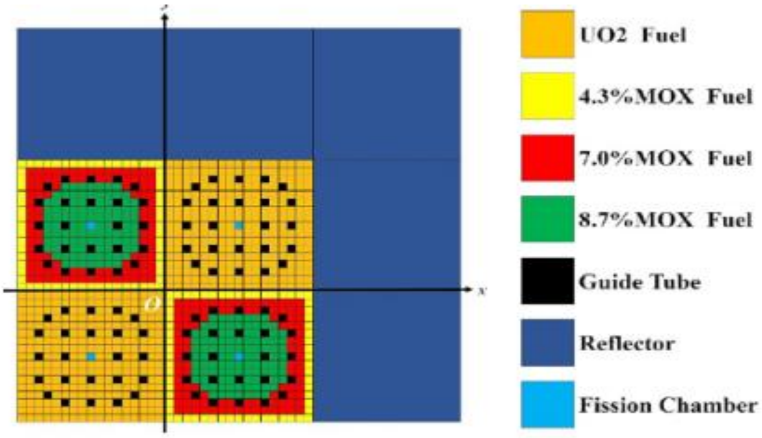
The output vector is described as follows:

$$\mathrm{Y_k} = [\phi_1, \phi_2, \phi_3, \phi_4, \phi_5, \phi_6, \phi_7, \phi_8, \phi_9, \phi_{10}]$$

where, $\phi_i(\mathrm{r_i}, \mathrm{E}, \mathrm{t}) = \mathrm{vN}(\mathrm{r_i}, \mathrm{t})$, and so that every i corresponds to a section along the main diagonal.

The geometry of the problem is best described in figure 2. It is through this setup that the data is collected and explored in the network.

Figure 2: reactor’s Geometry



Source: Hognhangh, 2023

Indeed, artificial neural networks can be mathematically described in such a way that each node Yj in the i layer is composed of a linear combination of its inputs multiplied by its weights, with the addition of a bias term. Mathematically, this is represented as described in Equation 2:

$$Y_{ji} = g(\sum_0^n \quad w_{ij}x_{ij}) \qquad \text{eq. 2}$$

Where g is the appropriate activation function for the chosen learning process.

The preferred architecture includes four layers: the first layer is composed of two inputs, the second layer with 3 nodes, the third layer with 4 nodes, and the output layer with 10 nodes.

2.2 Learning and activation functions

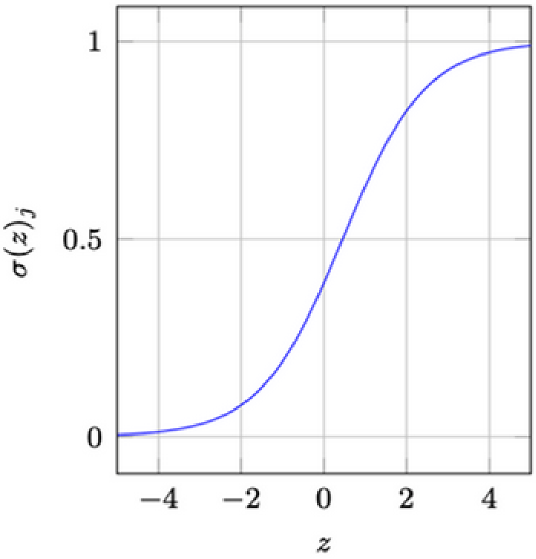
The choice of activation function for this neural network was based not only on the suitability of the employed learning method, namely, gradient descent but also on favoring the normalization of its values.

The output is described by density, and its ideal normalization is achieved through a ratio of the maximum observed neutron flux given a maximum macroscopic cross-section.

The distance along the diagonal is also represented by a ratio, this time of the maximum distance. However, energy is represented by the ratio of the chosen energy to the maximum possible energy, except that it is always a discrete value.

It is observed, therefore, that in general, there is a low expectation of encountering negative values, and thus the Softmax activation function can confidently be employed, as shown in Figure 3:

Figure 3: Sigmoidal Function Graph.



This can be described by the following equation:

$$\sigma(\vec{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$$

eq6

This always constrains the values within the neural network to be between zero and one and thus serves our needs.

In addition to that, in this neural network, the ReLU activation function was also used for all layers except the last one, which generates the outputs.

The learning method employed is simply backpropagation and gradient descent.

Essentially, once the neural network estimates a value for the neutron flux at a certain position along the diagonal, the target values will be either above or below. The neural network adjusts its weights and estimates again until it converges to a solution.

For the implementation of gradient descent, a loss function is first established to allow the neural network to take a step in the direction of the greatest reduction in error. This is achieved using a derivative that aims to minimize the error function for each of the variables. In each iteration, called an epoch the neural network reevaluates this multivariate derivative with all its updated weights.

The updated weights are described in Equation 7.

$$\Delta w_{ij}^{(\tau)} = -\eta \left. \frac{\partial E^q}{\partial w_{ij}} \right|_{w^{(\tau)}} + \mu \, \Delta w_{ij}^{(\tau-1)} \, . \qquad \text{eq7}$$

Where E is the error function, and there are known values called hyperparameters, specifically the learning rate η and the momentum μ.

The error function used is the sum of squared errors, as shown in Equation 8.

$$E = \sum_{q=1}^{n} E^q, \quad E^q = \frac{1}{2} \sum_{k=1}^{c} \{y_k(\mathbf{x}^q; \mathbf{w}) - t_k^q\}^2 \, , \qquad \text{eq8}$$

This choice is based on its simplicity and statistical origin.

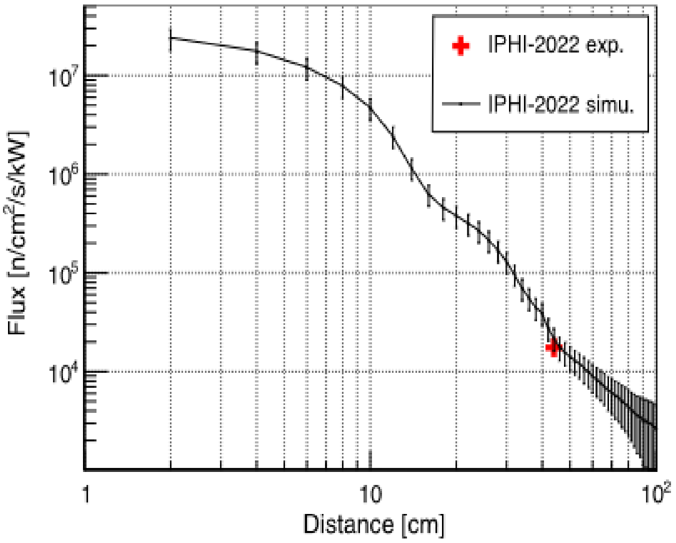
2.3 Data harvesting and treatment

The experimental data used comes from the ReGal project at the Belgian Nuclear Research Center and was obtained experimentally by placing sensors radially along the reactor [Govers, 2018].

The stochastic source data was extracted from studies of modern algorithms for solving the neutron transport equation, namely, the TOUCANS algorithm [Thulliez, 2023] and ROM [Honghang, 2023]. Both provide maps of the observed neutron flux along the main diagonal of the reactor.

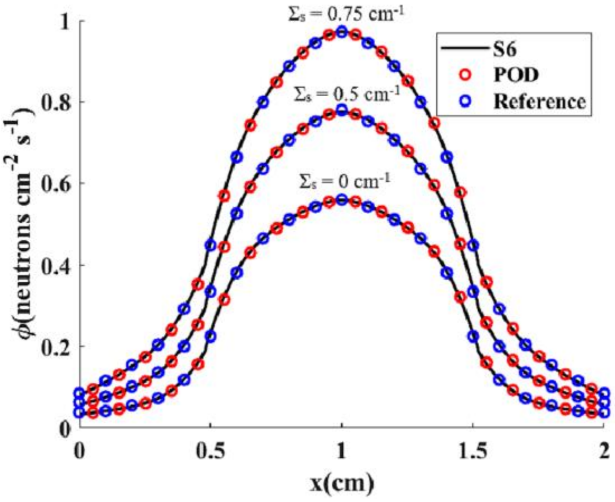
The data was manually extracted from the published papers and then fed into the neural network. The data obtained from TOUCANS and ROM are depicted in Figures 4 and 5, respectively.

Figure 4: Data TOUCANS



Source: Thulliez, 2023

Figure 5: Data ROM

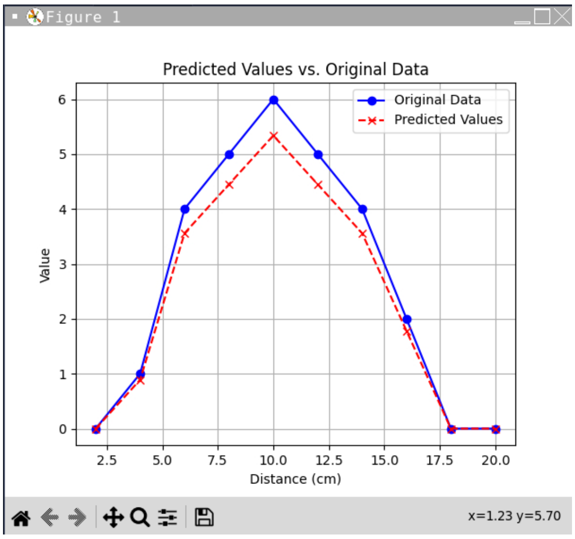


Source: Honghang 2023

3. RESULTS AND DISCUSSION

The results obtained through training the artificial neural network are presented in Figure 8, which illustrates the neutron flux distribution along the main diagonal, like what would be achieved using a modern stochastic method with up to 90% accuracy.

Figure 6: Neutron flux along the diagonal.



It's important to note that seeking higher precision than this with any databases can potentially overfit the network and impair its performance with data outside of its training set, which would be the case in real-world applications.

3.1. Data interpretation

The data displayed in Figure 6 were obtained through training that used data from stochastic methods. Despite the model's ideal accuracy of 90%, it is essential to be aware that it carries the error of these methods. Furthermore, it is crucial to note that the databases used are not ideal for applications, as they were manually extracted from other works instead of being properly collected experimentally from a reactor and refined. The same can be said about the architecture employed, as other architectures would likely yield more suitable results, given that the neutron transport equation has a time dependence and is a differential equation. Higher levels of success could easily be achieved using convolutional neural networks, recurrent neural networks, or a hybridization of the two, as they are better equipped to solve this type of problem than the equipped to solve this type of problem than the neural network used in this work.

With that said, it is worth acknowledging the success of employing neurocomputational techniques for solving the neutron transport equation. Based on the data obtained, it is possible to generate results with a fraction of the computational power typically required. As the methodology evolves, it is conceivable that such an approach could be significantly advantageous for the analysis and modeling of reactors.

4. CONCLUSIONS

Therefore, the training of the artificial neural network has shown promising results in approximating neutron flux distribution along the main diagonal, as illustrated in Figure 8. Achieving up to 90% accuracy comparable to modern stochastic methods is a significant achievement. However, caution is needed when aiming for higher precision, as it may lead to overfitting and diminish the network's performance on real-world data. The reliance on databases derived from stochastic methods and less-than-ideal experimental data highlights the potential for improvement through alternative architectures such as convolutional neural networks or their hybrids. Despite these challenges, the application of neurocomputational techniques offers a compelling path to enhancing the efficiency and accuracy of reactor analysis and modeling, paving the way for future advancements in this critical field.

ACKNOWLEDGMENT

Thanks to CNPQ-PIBIC & Institute of Nuclear Energy Research for supporting this article.

FUNDING

CNPQ-PIBIC

No funding award number, just a month support.

CONFLICT OF INTEREST

We have no conflicts of interest to disclose.

REFERENCES

[1] Duderstadt JJ, Hamilton LJ. Análise de Reatores Nucleares. 2nd ed. Rio de Janeiro: LTC; 2005.

[2] Bishop CM. Neural networks and their applications. Revista de Instrumentos Científicos. 65(6):1180-1183. 1994.

[3] Honghang Chi, Yahui Wang, Yu Ma. Reduced-order with least square-finite difference method for neutron transport equation. Annals of Nuclear Energy. 191:109914. 2023

[4] Thulliez L. TOUCANS: A versatile Monte Carlo neutron transport code based on GEANT4. Nuclear Instruments and Methods in Physics Research, Section A. 1051:168-190. 2023.

[5] UM-Dearborn. AIS's mysterious "black box" problem explained. [Internet]. Available from: https://umdearborn.edu/news/ais-mysterious-black-box-problem-explained.[cited 2023 Jul 26]

[6] Govers, K., & Verwerft, M. (2017). Experimental data for neutron transport & depletion code validation. In SCALE User Group Meeting, September 26–28, 2017 (pp. 1-27). SCK•CEN. Disponível em: https://www.ornl.gov/sites/default/files/12_Kevin_Govers.pdf

LICENSE