

USING ARTIFICIAL NEURAL NETWORKS TO ACCELERATE TRANSPORT SOLVES

Mauricio E. Tano¹ and Jean C. Ragusa¹

¹ Department of Nuclear Engineering, Texas A&M University,
College Station, TX 77840, United States

mtano@tamu.edu, jean.ragusa@tamu.edu

ABSTRACT

Discontinuous Finite Element Methods (DFEM) have been widely used for solving S_N radiation transport problems in participative and non-participative media. In this method, small matrix-vector systems are assembled and solved for each cell, angle, energy group, and time step while sweeping through the computational mesh. In practice, these systems are generally solved directly using Gaussian elimination, as computational acceleration for solving these small systems are often inadequate. Nonetheless, the computational cost of assembling and solving these local systems, repeated for each cell in the phase-space, can amount to a large fraction of the total computing time. In this paper, a Machine Learning algorithm is designed to accelerate the solution of local systems. This one is based on Artificial Neural Networks (ANNs). Its key idea is training an ANN with a large set of solutions to random one-cell transport problems and, then, replacing the assembling and solution of the local systems by the feedforward evaluation of the trained ANN. It is observed that the optimized ANNs are able to reduce the compute times by a factor of ~ 3.6 per source iteration, while introducing mean absolute errors between $0.5 - 2\%$ in transport solutions.

KEYWORDS: Artificial Neural Networks, Acceleration, Transport Sweeps, DFEM

1. INTRODUCTION

Machine Learning techniques are slowly paving it way into computational science. Beyond the current hype, Machine Learning techniques have proven effective in compressing and optimizing the solution for large computational scale problems [1]. Among these problems, there is the deterministic computational solution of radiation transport.

Historically, Discontinuous Finite Element Method (DFEM) has been widely used for solving S_N radiation transport problems. In this method, the transport equation is discretized into a set of algebraic equations that must be solved for each spatial cell, angular direction, energy group, and time. Each of this problem is called one-cell problem. For example, the one-cell problem consists of a system of 8 coupled algebraic equations for trilinear DFEM in 3D hexahedral cells. Reducing the problem to one-cell units is effective when reducing the memory requirements in the solution of the radiation transport problem. However, iterative accelerators, designed for large systems of algebraic equations, are usually not effective at the scale of the small local system of equations obtained for one-cell problems. Moreover, the computational cost of assembling and solving the

one-cell problem, repeated for each phase-space cell, amounts for most of the computational time in DFEM radiative transport solves. This imposes an upper computational bound on the potential acceleration of these methods when a fixed computational power is available.

However, when analyzing the local systems obtained for one-cell problems, an underlying structure can be observed in these systems because of the symmetries and directionality of the radiative transport process. In the present work, a Machine Learning algorithm is designed to exploit this structure for accelerating radiation transport solves is presented. The key idea is to replace the assembly and solution of each local system by a pre-trained ANN recursively evaluated during the sweeping process.

In the following sections are organized as follows. First, the DFEM discretization of the radiation transport problem is and the structure appearing in the local systems is analyzed by means of correlation matrices. Then, the design of a Shallow Neural Network to exploit this structure for accelerating the solution to one-cell problems is discussed. Finally, key results obtained by mean of this acceleration are presented.

2. DISCONTINUOUS GALERKIN FINITE ELEMENT METHOD IN TRANSPORT SWEEPS

The one-group S_n transport equation, with isotropic scattering and sources, is given by

$$(\boldsymbol{\Omega}_d \cdot \nabla + \sigma_t) \psi_d = \frac{\sigma_s \phi + q}{4\pi} = \frac{q_{\text{tot}}}{4\pi} \quad (1)$$

with d the direction index in the angular quadrature. The scalar flux is obtained via numerical integration: $\phi = \sum_{d'} w_{d'} \psi_{d'}$, with w_d the quadrature weight [2]. For simplicity, in Eq. (1), the total source (scattering source + extraneous source) has been introduced as the scattering source is often lagged on iteration behind, and thus merged in with the extraneous source, during a source iterations process. After the DFEM [3] spatial discretization of Eq. (1), the following local linear system for cell K in the phase-space (space, direction, energy, and time) is obtained

$$\left(\boldsymbol{\Omega}_d \cdot \mathbf{G} + \sigma_t^K \mathbf{M} + \sum_{f \in F^+} (\boldsymbol{\Omega}_d \cdot \mathbf{n}_f) \mathbf{S}_f \right) \boldsymbol{\Psi}^d = \frac{1}{4\pi} \mathbf{Q} + \sum_{f \in F^-} |\boldsymbol{\Omega} \cdot \mathbf{n}_f| \mathbf{S}_f \boldsymbol{\Psi}_f^{d,\uparrow} \quad (2)$$

where piece-wise constant material properties have been assumed. The set of faces for cell K is split into faces where radiation is incoming and outgoing, $F^\mp = \{f \in \partial K, \text{ s.t., } \boldsymbol{\Omega}_d \cdot \mathbf{n}_f \lesseqgtr 0\}$. Besides, $\boldsymbol{\Psi}$ denotes the DFEM solution in the current cell, while $\boldsymbol{\Psi}_f^\uparrow$ denotes the upwinded flux (transmitted through faces from the upwind cell neighbors). The local matrices are defined as follows:

$$\mathbf{G}^{ij} = - \int_K b_j \nabla b_i d^3 r \quad (3)$$

$$\mathbf{S}_f^{ij} = \int_f b_i b_j d^2 r \quad (4)$$

$$\mathbf{M}^{ij} = \int_K b_i b_j d^3 r \quad (5)$$

$$\mathbf{Q}^i = \int_K b_i q_{\text{tot}} d^3 r. \quad (6)$$

The DFEM basis functions on cell K are denoted by $b_i(\mathbf{r})$ and $(i, j) \in [1, N^K]$ where N^K is the number of spatial degrees of freedom for cell K . The size of the local system is $N^K \times N^K$. Here, we restrict ourselves to trilinear basis functions on hexahedral cells, so that $N^K = 8$.

In a more general form, Eq. (2) can be written as

$$\mathbf{A}_K^d \Psi_K^d = \mathbf{b}_K^d. \quad (7)$$

where \mathbf{A}_K^d is the matrix of coefficient for the local system and \mathbf{b}_K^d is the right-hand-side vector containing the incident fluxes and source. During a regular solution process, the right-hand-side \mathbf{b}_K^d is computed for each cell K , by updating the incident upwinded fluxes and scattering source from the previous iteration, and, then, system in Eq. (7) is solved by GE. In this case, this results in 729 computational operations. For simplicity, we assume equivalent computational cost in all operations, although divisions are proven to be more computationally expensive in modern processors [4]. This is the upper bound when replacing this process with the Machine Learning algorithm.

The base ground to design an accelerator based on a Machine Learning algorithm is to demonstrate that there exist a structure in problem (7) and, thus, a simpler mapping between inputs $(\mathbf{A}_K^d, \Psi_K^d)$ and output (\mathbf{b}_K^d) is achievable. In this case, in order to study the system's structure, a set random one-cell problems have been generated. When generating each random problem the internal sources, incident fluxes, total cross section, direction vector, and the cell's dimensions are randomly sampled from a uniform random distribution. Then, Eq. (2) is assembled and solved. Finally, the solution vector \mathbf{b}_K^d is collected for each of these problems. A total of 10^5 random problems are necessary to obtain statistical convergence among the second order correlations in each of the components of the discretized system Eq. (7). Therefore, in the system structure is analyzed in light of these correlations. In particular Pearson's correlations and Student's t-tests are performed as two complementary statistical measures of correlation among the data. The results obtained for the correlation coefficients between each of the entries of the system matrix (\mathbf{A}), the right-hand-side vector (\mathbf{b}), and the solution vector (Ψ) are presented in Figure 1.

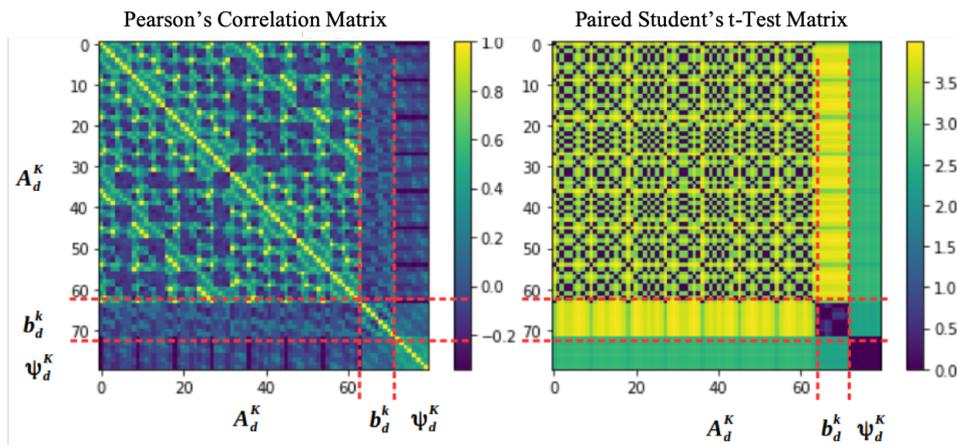


Figure 1: Pearson correlation matrix obtained for a set of 10^5 random one-cell DFEM problems.

In Figure 1 it can be observed that a large correlation exists among the coefficients in the system's matrix. This is a consequence of the way in which this matrix is assembled in Eq. (2) and because there are more coefficients than random inputs in the system. Moreover, a non-negligible correlation exists among the components of the right-hand-side vector. This results from the structure maintained in the construction of the right-hand-side vector (incident fluxes + sources). Furthermore, structures appear when looking at the correlations of these two with the solution Ψ . This is evident since the solution vector is univocally determined by the matrix of coefficients and the right-hand-side vector.

The main conclusion of this section is that there is a structure local systems obtained in the DFEM discretization of radiation transport. The of next section is to exploit this structure to reduce the number of operations currently required in the solutions of one-cell problems.

3. MACHINE LEARNING APPROACH FOR ACCELERATING TRANSPORT SWEEPS

The key idea of this work is to replace the assembly and solution by GE of system Eq. (2) with a shallow Artificial Neural Network (ANN). Note that for a sweeping process the same trained ANN will be providing the solution for each cell in the phase-space. This means that the sweeping process is replaced by a staggered recurrent evaluation of the ANN.

The architecture of this ANN is to be optimized in order to reduce the number of operations involved in its feedforward evaluation, while keeping a bounded error in the solution. The resulting ANN, is presented in Figure 2. It consists of an ANN with only one hidden layer (*shallow ANN*). Hyperbolic tangents have been used as activation functions in the hidden layer, while Exponential Linear Units (ELUs) has been used as activation functions for the output layer. The reason for first ones is that an hyperbolic tangent provide the mixed linear-exponential behaviour as a function of its inputs that is required in the solution to transport problems. Moreover, ELUs can scale the outputs of the ANN without a bound and, thus, provide the ANN the ability to represent any solution for the angular fluxes (unlimited positive range). The key part of acceleration arrives when determining appropriate inputs for the ANN. Rather than feeding in the linear system coefficients (matrix and vector entries), the input layer takes in as inputs the variables determining the solution of linear system. Specifically, these are: (i) the incident fluxes over the three incident faces of a sweeping octant (Ψ_{inc}^d), (ii) the total source term in the DFEM nodes within the cell (q), (iii) two independent direction cosines determining the sweeping angle (μ, η), (iv) the three factors determining the sizes of the hexahedral cells ($\Delta x, \Delta y, \Delta z$), and (v) the total cross section in the cell (σ_t). Thus, the resulting ANN has 26 inputs, instead of the 72 inputs that will be required if the coefficients of the system's matrix and right-hand-side vector had been used.

To fundament this idea, a comparison of the number of operations required for the general assembly and solution by Gauss elimination of system is compared against the number of operations required for the feed-forward evaluation of the ANN in Figure 3. It is observed that the feed-forward evaluation of the ANN may allow to reduce the number of operations by a factor of ~ 3.2 , when using 8 neurons in the hidden layer as shown in Figure 2. The number of neurons in this hidden layer was set by balancing the ANN fitting error produced by solving local systems with the ANN versus the acceleration provided by this one.

At this points, three points that differ from classical Machine Learning approaches that are necessary

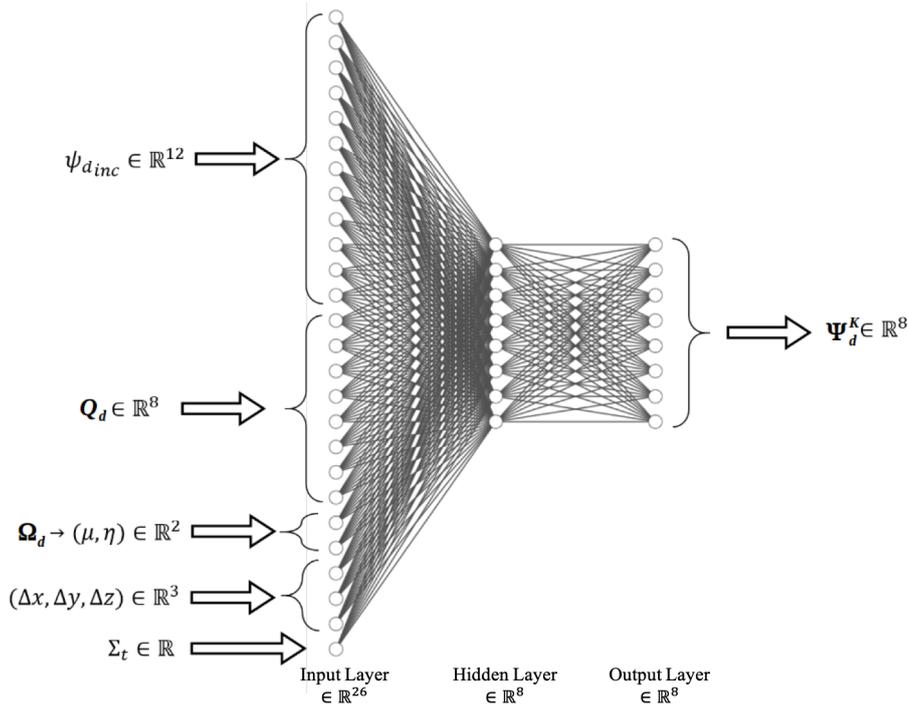


Figure 2: Concept of feed-forward ANN used in the present study.

to obtain accurate solutions and improve acceleration when building a system based in this ANN: (i) appropriate definition of a regularized loss function for training the weights of the ANN, (ii) suitable algorithm to randomly build the training set avoiding the introduction of biases that can cause overfitting, (iii) application of an adapted DropConnect that allows accelerating the feedforward evaluation of the ANN. The first point is discussed here since it is of paramount importance in this work. The reader is referred to our paper [5] for more information on the other two.

The training of the ANNs in each direction is performed on a set of 9×10^4 solutions to one-cell problems and the testing set consists of 1×10^4 equivalent ones. The training is performed by backpropagation using the Adam optimization algorithm. For training and testing, the regularized loss function is defined as follows:

$$\tilde{C} = \frac{1}{P} \sum_{p=1}^P \frac{\|\Psi_{d,p} - \Psi_{d,p}^r\|_1}{\|\Psi_{d,p}^r\|_1} + \frac{\lambda_1}{P} \sum_{p=1}^P \sum_{i=1}^8 \frac{|\Psi_{d,p,i} - \Psi_{d,p,i}^r|}{\gamma_1 |\Psi_{d,p,i}^r| + \gamma_2} + \frac{\lambda_2 e^{-\tau}}{\tau + \gamma_3} \frac{1}{P} \sum_{p=1}^P \|\Psi_{d,p} - \Psi_{d,p}^r\|_1. \quad (8)$$

where τ is a characteristic optical length of the cell, taken in this case as $\tau = \frac{\Sigma_t \Delta}{\mu}$, with the cell size parameter $\Delta = \sqrt{\Delta x^2 + \Delta y^2 + \Delta z^2}$ and μ the director cosine in the \hat{x} -direction. Furthermore, the parameters λ_1 , λ_2 , γ_1 , γ_2 , and γ_3 are fitted to obtain the required behavior in the ANN. The values used in this case are presented in Table 1. The first term in the right-hand-side of Eq. (8) is a classical quadratic loss function. The second one is a Lasso regularization that assigns higher weights to errors produced by the ANN when the fluxes are small in magnitude. This one is introduced because large relative errors in small components of the solution vector are hindered in the classical quadratic definition of the loss function in Eq. (8). In our case, this hindering results

Comparing the numbers of operations between DGFEM and the ANNs
 for one computational cell, one direction, one energy group and time step

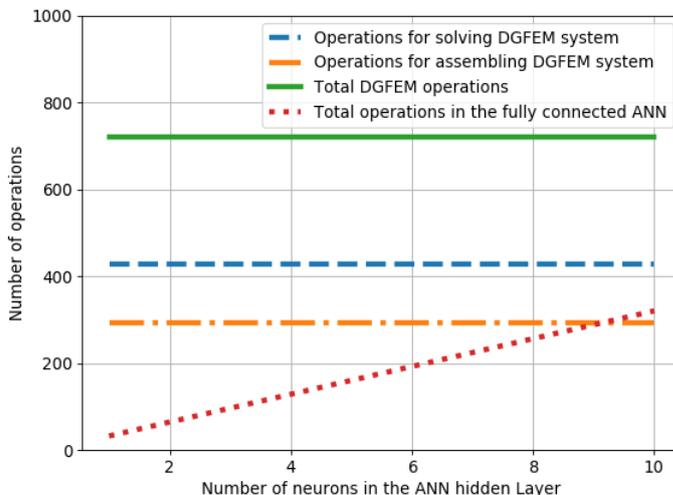


Figure 3: Comparison of the number of operations required for the assembly and solution by Gauss Elimination of system vs the number of operations required for a feed-forward evaluation of the ANN.

λ_1	λ_2	γ_1	γ_2	γ_3
0.15	0.2	10.0	0.01	0.1

Table 1: Parameters used in the regularizer, Eq. (8)

in an error buildup when recurrently evaluating the ANN in a sweeping process. The third term is also a Lasso regularization parameter that assigns higher values for errors produced in optically thin problems. This regularizer channels the ANN to find solutions by compressing the inputs in the hidden layers of the ANN towards zero. The parameters in both regularizing terms have been fitted after a sensitivity study.

In the following section, the performance of a trained ANN in replacing the solution to one cell problems is analyzed for a benchmark case.

4. RESULTS

In this paper, we present the results of applying the trained ANNs presented in the previous section to benchmark Problem 1 proposed in [6]. The geometry of the Benchmark problem is shown in Figure 4. The problem has one internal source (Region I), followed by a void region (Region II), and a thick absorber (Region III). The parameters for the three regions are shown in Table 2. This

benchmark case was selected since the ANN could be tested in limiting cases with very different properties, maximizing the probability of error the buildup in the recurrent evaluation of the ANN during a transport sweep. The 3D domain was discretized in $100 \times 100 \times 100$ cells. Furthermore,

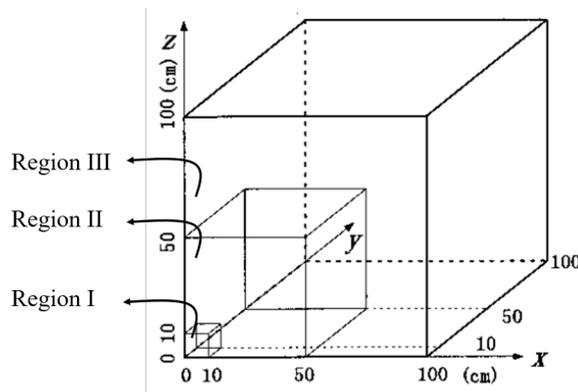


Figure 4: Geometry of the benchmark used.

4 different ANNs were trained for the S_2 , S_4 , S_6 , and S_8 angular quadratures. The average error for each discretization is shown in Table 3. These errors are larger than the ones obtained during the training and testing of the ANN for one cell problems ($\sim 10^{-4}$). This indicates that an error build-up exists when recurrently evaluating the ANN. However, due to the regularizer in Eq. (8) the error buildup is small enough to yield acceptable solutions. Subsequent work will focused on techniques to further reducing this error. A large speed-up in computing time is obtained when employing the ANN. This one is produced by two factors. First, the reduction in the number of operations required with the ANN (factor 3.2). Then, the most rapid convergence of the solutions produced with the ANNs during source iterations due to the noise in the connections of the ANN, which act with a diffusive character when recursively evaluating the network (factor 4.9). The reader is referred to our paper [5] for more detailed testing cases for this ANN.

Region	$S [ncm^{-3}s^{-1}]$	$\sigma_t [cm^{-1}]$	$\sigma_s [cm^{-1}]$
I	1	0.1	0.0999
II	0	10^{-4}	5×10^{-5}
III	0	0.1	0.0999

Table 2: ANN results for the Benchmark problem [6] with $c = 0.999$.

5. CONCLUSION

In the present work an ANN is proposed for accelerating radiation transport solves. This one replaces the local system solution for the one-cell problems obtained in the DFEM discretization of radiation transport. Thus, the sweeping problem to solve radiation transport is replaced by the recurred evaluation of the ANN. The ANN structure has been optimized by taking a reduced set

S_n Order	Mean diff. between ANN & GE	Speedup Factor per Digit Accuracy
2	1.20%	15.8
4	1.83%	15.7
6	2.22%	15.8
8	2.22%	15.8

Table 3: ANN results for the Benchmark problem [6] with $c = 0.999$

of parameters as inputs that fully determines the solutions to the one-cell problems. Moreover, accurate solutions are obtained by applying Lasso regularizers to the loss function when training the ANN. In this paper the results obtained with an ANN trained with 9×10^4 one-cell random radiative transfer problems and tested on 1×10^4 equivalent ones are presented. This ANN is then introduced in a large scale 3D problems used as benchmark, where it is observed that replacing the solution of one-cell problems by the recurrent evaluation of the ANN introduces error between $\sim 1 - 2\%$, while introducing an acceleration factor of $\sim 15 - 6\%$. Our future work is oriented to improving the error in the ANNs, developing hybrid ANN-exact solutions, and using the ANNs as low order operators in High-Order Low-Order schemes in radiative transfer.

REFERENCES

- [1] S. Raschka. “MLxtend: Providing machine learning and data science utilities and extensions to Python’s scientific computing stack.” *J Open Source Software*, **volume 3**(24), p. 638 (2018).
- [2] E. E. Lewis and W. F. Miller. “Computational methods of neutron transport.” (1984).
- [3] W. H. Reed, T. Hill, F. Brinkley, and K. Lathrop. “TRIPLER: A two-dimensional, multigroup, triangular mesh, planar geometry, explicit transport code.” Technical report, Los Alamos Scientific Lab., N. Mex.(USA) (1973).
- [4] T. Güneysu and C. Paar. “Ultra high performance ECC over NIST primes on commercial FPGAs.” In *International Workshop on Cryptographic Hardware and Embedded Systems*, pp. 62–78. Springer (2008).
- [5] M. Tano and J. Ragusa. “Acceleration of Radiation Transport Solves Using Artificial Neural Networks.” *arXiv preprint arXiv:190604017* (2019).
- [6] K. Kobayashi, N. Sugimura, and Y. Nagaya. *3-D radiation transport benchmark problems and results for simple geometries with void regions*. Nuclear Energy Agency (2000).