IMPROVEMENT OF RADIATION DOSE ESTIMATION DUE TO NUCLEAR ACCIDENTS USING DEEP NEURAL NETWORK AND GPU

Filipe S. M. Desterro[1], Adino A. H. Almeida[1] and Claudio M. N. A. Pereira[1]

Instituto de Engenharia Nuclear (IEN / CNEN - RJ)
Cidade Universitária - R. Hélio de Almeida, 75 -
21941-614 Ilha do Fundão, Rio de Janeiro - RJ
filipesantana18@gmail.com
adino@ien.gov.br
cmcoelho@ien.gov.br

ABSTRACT

Recently, the use of mobile devices has been proposed for dose assessment during nuclear accidents. The idea is to support field teams, providing an approximated estimation of the dose distribution map in the vicinity of the nuclear power plant (NPP), without needing to be connected to the NPP systems. In order to provide such stand-alone execution, the use of artificial neural networks (ANN) has been proposed in substitution of the complex and time consuming physical models executed by the atmospheric dispersion radionuclide (ADR) system. One limitation observed on such approach is the very time-consuming training of the ANNs. Moreover, if the number of input parameters increases the performance of standard ANNs, like Multilayer-Perceptron (MLP) with backpropagation training, is affected leading to unreasonable training time. To improve learning, allowing better dose estimations, more complex ANN architectures are required. ANNs with many layers (much more than a typical number of layers), referred to as Deep Neural Networks (DNN), for example, have demonstrating to achieve better results. On the other hand, the training of such ANNs is very much slow. In order to allow the use of such DNNs in a reasonable training time, a parallel programming solution, using Graphic Processing Units (GPU) and Computing Unified Device Architecture (CUDA) is proposed. This work focuses on the study of computational technologies for improvement of the ANNs to be used in the mobile application, as well as their training algorithms.

1. INTRODUCTION

Tools that provide fast and accurate information for an emergency response team at the field, can be the differential, in the ability of these professionals to make the right decision in a short period of time. For this, a mobile application [1] was developed to assist the teams in the event of an accident at a nuclear power plant (NPP), with a dispersion of radionuclides in the air, determining the dose of radiation in the vicinity. This application used the Artificial Neural Network (ANN) to predict doses of radionuclides in the vicinity of the NPP, facilitating the decision-making by the rescue teams, determining the priority areas for care.

This work will focus on decreasing training time and error in dose predictions. Due to the large data set, the CPU is not able to train the ANNs in a timely manner. Because of this, the Deep Neural Network (DNN) will be used as the learning model, which is designed to use a large volume of training data. The use of Graphics Processing Units (GPU) is what makes the difference in this model because without the GPU it would be difficult to train DNN.

For the use of the DNN together with GPU, the Tensorflow will be used. Tensorflow is a framework, developed by the team of researchers from Google [4,5], which uses the Computing Unified Device Architecture (CUDA) to improve training.

The Research has as its main objectives:

I. Use the Tensorflow framework to aid in the development of the machine learning code.
II. Use of DNN as a learning model.
III. Describe the system architecture.
IV. Estimate differences and improvements of the proposed system with the realized ones.

During the research, it was necessary to evaluate several DNN configurations, until the one with the best computational cost and smaller errors in the prediction was found.

The paper is organized into 6 sections: Section 2 shows the fundamentals of the research. Section 3 presents the tools and methodologies used. Sections 4 and 5 describe the implementation and evaluation of the model. Section 6 presents final considerations.

2. RELATED WORKS AND CONTEXTUALIZATION

Here the focus will be on two characteristics of this work, which are: A: The use of DNNs in radiation dose prediction; B: Use of the Tensorfow framework in the production of the DNN architecture.

The training of neural networks for dose determination in accident areas has already obtained solutions with other machine learning tools, such as the research Development of a mobile dose prediction system based on an artificial neural network for NPP emergencies with radioactive material release [1]. Two means of machine learning to determine the radiation dose were used, the multi-layer perceptrons (MLP) architectures with backpropagation training algorithms that had a training time of 4:30h. Another test was performed with the General Regression Neural Network (GRNN) [11] optimized by Genetic Algorithm (GA) [12] and had a training time of 00: 19h, but despite the reduced time, it did not show good Results. All these tests were implemented using the CPU for training and learning.

Currently, there is a system dedicated to dose prediction (Atmospheric Dispersion Systems (ADS), which is used at the Brazilian NPP, which use complex mathematical and physical models, executed by supercomputers. The dose forecast is made from the atmospheric releases of radionuclides, made by the NPP and usually they generate the projections between one and two hours after the accident.

3. PROPOSED APPROACH

In this research will be used a framework to investigate the training of deep neural networks. In this section, the tools and algorithms used will be described briefly.

3.1. Deep Neural Network (DNN)

The Deep Neural Network (DNN) operates with a trait model very similar to the one used in the Multilayer Perceptron Network (MLP) [2]. However, the DNN training process can become complex. Considering, that each layer of nodes trains with a different group of resources, based on the exit of the previous layer. As result, the complexity of the training increases proportionally to the number of layers in the network[3].

Nevertheless, in the recent years, with improvement of the activation functions algorithms and the advancement of the parallel computing hardware, more specifically with the processing power of the GPUs that have multiple cores capable of executing instructions in parallel, being exponentially more efficient than normal processors, that work sequentially and have few cores, to training the multiple layers of nodes of a DNN.

3.2. TensorFLow

Tensorflow is a framework developed by Google's machine learning team, which seeks the facilitation of the development of machine learning algorithms to solve different problems that use a large amount of data. The framework focuses on training applications based on deep neural networks and has been used on many platforms, from data centers to mobile devices. Because of the high-level application development language used, by the Tensorflow, different learning models and training algorithms can be utilized without changing the core of the system. Thus, it is possible to create an easy and fast system of new learning algorithms of machines that can be installed in several devices [4].

3.2.1 Dataflow graph

For TensorFlow representation, a single data flow graph is used. In each part of the data flow graph, the computation and learning algorithms can be observed. In each graph, a vertex can be found in the representation of an operation, which can contain no inputs or several inputs and outputs [5]. The Tensorflow graph has two characteristics that differ from contemporary models:

- Support to multiple simultaneous executions of your subgraphs.
- Its individual vertices can have variable status and can be shared between different moments of execution.

With its characteristics, Tensorflow becomes a very versatile framework for both research and industry compared to other similar frameworks (Caffe, Theano, and Torch)

4. IMPLEMENTING DEEP NEURAL NETWORK IN TENSORFLOW

4.1. Hardware

To carry out the investigation it was necessary to use hardware components that allow the use of the framework, they are:

Table 1: Hardware used for processing

Video Card	Geforce GTX 1050_{Ti} 4GB 128 Bit GDDR5
CPU	Intel® Core™ i7 CPU 960 @ 3.20GHz × 8
RAM	8 GiB

4.2. Deep Neural Network Architectures

In Deep Neural Network will be used the backpropagation to train the network, where its signals are propagated to fronts going through all the layers, beginning with the first layer until the last one. The training occurs to predict the dose based on the spatial position (X, Y), so in order to validate the process, a small group of meteorological variables, will be utilized: wind direction, wind speed, X position and Y position.

During the execution of the training, the focus was on finding a more optimized architecture, with the goal of finding better results and a shorter training time compared to the previously mentioned work [1]. In this network was implemented, a configuration of an input layer, seven hidden layers and one output layer, shown in table 2.

Table 2: Activation function architecture

Activation Function	Neurons	Number Layer
Input	4	1
Relu	256	2
Relu	128	3
Relu	64	4
Elu	32	5
Elu	16	6
Elu	8	7
Elu	4	8
Output	1	9

4.2.1 Activation function

4.2.1.1 Rectified linear unit (ReLU)

It was first proposed by Restricted Boltzmann Machines [8], the ReLU is formally defined by:

$$y_i = \begin{cases} x_i & if\ x_i \geq 0 \\ 0_i & if\ x_i \leq 0 \end{cases} \quad (1)$$

Its main advantage is the solution to the problem of gradient disappearance in the networks of many layers [9, 10]. However, Relu shows that the mean activation will be greater than zero [8].

4.2.1.2 Exponential linear unit (ELU)

The difference between ReLU and Exponential Linear Units (ELU) is an exponential function and has an average activation of neurons close to zero and this activation method accelerates the learning process because it brings the closest gradient to the natural gradient.

Mathematic ELUs are presented:

$$f_{(x)} = \begin{cases} x_i & if\ x > 0 \\ \alpha\left(e^x - 1\right) & if\ x \leq 0 \end{cases} \quad (2)$$

For positive values at x, the ELU will behave as ReLU, but if x is negative, it sets the value of x = -1 and α = 1. This helps in learning because it stimulates the mean activation of the neurons to be close to zero[6, 7].

4.3 Datasets used in training

For this model, three datasets were used to perform the training, network testing and validation with production data. These data were obtained through accurate simulations using the atmospheric dispersion system (ADS), used at the NPP. Each dataset contains different examples.

Table 3: Used datasets

Dataset	Number of Sample
Training	3825
Test	2448
Production	6120

5. PROGRAMME IMPLEMENTATION AND EVALUATION

Table 4 shows the results obtained in DNN training and evaluation.

Table 4: Training Results

	Training set	Test set	Production set
Number of patterns	3825	2448	6120
Mean absolute error	0.0913601	1.01615	0.799963
Max absolute error	2.38262	50.2283	88.6772
Median absolute error	0.00521219	0.00594139	0.00548756
Mean squared error	0.0574086	21.1855	17.1109
r2 score (coef determination)	0.0574086	0.974079	17.1109
Learning epochs	40,000		
Training time	00:02:44 minutes		

During the training it was observed that from five thousand times the dose prediction using the test data did not improve significantly.

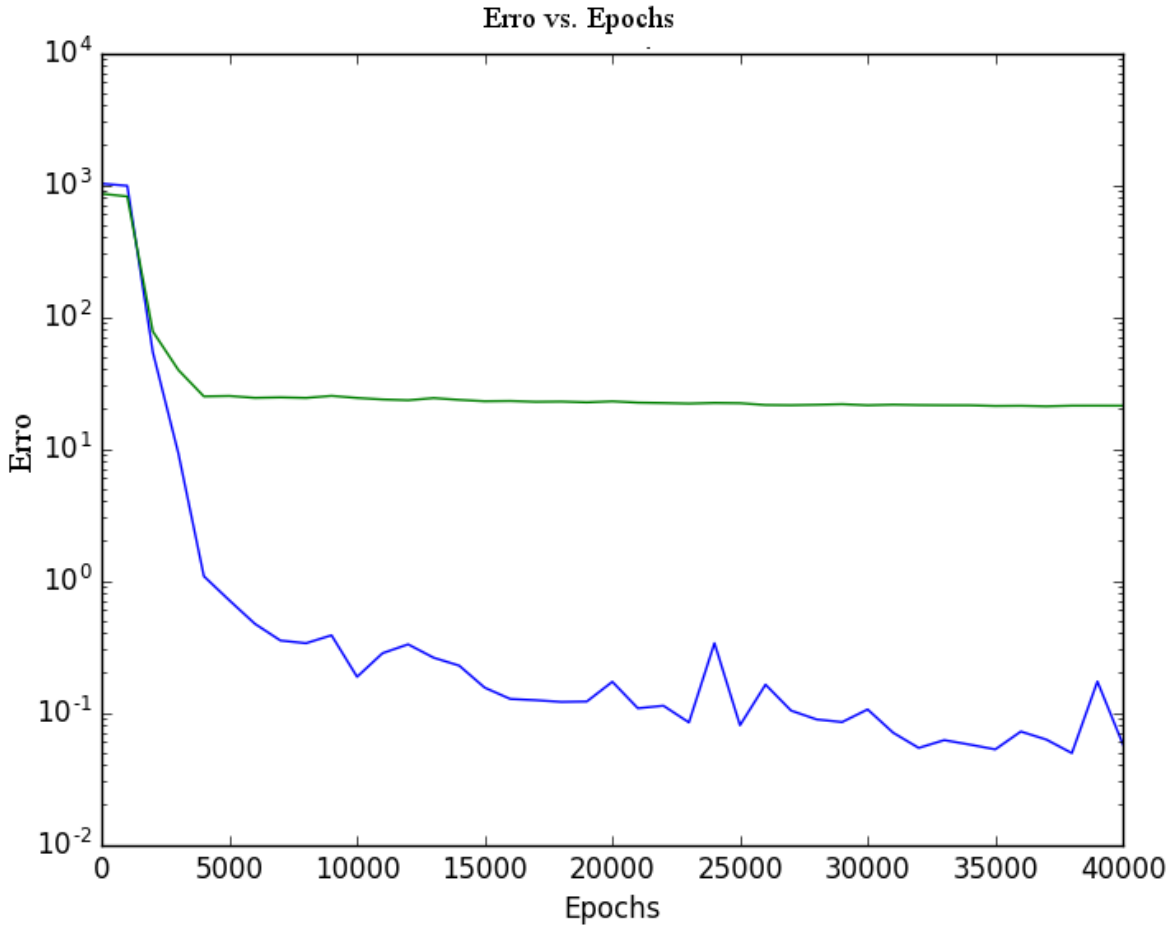

Fig. 1: Progress of average error of training and test data (Blue line: training data; Green line: test data).

The result of the mean error of the prediction is compared in figure 1. It was verified that a 23% smaller error than other networks trained by the CPU [1] was obtained, thus improving the dose prediction in accident areas.

Fig. 2: Here we show the final difference of the mean absolute error in the data set in the production of neural networks trained in the CPU and the DNN executed in the GPU.

There was also a speedup of 110 in network training and comparison with 5L-MLP, and in relation to GRNN a speedup of 7, as seen in figure 2.

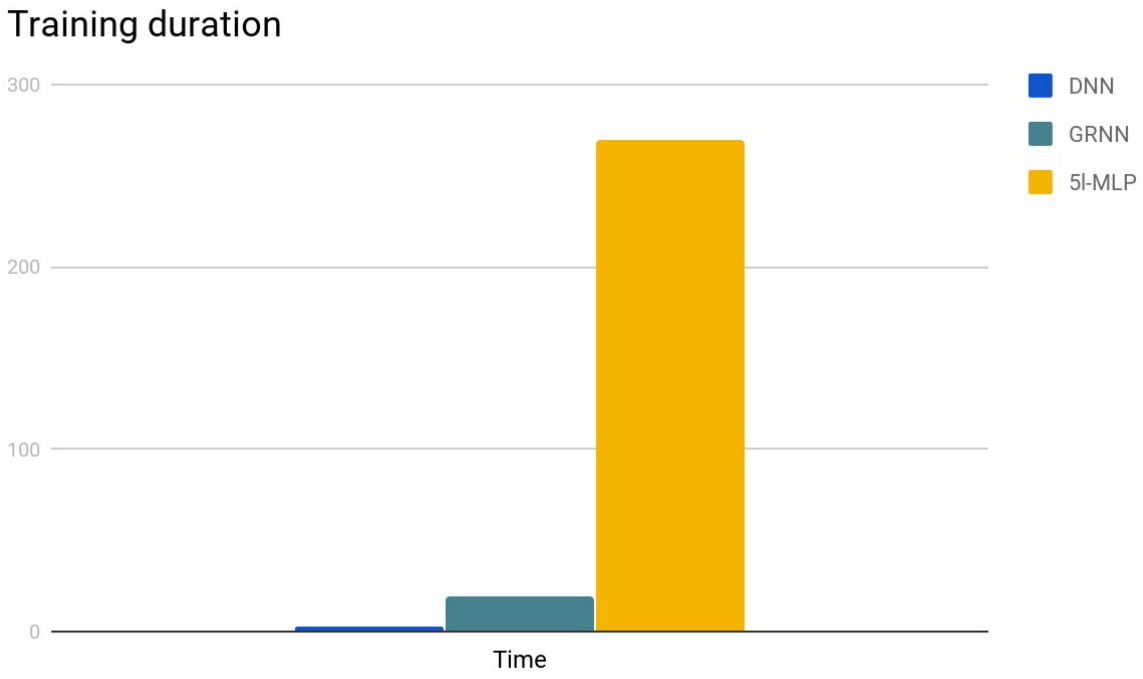

Fig. 3: Difference of times in the training of neural networks executed in the CPU and the DNN executed in the GPU

6. CONCLUSIONS

In this work, a deep neural network training model was presented and evaluated. Its development was based on works already done [1], seeking to improve the given solution.

It was observed that DNN was much superior in the training, where it took a few minutes to perform it, reaching an average error lower than the networks trained in other models.

Thus, it was verified that DNN with the framework Tensorflow proved to be great tools for the development of the solution for engineering problems. For this same problem, in the future the number of atmospheric characteristics can be increased, thus increasing the complexity of the DNN to seek to find minor errors of the dose prediction.

REFERENCES

1. Claudio M.N.A. Pereira, Roberto Schirru, Kelcio J. Gomes, Jose L. Cunha, "Development of a mobile dose prediction system based on artificial neural networks for NPP emergencies with radioactive material releases", *Annals of Nuclear Energy*, **Vol. 105**, pp.219-225 (2017).
2. Ayhan B., Kwan C, "Application of Deep Belief Network to Land Cover Classification Using Hyperspectral Images", *Advances in Neural Networks - ISNN 2017. Lecture Notes in Computer Science*, **Vol. 10261**, pp. 269-276 (2017).
3. Deeplearning4j Development Team. Deeplearning4j: Open-source distributed deep learning for the JVM, Apache Software Foundation License 2.0. http://deeplearning4j.org.
4. Martin Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, Manjunath Kudlur, Josh Levenberg, Rajat Monga, Sherry Moore, Derek G. Murray, Benoit Steiner, Paul Tucker, Vijay Vasudevan, Pete Warden, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng, Google Brain, "TensorFlow: A System for Large-Scale Machine Learning", Savannah, Georgia, United States of America, November 2–4, pp.265-283, https://www.usenix.org/conference/osdi16/technical-sessions/presentation/abadi (2016).
5. Martin Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Rafal Jozefowicz, Yangqing Jia, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mane, Mike Schuster, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viegas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. "TensorFlow: Large-scale machine learning on heterogeneous systems", (2015). Software available from tensorflow.org.
6. Djork-Arne Clevert, Thomas Unterthiner, Sepp Hochreiter, "Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs)", *International Conference on Learning Representations (ICLR-16)*, San Juan, Puerto Rico, May 2-4, (2016).
7. Bing Xu, Naiyan Wang, Tianqi Chen, Mu Li, "Empirical Evaluation of Rectified Activations in Convolution Network", *Deep Learning Workshop (ICML-15)*, Lille Grande Palais, France, July 10-11, (2015).
8. Nair, Vinod and Hinton, Geoffrey E. "Rectified linear units improve restricted Boltzmann machines". *Proceedings of the 27th International Conference on International*

Conference on Machine Learning (ICML-10), Haifa, Israel, June 21-24, pp. 807–814, (2010).
9. Hochreiter, S. “The vanishing gradient problem during learning recurrent neural nets and problem solutions”. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, **Vol. 6**, pp.107–116, (1998).
10. Hochreiter, S., Bengio, Y., Frasconi, P., and Schmidhuber, J. “Gradient flow in recurrent nets: the difficulty of learning long-term dependencies”, *A Field Guide to Dynamical Recurrent Neural Networks*, **Vol 1**, pp.237-243 (2001).
11. D. F. Specht, "A general regression neural network", *IEEE Transactions on Neural Networks*, **Vol 2**, pp.568-576 (1991).
12. D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, Massachusetts & United States of America (1989).