Large datasets: a mixed method to adapt and improve their learning by neural networks used in regression contexts

Marc Sauget[1], Julien Henriet[1], Michel Salomon[2], and
Sylvain Contassot-Vivier[3]
marc.sauget@univ-fcomte.fr

[1] Femto-ST, ENISYS/IRMA, F-25210 Montbéliard Cedex, France
[2] LIFC, EA 4269, University of Franche-Comté,
BP 527, F-90016 Belfort Cedex, France
[3] LORIA, UMR CNRS 7503, University Henri Poincaré Nancy-1, France

Abstract. The purpose of this work is to further study the relevance of accelerating the Monte-Carlo calculations for the gamma rays external radiotherapy through feed-forward neural networks. We have previously presented a parallel incremental algorithm that builds neural networks of reduced size, while providing high quality approximations of the dose deposit [4]. Our parallel algorithm consists in an optimized decomposition of the initial learning dataset (also called learning domain) in as much subsets as available processors. However, although that decomposition provides subsets of similar signal complexities, their sizes may be quite different, still implying potential differences in their learning times. This paper presents an efficient data extraction allowing a good and balanced training without any loss of signal information. As will be shown, the resulting irregular decomposition permits an important improvement in the learning time of the global network.

Keywords: Pre-clinical studies, Doses Distributions, Neural Networks, Learning algorithms, External radiotherapy, Data extraction.

1 Introduction

The work presented in this paper takes place in a multi-disciplinary project called *Neurad* [1], involving medical physicists and computer scientists whose goal is to enhance the treatment planning of cancerous tumors by external radiotherapy [12]. The final objective of our work is to propose a new tool to evaluate the result of an external radiotherapy treatment. In our previous works [2,4], we have proposed an original approach to solve scientific problems whose accurate modeling and/or analytical description are/is difficult. A short review of this problem is described in [10]. One of the major originality of that work is to use a Monte-Carlo simulator to build a very accurate dataset instead of measured data [6] (less accurate and sparser).

More precisely, the *Neurad* project proposes a new mechanism to simulate the dose deposit during a clinical irradiation. Our method consists in using a set of neural networks (one per processor), which act as universal approximators, in combination with a specific computational code. Each neural network predicts the dose delivered in a homogeneous environment on a subdomain of the domain to be treated, whereas a specific algorithm expresses the rules to manage any heterogeneous environment. The feasibility of this original project has been clearly established in [12]. It was shown that our approach results in an accuracy similar to the Monte-Carlo one and fulfills the time constraints of the external radiotherapy evaluation. The accuracy of a neural network is a crucial issue; therefore it is the subject of many research works on neural network learning algorithms.

The main idea of our work is to improve the performance of our incremental learning algorithm [3,4] by an efficient decomposition of the learning set. We have already proposed an optimized decomposition method based on an evaluation of the signal complexities of each data subset [17]. This approach provides subsets of similar signal complexities, however these subsets may have different sizes. So, although that decomposition offers a better homogeneity in the learning times of the subsets than classical decompositions, it can still be enhanced. A way to further improve the learning step is to work on the subsets sizes. More precisely, the work presented thereafter is motivated by the following question:

> *"If two data sets have the same complexity, why do they not have the same number of data samples?".*

2 Extraction of a representative set from massive data

Our study deals with the construction of a representative dataset from a larger database, in order to train a neural network. The data samples have the following characteristic: they are usually regularly spaced and often they contain statistical noise. These samples could not be used directly to obtain an efficient data learning. Indeed, the larger is the dataset, the longer is the learning time. When the dataset is too large, the neural network behaves more like a database with a low generalization (interpolation) capacity. In our first studies we have removed the statistical noise, but the limits of this approach were rapidly reached. Therefore, we are now looking for a method that limits the size of the dataset and captures the salient characteristics of the original dataset to be approximated.

There are many solutions in the literature to limit the size of a dataset. In particular, Jankowski and Grochowski present in [8,11] a state of the art of these methods in the context of neural networks used for classification. On the one hand they describe solutions that filter data to remove noise and on the other hand, they present ones doing data selection based on a good value repartition or using a mask. Another solution is to keep only the most homogeneous data to build the dataset [7,9]. However, those solutions are not fully adapted to neural networks used as universal approximators where the data must reflect the signal on the entire domain while preserving the sharp variations. In the classifier

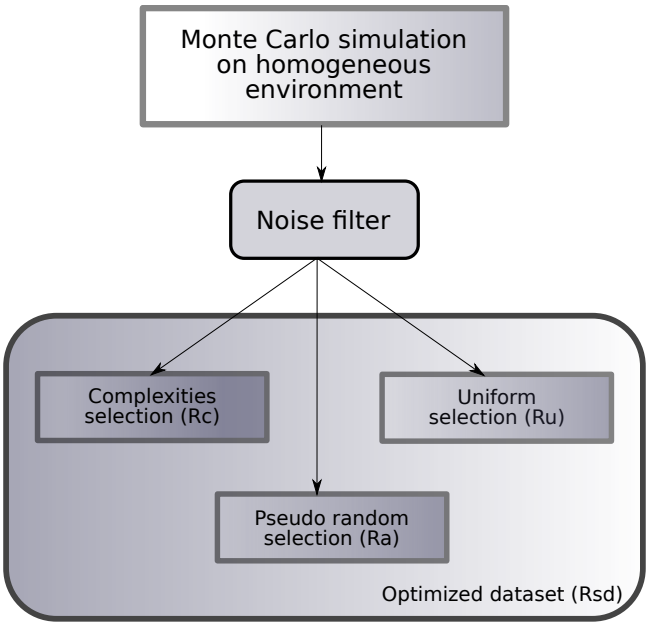

Fig. 1. A mixed selection method for optimized dataset decomposition

context, the most important data are at the frontiers between the different data classes and their accuracy must be preserved whereas other parts can be sparser and less accurate.

In this paper, we present an innovative method to select data samples based on the combination of two existing solutions together with a selection based on the complexities (see Fig. 1). As shown in the following, the proposed method is designed for neural networks used as approximators. The interest of this method is to improve the learning time thanks to a suitable selection of the data samples, leading to an optimized dataset decomposition. That approach prevents overfitting due to the presence of useless noising data and reduces the computational cost of the learning step [19].

2.1 Conservation of the global aspect of the signal

The general aspect of the signal could be easily guaranteed by the use of a uniform grid applied to the ranges of the dataset. The first tool used to build our dataset is very simple: the position of the new uniform grid is evaluated and we select the corresponding samples to build the subset R_u. The resolution of that grid could be quite low because its only interest is to constrain the learning on the global aspect of the signal. Its particularities will be taken into account by another group of data samples selected according to their complexities.

2.2 Conservation of the particularities of the signal

The most difficult point during the learning phase of a neural network used as a universal approximator is to keep its general aspect without forgetting the quality of interpolation in local areas having an important gradient value. The equilibrium between these two parts is always difficult to preserve. The solution presented here consists in only selecting the most important values taking into account their local complexity. Our notion of *data complexity* has already been presented in [17]. We recall here that, in order to evaluate the local complexity $lC_{i,j}$ at spatial point (i, j), we use the variations between that point and a given set of its neighbors (x, y), each of them being distance weighted. So, the local complexity at point (i, j) has the following form:

$$lC_{i,j} = \sum_{x=i-r}^{i+r} \sum_{y=j-r,(x,y)\neq(i,j)}^{j+r} \frac{|f(x,y) - f(i,j)|}{||\overrightarrow{(x,y)} - \overrightarrow{(i,j)}||} \tag{1}$$

where parameter r defines the size of the neighborhood taken into account.

The algorithm used to select n points is described in Algorithm 1. This algorithm evaluates the complexities of every point according to (1) and produces a decreasing order sorted list. Once the list is built, the algorithm returns a subset of the initial dataset composed of the n most characteristic points. The resulting subset corresponds to the R_c data subset.

Algorithm 1 Most particular points selection

Require: Dataset viewed as elements set $E = \{e_1, e_2, \dots\}$,
$|E| = m$, and $n < m$ is the number of points to select.

1: $R_c = \{\}$, $R_p = \{\}$
2: **for** i = 1 to m **do**
3: c = evaluateComplexity(e_i)
4: decreasingOrderInsertion(e_i, c, R_p)
5: **end for**
6: append($R_c, R_p\{1, n\}$)
7: **return** R_c

2.3 Homogeneous density of the complexity classes

The combination of the two previous methods does not ensure that the obtained dataset provides an optimal learning. In [15], the authors have shown for a neural network used as a classifier that a dataset is composed of a few sets of data samples with the same complexity. They called such a set a class and also showed that the different classes should have the same density (number of samples).

In our case, the neural network is used as an approximator, so the data samples values are just discretized to define the complexity classes induced by a dataset. Hence, we firstly compute the complexity classes considering the data samples that compose subsets R_u and R_c. Secondly, once the classes are identified, new data samples are selected from the initial dataset to complete the classes that are not sufficiently represented. The objective is to obtain classes with similar densities, in order to respect the quality constraint. The new data samples selected to complete R_u and R_c are chosen with a classical random method, they define the data subset R_a.

2.4 Global building of the dataset

To obtain each dataset affected to a single processor, more precisely a subset of the global dataset, the process is as follows.

Firstly, the initial global dataset is decomposed in p subsets, where p is the number of available processors, using our URB based decomposition described in [17]. Since the resulting decomposition is irregular, the subsets have different sizes. In order to have similar sizes, we impose that all the subsets (R_{sd}) have N data samples: the size of the smaller subset in the decomposition. Obviously, this last subset keeps all its data samples, while in the remaining $p-1$ subsets, some data samples must be discarded.

The second step, which reduces the size of the larger R_{sd} subsets, uses the three methods previously described to select for each of them the N data samples that will be retained. In fact, R_u, R_c, and R_a are computed for each subset R_{sd} such that these data subsets have each N data samples. Then, the reduced subset R_{sd} is built by picking elements from its corresponding sets R_u, R_c, and R_a, according to the respective proportions α, β and γ, such that:

$$\alpha = \frac{||R_{sd} \cap R_u||}{N}, \quad \beta = \frac{||R_{sd} \cap R_c||}{N}, \quad \gamma = \frac{||R_{sd} \cap R_a||}{N} \tag{2}$$

where $\alpha + \gamma + \beta = 1$. In a first approach, similar ratios might be used ($\alpha = \gamma = \beta = 1/3$), but this choice could not be retained because R_a is not a real data subset. Indeed, its main interest is to guarantee the equilibrium between the complexity classes. 20% of the data subset R_a (γ) are sufficient and necessary to balance the complexities. The most important data are provided by R_u, thus 50% of the data samples from R_{sd} are selected in it. Finally, the remaining samples correspond to the 30% (β) most characteristic ones in R_c.

Once the p data subsets of N samples have been built, they are learned using our fault-tolerant parallel algorithm [4].

3 Experimental results

In this section, both quality and performance of our approach are experimentally assessed. Our algorithm has been implemented in standard C++ with OpenMPI [14] for the parallel part. All the experiments have been performed using

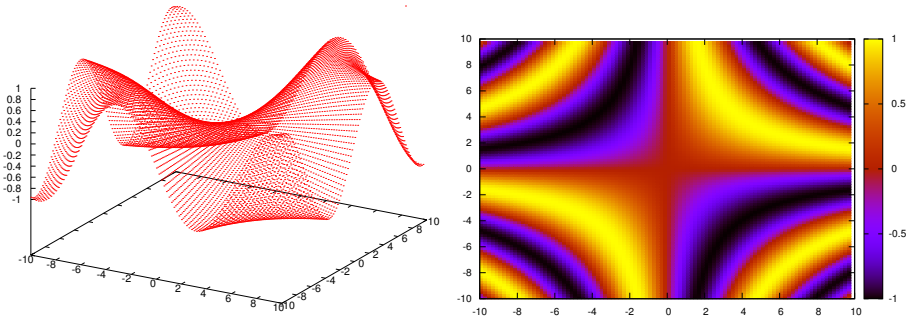

Fig. 2. Synthetic dataset generated from the function: $f(x, y) = sin(0.1 * x * y)$

the computing resources from the *Mesocenter of Franche-Comté* [13]: a cluster consisting of 8-core nodes (two Intel Xeon Nehalem quad-core processors) with 12GB RAM.

3.1 Generic function

The objective of our method is to provide an accurate learning using only a selection of samples from a large dataset. To perform a first evaluation, we have built a synthetic dataset using a generic 2D function given in Fig. 2. The dataset is composed of 10000 points uniformly distributed over the two dimensions.

We have evaluated the training times to obtain the desired accuracy of learning in two cases. In the former case, the domain decomposition method presented in [4] has been used with the entire dataset. In the latter case, a filtered version of the dataset has been used. The decomposition method allows us to obtain different subdomains with a relative similarity of complexities. We have also chosen to limit the size of each data subset using as reference the size of the smaller subset obtained with the decomposition mechanism. Several dataset decompositions have been tested and for each one we give the average size of the data subsets.

The results are presented in Table 1. They show the average learning time of the subdomains for each decomposition, as well as the standard deviation. This last information allows us to check the good balancing of the training times between the subdomains. Finally, the last column indicates the speedup between the average learning time obtained with the full dataset and the one obtained with the filtered one. We show with this first test the good performance of our data subset reduction method: an average speedup value of 3 can be observed. Furthermore, the speedup remains almost constant, whatever the decomposition degree used. This observation can be explained by the structure of the function, since it presents only a few parts with important gradient values which are also regularly distributed on the studied domain. Let us notice that the cumulative size of all the subdomains resulting from a decomposition is greater than the size of the initial global dataset (10000 samples in this case). Indeed, to ensure a homogeneous quality of the results of our parallel algorithm [4] throughout the global domain, each subdomain must have a small overlapping area with all its neighbors, thus some data samples belong to several subdomains.

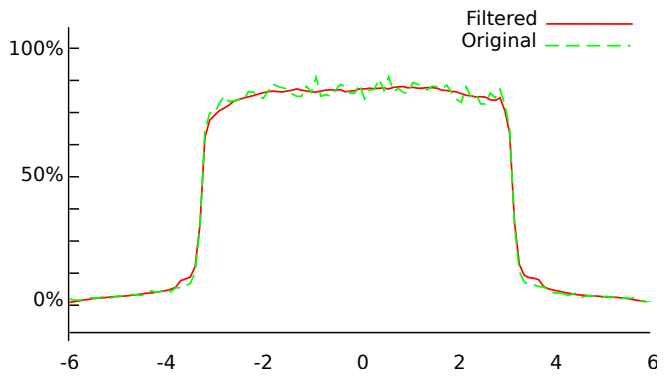

Fig. 3. The impact of the filtering step.

3.2 Radiotherapy context

In the radiotherapy context, we use data resulting from a Monte-Carlo simulation that represents an irradiation in a homogeneous environment. The first part is the evaluation of the initial dataset obtained with a Monte-Carlo simulator [5]. These data are very dense and quite noisy. To enhance the learning process, a first treatment is used to limit the negative impact of the noise, as shown in Fig. 3. We use a simple filter to reduce the noise. Figure 4 presents a comparative view of the original data and the final data obtained from our selection process. These curves correspond to the dose deposit for a single material. The next step is the learning process. The neural networks obtained after the learning phase are then used in conjunction with a specific algorithm to compute the result of an irradiation in any kind of environment, potentially heterogeneous [18].

To evaluate the behavior of our algorithm, we have chosen a restricted learning set. This configuration allows us to verify the interest of our algorithm without requiring a very long learning time. In fact, we have chosen to use only two materials (tissue and bones) and one configuration of irradiation. Since we use a grid of 120×100 points, the dataset is obviously composed of 120×100×2 samples. Each sample is characterized by seven components: the position in the three spatial dimensions; the material density; its distance from the irradiation source and the beam width (minimum and maximum values).

The tests have be performed with different decomposition degrees to evaluate the accuracy of the learning and to determine the limits of our solution. The

Subdomains	Original dataset			Selective dataset			Speedup
	Size	Average times (s)	SD	Size	Average times (s)	SD	
4	2962	393	92	1155	131	14	3.0
8	1591	173	61	519	59	5	2.9
16	846	92	37	221	29	3	3.1

Table 1. Performances of the learning phase for the generic function.

Subdomains	Original dataset			Selective dataset			Speedup
	Size	Average times (s)	SD	Size	Average times (s)	SD	
4	14520	1054	887	7258	297	84	3.5
8	4249	541	611	3035	113	17	4.7
16	3546	276	371	846	055	11	5.0

Table 2. Performances of the learning phase for the dose deposit curves.

objective of this work is to reduce the time of the learning step without affecting the final quality. So, the learning time is the essential parameter to take into account to validate our dataset construction. In all cases, the given times are only the learning times, they do not include any other data manipulation. The training stops when the specified accuracy is reached.

Table 2 shows the average learning time for each training case, as well as the standard deviation. As for the generic function in the previous subsection, this last information allows us to check the good balancing of the subdomains training times. Note that the load balancing is a very important parameter because it indicates the degree of use of all the computing resources during the training step. These results, together with the speedup information, show that the use of a selected dataset efficiently improves the learning time without inducing any significant accuracy loss (controlled during the learning process). So, we obtain in all cases a good speedup (values ranging from 3 to 5) with a very good improvement (reduction) of the standard deviation. We can also notice that with real data, like the radiotherapy case, the speedup is not as regular as in the previous case, with the generic function. We explain this difference by the fact that the parts of the dose deposit curves that exhibit an important gradient are less regularly distributed (as it was the case for the generic function). Thus, as one could expect, the size of the data subsets is not the only parameter having an impact on the learning time.

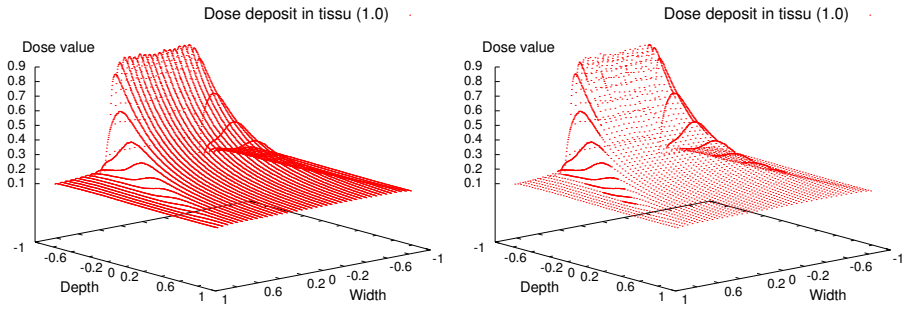

Fig. 4. General aspect of the dose deposit curves (left: original set, right: selected points).

4 Conclusion and future work

In this paper, we have presented a simple and efficient method to build an optimized learning dataset from data produced by a physical simulation. Thanks to this method, both data size and noise can be reduced without loosing any information. Hence, this approach significantly reduces the learning times without impacting the final quality.

Qualitative and quantitative evaluations of the algorithm have been performed experimentally on artificial and real datasets. The use of an artificial dataset points out that this method is not restricted to the specific radiotherapy context. The tests with real datasets confirm the good behavior of our algorithm, whether in terms of quality or performance, for complex datasets. The low standard deviation values show that the method gives well-balanced decompositions, and consequently smaller learning times.

In the following of the *Neurad* project, it will be necessary to test our different algorithms in real conditions, which means with complete datasets required in the medical operational context. A training using a full dataset, as well as a dose deposit calculation in real 3D images will allow us to fully validate our project.

Acknowledgements

The authors thank the LCC (Ligue Contre le Cancer) for the financial support, the Franche-Comté county and the APM (Agglomération du Pays de Montbéliard).

References

1. Bahi, J.M., Contassot-Vivier, S., Makovicka, L., Martin, E., Sauget, M.: Neurad. Agence pour la Protection des Programmes. No: IDDN.FR.001.130035.000.S.P.2006.000.10000 (2006)
2. Bahi, J.M., Contassot-Vivier, S., Makovicka, L., Martin, E., Sauget, M.: Neural network based algorithm for radiation dose evaluation in heterogeneous environments. In: Artificial Neural Networks - ICANN 2006. Lecture Notes in Computer Science, vol. 4132/2006, pp. 777–787. Springer Berlin / Heidelberg, Athens, Greece (Sep 2006)
3. Bahi, J.M., Contassot-Vivier, S., Sauget, M.: An incremental learning algorithm for functional approximation. Advances in Engineering Software 40(8), 725–730 (2009), doi:10.1016/j.advengsoft.2008.12.018
4. Bahi, J.M., Contassot-Vivier, S., Sauget, M., Vasseur, A.: A parallel incremental learning algorithm for neural networks with fault tolerance. In: Palma, J.M.L.M., Amestoy, P., Daydé, M.J., Mattoso, M., Lopes, J.C. (eds.) VECPAR. Lecture Notes in Computer Science, vol. 5336, pp. 174–187. Springer (2008)
5. BEAM-nrc: NRC of Canada. `http://www.irs.inms.nrc.ca/BEAM/beamhome.html`
6. Blake, S.W.: Artificial neural network modelling of megavoltage photon dose distributions. Physics in Medicine and Biology 49, 2515–2526 (2004)

7. Chu, C.K., Deng, W.S.: An interpolation method for adapting to sparse design in multivariate nonparametric regression. Journal of Statistical Planning and Inference 116(1), 91 – 111 (2003)
8. Grochowski, M., Jankowski, N.: Comparison of instance selection algorithms ii. results and comments. In: Rutkowski et al. [16], pp. 580–585
9. Guo, G., Zhang, J.S., Zhang, G.Y.: A method to sparsify the solution of support vector regression. Neural Comput. Appl. 19(1), 115–122 (2010)
10. Haas, O., Goodband, J.: Intelligent and Adaptive Systems in Medicine, chap. Artificial Neural Networks in Radiation Therapy, pp. 213–258. Series in Medical Physics and Biomedical Engineering, Taylor & Francis (2008)
11. Jankowski, N., Grochowski, M.: Comparison of instances seletion algorithms i. algorithms survey. In: Rutkowski et al. [16], pp. 598–603
12. Makovicka, L., Vasseur, A., Sauget, M., Martin, E., Gschwind, R., Henriet, J., Salomon, M.: Avenir des nouveaux concepts des calculs dosimétriques basés sur les méthodes de Monte Carlo. Radioprotection 44(1), 77–88 (jan 2009), http://dx.doi.org/10.1051/radiopro/2008055
13. Computing Mesocenter of Franche-Comté. `http://meso.univ-fcomte.fr`
14. Open Source High Performance Computing. `http://www.open-mpi.org`
15. Pan, F., Wang, W.: Finding representative set from massive data. Tech. rep., IEEE International Conference on Data Mining (2005)
16. Rutkowski, L., Siekmann, J.H., Tadeusiewicz, R., Zadeh, L.A. (eds.): Artificial Intelligence and Soft Computing - ICAISC 2004, 7th International Conference, Zakopane, Poland, June 7-11, 2004, Proceedings, Lecture Notes in Computer Science, vol. 3070. Springer (2004)
17. Sauget, M., Laurent, R., Henriet, J., Salomon, M., Gschwind, R., Contassot-Vivier, S., Makovicka, L., Soussen, C.: Efficient domain decomposition for a neural network learning algorithm, used for the dose evaluation in external radiotherapy. In: Diamantaras, K.I., Duch, W., Iliadis, L.S. (eds.) ICANN (1). Lecture Notes in Computer Science, vol. 6352, pp. 261–266. Springer (2010)
18. Vasseur, A., Makovicka, L., Martin, E., Sauget, M., Contassot-Vivier, S., Bahi, J.M.: Dose calculations using artificial neural networks: a feasibility study for photon beams. Nucl. Instr. and Meth. in Phys. Res. B 266(7), 1085–1093 (2008)
19. Wu, A., Hsieh, W.W., Tang, B.: Neural network forecasts of the tropical pacific sea surface temperatures. Neural Networks 19(2), 145 – 154 (2006), earth Sciences and Environmental Applications of Computational Intelligence