Parallel algorithm based on Singular Value Decomposition for high performance training of Neural Networks

Gabriele Maria Lozito¹, Valentina Lucaferri¹, Mauro Parodi², Martina Radicioni¹, Francesco Riganti Fulginei¹ and Alessandro Salvini¹

 Roma Tre University, Rome, Italy valentina.lucaferri@uniroma3.it
 ² University of Genoa, Genoa, Italy

Abstract. Neural Networks (NNs) are frequently applied to Multi Input Multi Output (MIMO) problems, where the amount of data to manage is extremely high and, hence, the computational time required for the training process is too large. Therefore, MIMO problems are often split into Multi Input Single Output (MISO) problems; MISOs are further decomposed into several Single Input Single Output (SISO) problems. The aim of this paper is to present an optimized approach for NNs training based on properties of Singular Value Decomposition (SVD), allowing to decompose the MISO NN into a collection of SISO NNs. The decomposition provides a two-fold advantage: firstly, each SISO NN can be trained by using a one-dimensional function, namely a limited dataset, and then a parallel architecture can be implemented on a PC-cluster, decreasing the computational cost. The parallel algorithm performance are validated by using magnetic hysteresis dataset with the aim to prove the computational speed up by preserving the accuracy.

Keywords: Neural Networks \cdot Parallel Computing \cdot Magnetic Hysteresis

1 Introduction

Neural Networks (NNs) for Multi-Input- Multi-Output (MIMO) problems have gained wide attention in many scientific fields [6, 3], especially regarding complex non-linear systems without a closed-form solution. In these applications, a NN, able to learn non-linear relationship between quantities by means of a dataset, represents a reasonable alternative to the use of mathematical models [19, 17]. The main drawback of any neural model is, however, the necessity of a large amount of data/measurements for its set up; in fact, a NN with m inputs and n outputs can be directly implemented to approximate a non-linear function, provided the amount of data to manage is not too large; moreover the high computational cost, needed for the wide size of training patterns and for complex NN architecture, makes the direct approach inefficient.

Therefore, several authors make the problem easier by dividing MIMO NN into a collection of n Multi-Input-Single-Output (MISO) NNs [8,9], further, split

1

into several Single-Input-Single-Output (SISO) NNs; in this way, their structure appears to be considerably simplified and the time required by the whole learning process results to be lower. Singular Value Decomposition (SVD) turns out to be a powerful instrument for switching between MISO NNs and a group of SISO NNs, since it is defined for any array of data and it provides a low-rank matrix, which is a good approximation to the original one [1, 7]; the efficiency of this approach has already demonstrated in different works [16, 12]. The presented paper concerns the implementation of parallel algorithm on a PC-cluster that is able to address the SISO NN training process to a cluster node, separately, decreasing strongly the computational cost. Moreover, the parallel architecture obtained from SVD-based approximation is applied to magnetic hysteresis data and the study of its ability to model the magnetic hysteresis constitutes the main issue of this work. Indeed, although the magnetic hysteresis problem has been already solved by the authors through a suitably trained NN [11, 2, 10, 13], it is still a difficult task [5, 4, 14], especially when dealing with materials having large variance in characteristics and NN must be trained with a dataset describing these conditions. Obtaining such data involves a whole set of measurements that can be very difficult to perform; thus, to solve this problem in a feasible way, the parallel SVD-based algorithm is applied to a 2D-array, which represents the magnetic permeability, μ , obtained starting from experimental values of magnetic field, H, and flux density, B [15].

This paper is organized as follows: Section 2 summarizes some of the main properties of the SVD technique, emphasizing the decomposition of bidimensional functions. Section 3 focuses on the parallel algorithm implemented for training each SISO NN: firstly, MISO-SISO decomposition is introduced, pointing out the training of each SISO NN, then, PC-cluster characteristics are introduced. In Section 4 the performances of parallel algorithm are evaluated by using magnetic hysteresis data and, finally, in Section 5, the conclusions follow.

2 The approximation method: SVD

Among the methods used to simplify a MISO problem into a collection of SISO, the SVD represents a valid tool, that allows to approximate a multivariate functions without losing accuracy.

The SVD of a rectangular matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ is the factorization of \mathbf{A} into the product of three matrices: $\mathbf{A} = \mathbf{U} \mathbf{\Sigma} \mathbf{V}^{\mathbf{T}}$ where $\mathbf{U} = (u_1 \dots u_m) \in \mathbb{R}^{m \times m}$, $\mathbf{V} = (v_1 \dots v_n) \in \mathbb{R}^{n \times n}$ and $\mathbf{\Sigma} \in \mathbb{R}^{m \times n}$ is a diagonal one having its non-zero diagonal entries equal to the singular values $\sigma_{k,k}$ with $k=1,\dots,p$ and $p=\min(m,n)$, written in descending order $\sigma_{1,1} > \sigma_{2,2} > \dots > \sigma_{p,p} > 0$. Thus:

$$\mathbf{A} = \sum_{k=1}^{p} \sigma_{k,k} \mathbf{u}^{\langle k \rangle} (\mathbf{v}^{T})^{\langle k \rangle}$$
(1)

where $\mathbf{u}^{\langle k \rangle} \subset \mathbb{R}^m$ and $\mathbf{v}^{\langle k \rangle} \subset \mathbb{R}^n$ are orthonormal vectors coincident with the column vectors of matrices \mathbf{U} and \mathbf{V}^T , respectively. Usually, the smallest

singular values are neglected and an approximated version of (??) and (1) is considered:

$$\hat{\mathbf{A}} = \sum_{k=1}^{\hat{p}} \sigma_{k,k} \mathbf{u}^{\langle k \rangle} (\mathbf{v}^T)^{\langle k \rangle}$$
(2)

with $\sigma_{1,1} > \sigma_{2,2} > \ldots > \sigma_{\hat{p},\hat{p}} > 0$ and $\hat{p} < \min(m,n)$, whereas $\sigma_{j,j}$ is forced to be 0, for $\hat{p} < j < p$.

Let us assume that the **A** entries are coming from the sampling of a bivariate function $f(x_1, x_2)$, so $A_{i,j} = f(x_{1,i}, x_{2,j})$ with $i = 1 \dots m$ and $j = 1 \dots n$. Correspondingly, the columns of the matrices **U** and V^T are arising from the sampling of unknown univariate functions, $u_{k,i} = \psi_k(x_{1,i})$ and $v_{k,j}^T = \eta_k(x_{2,j})$. Once obtained \hat{p} with a fixed accuracy ε , the approximation can be written as follows:

$$\hat{A}_{i,j} = \hat{f}(x_{1,i}, x_{2,j}) = \sum_{s=1}^{\hat{p}} \sigma_s \hat{\psi}_s(x_{1,i}) \hat{\eta}_s(x_{2,j})$$
(3)

The different unknown univariate functions in (3) can be evaluated by appropriate curve fitting techniques, such as NNs.

3 Parallel algorithm for SISO NNs training

3.1 MISO NN decomposition

The SVD-based approximation of bivariate functions described in the previous section can be exploited for optimizing the feed-forward MISO NNs learning process. Thus, the idea is to split the MISO NN into several SISO NN by applying the SVD-based approximation of multivariate functions, to reduce the time of learning process and to simplify the NN architecture.

Let us consider an array of data **A**, obtained by sampling a 2D-function $f(x_1, x_2)$, and define a decomposition operator, D, which indicates the SVD of function $f(x_1, x_2)$, $D\{f(x_1, x_2)\} = \sum_{i=1}^{\hat{p}} \sigma_i \psi_i(x_1) \eta_i(x_2)$.

When the operator is applied, it generates $2 \times \hat{p}$ univariate functions and each of them can be approximated by a SISO NN, as shown in Figure 3.

Every SISO NN presents a simple feed-forward architecture, composed by an input neuron, one hidden layer and an output neuron. The main advantage is that each NN is independent from the other, so its training can be implemented as a stand-alone process being part of a larger parallel architecture.

3.2 Parallel computing in a PC-cluster system

The principal aim of using a PC-cluster is to improve the performance for large computational task, such as MISO NN training, that can be divided into smaller tasks distributed around the nodes. The SISO NN training is implemented on



Fig. 1. Scheme of Parallel algorithm implemented for SISO NNs training.

a high performance computing, HPC, cluster, that consists of 4 compute nodes, each one composed of 24 CPUs. A Single Program Multiple Data, SPMD, parallel model is performed; thus, the master node sends the program, involving training set and algorithm, to the others computational nodes and it commands every one to execute the learning process of a SISO NN. The communication is realized through the Message Passing Interface standard, MPI, which allows the programmer to easily pass the message from one of computers to the others by calling the function offered from the libraries. Generally, a parallel process requires a communicator, containing a group of processes, whose number defines its size, r; each process has a rank inside the communicator, namely a number that permits to recognize it. So, the implementation of parallel NNs training involves a communicator having $n \times \hat{p}$ size, where n indicates the dimension of the initial array of data; especially, $r = 2 \times \hat{p}$ for 2-D array, n=2. As shown in Figure 2, each process returns a SISO NN, able to approximate the one-dimensional function, representing the U matrix columns or $\mathbf{V}^{\mathbf{T}}$ matrix rows. Once, the U and \mathbf{V} matrices have been reconstructed, the approximation of original array can be obtained, by using equation 2.

4 Validation of SVD-based parallel algorithm: magnetic hysteresis data

With the aim to validate the presented approach, the parallel algorithm is used to train a neural network implemented for modelling magnetic hysteresis. A collection of asymmetric loops has been generated for building the data set for NN training by employing Preisach model in Matlab environment [18]. Each loop has been sampled in n points and for every couple of coordinates $[H_k, B_k]$ the corresponding differential magnetic permeability, μ_k has been computed. As a result, an array of data $\mu(B_n, H_n)$ function of H and B variables is obtained to test the SVD-based algorithm.

Thus, assuming $x_1 = B_1, ..., B_m, x_2 = H_1, ..., H_n$ and $f(x_1, x_2) = \mu(H, B)$, the SVD applied to bivariate function returns



Fig. 2. Parallelization of training process. From left to right: trasformation of MISO NN into several SISO NNs. Each NN training represents a process of rank k, with $k = 1...\hat{p}$, executed by a cluster node.

$$\mu_{i,j} = \mu(B_i, H_j) = \sum_{s=1}^p \sigma_s \psi_s(B_i) \eta_s(H_j)$$
(4)

To reduce the total number of NNs required for reconstructing the original function, a threshold value is fixed to $\epsilon = 0.001$ in order to obtain a reduced SVD form with $\hat{p} = 3$.

$$\hat{\mu}_{i,j} = \sum_{s=1}^{3} \sigma_s \hat{\psi}_s(B_i) \hat{\eta}_s(H_j)$$
(5)

After decomposition, the total number of SISO NNs results to be 6: 3 univariate functions of B, $\psi_1(B), \psi_2(B), \psi_3(B)$; 3 univariate functions of H, $\eta_1(H), \eta_2(H), \eta_3(H)$. Each SISO NN has a feed-forward architecture and it was trained by using Levenberg Marquardt algorithm. The NN architecture consists of a input neuron, representing the B or H variable sampled in n points, the hidden layer is composed by 8 neurons and the output neuron returns the one dimensional function $\psi(B)$ or $\eta(H)$, respectively. The training set for each NN is made by 70 of 100 sampled points belonging to the univariate function, which is intended to approximate. So, for computing the parallel learning process, the master node has to send a program, including the training set and Levenberg Marquardt algorithm, to the others one. Hence, a communicator, involving 6 processes distributed between the 4 available compute nodes, is set up as shown in Figure 1, with the purpose to minimize the computational effort of a single node. The performance of NN is compared with the points excluding the ones in the training, especially, test set is composed by 30 samples. The NN shows a good accuracy in modelling, figure 3 shows that each NN is able to reproduce the univariate function trend, with a $MSE \simeq 10^{-8}$. Finally, the NN outputs are exploited to reconstruct the μ array, following the equation 5. Table 1 shows a comparison between the time required by whole learning process with and without parallel architecture implementation; a strong reduction of computational time is achieved.



Fig. 3. Results on validation test of NNs: left $\psi_s(B)$; right $\eta_s(H)$.

System	Training time (s)
Serial implementation	30.81
Parallel implementation	7.24

Table 1. Computational time results achieved for the bivariate function, $\mu(B, H)$.

5 Conclusions

An optimization algorithm able to perform parallel learning process of SISO NNs, by exploiting the SVD properties has been presented. A well-known challenging topic is constituted by the modelling of magnetic hysteresis, a typical MISO problem. Thus, a procedure for magnetic hysteresis loops identification has been proposed. The method is based on the parallelization of NNs training processes, that allows to accurately model the hysteresis problem achieving a reduction of computational cost. The implemented SISO-based approach allows to provide a solution, even when the conventional use of a MISO strategy fails. Results show that the proposed method is a suitable tool for modelling the magnetic hysteresis, it is capable of reducing the processing time strongly and, at the same time, preserving the accuracy of solution. The presented technique constitutes an effective solution based on which the more complex problem of 3D magnetic hysteresis can be solved, hence, should be considered a valid starting point for future developments. In particular, it is worth noticing that the method can, in general, suit any multidimensional problem.

References

- Bizzarri, F., Parodi, M., Storace, M.: Svd-based approximations of bivariate functions. In: IEEE INTERNATIONAL SYMPOSIUM ON CIRCUITS AND SYS-TEMS. vol. 5, p. 4915. IEEE; 1999 (2005)
- Cardelli, E., Faba, A., Laudani, A., Riganti Fulginei, F., Salvini, A.: A neural approach for the numerical modeling of two-dimensional magnetic hysteresis. Journal of Applied Physics 117(17), 17D129 (2015)
- Chen, M., Ge, S.S., How, B.V.E.: Robust adaptive neural network control for a class of uncertain mimo nonlinear systems with input nonlinearities. IEEE Transactions on Neural Networks 21(5), 796–812 (2010)
- Duan, N., Xu, W., Wang, S., Zhu, J., Guo, Y.: Hysteresis modeling of hightemperature superconductor using simplified preisach model. IEEE Transactions on Magnetics 51(3), 1–4 (2015)

- Handgruber, P., Stermecki, A., Biro, O., Goričan, V., Dlala, E., Ofner, G.: Anisotropic generalization of vector preisach hysteresis models for nonoriented steels. IEEE transactions on magnetics 51(3), 1–4 (2015)
- Hovakimyan, N., Calise, A.J., Kim*, N.: Adaptive output feedback control of a class of multi-input multi-output systems using neural networks. International Journal of Control 77(15), 1318–1329 (2004)
- Huynh, H.T., Won, Y.: Training single hidden layer feedforward neural networks by singular value decomposition. In: Computer Sciences and Convergence Information Technology, 2009. ICCIT'09. Fourth International Conference on. pp. 1300–1304. IEEE (2009)
- Jianye, L., Yongchun, L., Jianpeng, B., Xiaoyun, S., Aihua, L.: Flaw identification based on layered multi-subnet neural networks. In: 2009 Second International Conference on Intelligent Networks and Intelligent Systems. pp. 118–121. IEEE (2009)
- Kabir, H., Wang, Y., Yu, M., Zhang, Q.J.: High-dimensional neural-network technique and applications to microwave filter modeling. IEEE Transactions on Microwave Theory and Techniques 58(1), 145–156 (2010)
- Laudani, A., Lozito, G.M., Riganti Fulginei, F.: Dynamic hysteresis modelling of magnetic materials by using a neural network approach. In: AEIT Annual Conference-From Research to Industry: The Need for a More Effective Technology Transfer (AEIT), 2014. pp. 1–6. IEEE (2014)
- Laudani, A., Lozito, G.M., Riganti Fulginei, F., Salvini, A.: Modeling dynamic hysteresis through fully connected cascade neural networks. In: Research and Technologies for Society and Industry Leveraging a better tomorrow (RTSI), 2016 IEEE 2nd International Forum on. pp. 1–5. IEEE (2016)
- Laudani, A., Salvini, A., Parodi, M., Riganti Fulginei, F.: Automatic and parallel optimized learning for neural networks performing mimo applications. Advances in Electrical and Computer Engineering 13(1), 3–12 (2013)
- Makaveev, D., Dupré, L., De Wulf, M., Melkebeek, J.: Modeling of quasistatic magnetic hysteresis with feed-forward neural networks. Journal of Applied Physics 89(11), 6737–6739 (2001)
- Rasilo, P., Singh, D., Aydin, U., Martin, F., Kouhia, R., Belahcen, A., Arkkio, A.: Modeling of hysteresis losses in ferromagnetic laminations under mechanical stress. IEEE Transactions on Magnetics 52(3), 1–4 (2016)
- Riganti Fulginei, F., Salvini, A.: Neural network approach for modelling hysteretic magnetic materials under distorted excitations. IEEE Transactions on Magnetics 48(2), 307–310 (2012)
- Riganti Fulginei, F., Salvini, A., Parodi, M.: Learning optimization of neural networks used for mimo applications based on multivariate functions decomposition. Inverse Problems in Science and Engineering 20(1), 29–39 (2012)
- Salvini, A., Riganti Fulginei, F.: Genetic algorithms and neural networks generalizing the jiles-atherton model of static hysteresis for dynamic loops. IEEE Transactions on Magnetics 38(2), 873–876 (2002)
- Salvini, A., Riganti Fulginei, F., Pucacco, G.: Generalization of the static preisach model for dynamic hysteresis by a genetic approach. IEEE transactions on Magnetics 39(3), 1353–1356 (2003)
- Turk, C., Aradag, S., Kakac, S.: Experimental analysis of a mixed-plate gasketed plate heat exchanger and artificial neural net estimations of the performance as an alternative to classical correlations. International Journal of Thermal Sciences 109, 263–269 (2016)