Proceedings of Engineering & Technology (PET)

pp. 902 - 907

State of the art of Support Vector Machine implementaion

Intissar Sayehi
Laboratory of Electronics and Microelectronics ,
(E. µ. E. L), FSM, Monastir, Tunisia
Intissar_sayahi@live.fr

Belgacem Bouallegue, Mohsen Machhout, Rached Tourki
Laboratory of Electronics and Microelectronics ,
(E. µ. E. L), FSM, Monastir, Tunisia

***Abstract*—the support vector machine (SVM) is a kernel method derived from statistical learning theory. It is a powerful tool for classification and regression. But the quadratic programming phase requires an important computation time comparing with other methods like regularization networks. Thus, a community of researchers proposed to exploit the programmable platforms to accelerate and ameliorate the SVM performances. In this work we present a consistent resume of the different implementations and the methodologies used. Then we compare and discuss the results and we proposed a model based design for SVM regression through Xilinx system generator. We identify its advantage and inconvenient.**

Keywords—support vector machine, classification, regression, microcontroller, Digital Signal Processing ,Field Programmable Gate Array ,system generator.

I. INTRODUCTION

Machine learning is the ability for a machine to learn from past examples and continually adapt to new situations. There are many easy and popular algorithms such K-NN, Neural Networks, K-means…whereas in term of performances and aptitude to generalization the kernel method excels such as the Support Vector Machine (SVM). It is a powerful and multidisciplinary tool for classification and regression problems. Solving the SVM training problem by using quadratic programming (QP) techniques is a severe and computationally expensive task. That SVM training time is critically reliant on the training dataset size and the problem's dimension. There were many algorithms like Sequential Minimum Optimization (SMO) and SVMLIGHT [1] have been proposed to resolve these problems analytically but don't give an enormous amelioration. That's motivates researchers to implement this method on programmable device to accelerate the computation time especially in case of online training. The embedded digital systems like microcontroller, Digital Signal Processors (DSPs) or Field Programmable Gate Arrays (FPGAs) allow attaining better resource-performance relation, but necessitating a careful implementation design.

In this paper we begin by presenting the SVM for classification and regression in section 2. Then, in section 3 we highlight different implementations of SVM and we compare between them. In section 4 we present the commonly used methods for FPGA design and we present our proposed design via the Xilinx System generator and we enumerate its advantages and limitations. In section 5 we conclude by recapitulating the paper contributions.

II. SUPPORT VECTOR MACHINE FOR CLASSIFICATION AND REGRESSION

A. Support Vector Machine for classification

SVM was first developed for classification problems then extended to regression problems. It is a supervised learning model based on the Vapnik and Chervonenkis learning theory. From a given training data set containing n samples $D=\{(x_1,y_1),(x_2,y_2),\ldots,(x_n,y_n)\}$. With $x \in \mathbb{R}^l$ and $y_i = \pm 1$.

The training phase consists of resolving the following quadratic programming problem with linear restrictions [2].

$$\min_{\alpha \in \mathbb{R}^n} \text{imise} \frac{1}{2}\sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j K(x_i, x_j) - \sum_{i=1}^{n} \alpha_i \qquad (1)$$

$$\text{Subject to } 0 \le \alpha_i \le C \text{ for i=1,…,n}$$

$$\sum_{i=1}^{n} \alpha_i y_i = 0$$

Where $K(x_i,x_j)$ is a kernel function. C is a regularization term used as a constraint for the value of the Lagrange multipliers α_i.

The feed-forward classification function of a new vector x is:

$$y(x) = \text{sgn}\left(\sum_{i=1}^{Nsv} y_i \alpha_i K(x_i, x) + b\right) \qquad (2)$$

αi and b are parameters calculated in the learning phase. Those training samples whose related αi are not equal to 0 are called the support vectors (SV). The learning step contest the model aptitude to the data complexity offering high generalization capability and guaranteeing great performances on non classed patterns.

Proceedings of Engineering & Technology (PET)
Copyright IPCO-2016
pp. 902 - 907

B. Support Vector Machine for regression

In this case the SVM task is to find a model for the data set $D=\{(x_1,y_1),(x_2,y_2),\ldots,(x_n,y_n)\}$ that matches the input x_i to the real output y_i (with $x_i \in \mathbb{R}^l$ *and* $y_i \in \mathbb{R}$).

By resolving the following quadratic programming problem with linear restrictions and an ε-insensitive loss function:

$$\min_{\alpha,\alpha^* \in \mathbb{R}^l} \text{imise} \frac{1}{2}\sum_{i,j=1}^{n}(\alpha_i^* - \alpha_i)(\alpha_j^* - \alpha_j)K(x_i, x_j) + \varepsilon\sum_{i=1}^{n}(\alpha_i^* + \alpha_i) - \sum_{i=1}^{n}(\alpha_i^* - \alpha_i)y_i \quad (3)$$

$$\text{St}: \ 0 \le \alpha_i^*, \alpha_i \le C \ \text{ for i=1,…,n } \sum_{i=1}^{n}(\alpha_i^* - \alpha_i) = 0$$

where $K(x_i,x_j)$ is a kernel function, C is the regularization term and ε is a positive constant presenting the called insensitive region in the interior of which the training errors are unseen. C and ε are predefined constants. The feed-forward evaluation function of a new, unlearned vector x is:

$$y(x) = \sum_{i=1}^{Nsv}(\alpha_i^*\alpha_i)K(x_i, x) + b \quad (4)$$

where parameters α_i, α^*_i and b are calculated in the learning phase. As in the classification model, only those resulting support vectors are used in the feed-forward phase.

III. PLATFORMS FOR SVM ALGORITHM IMPLEMENTATION

A. A microcontroller

It is a small and inexpensive computer fabricated to accomplish definite tasks. It has a variety of applications: industry, telecommunication, automotive… On a single integrated circuit, exists a processor core, memory, and programmable input/output peripherals. The SVM can be implemented in this device like in reference [3], for classifying sensor data. In this work, Thomas Nowak had done "Implementation and Evaluation of a Support Vector Machine on an 8-bit Microcontroller". The software package μSVM, was implemented on an Atmel AVR ATmega16 microcontroller. It was shown that despite the limited processing power and severe memory space limits in small microcontroller units, it is possible to execute an entirely SVM application there. The difficulty is the long execution time of the implementation when we have a large number of training examples.

In reference [4], the authors were proposed a new svm: RSVM (reduced svm) algorithm for embedded systems which reduces computational difficulty and memory use. In this work, the target device was a simple 8-bit microcontroller (AVR present inside all FPSLIC family devices from Atmel) which is designed for low-power applications. The goal was the design of a node of a wireless video-sensor network performing people detection. The results validate the successful design of simple smart system realizing classification tasks.

B. Graphics Processing Units (GPU)

GPUs architectures are devoted for severe and rigorous parallel computation. However in reference [5], GPUs are essentially used to accelerate special parts of the algorithm to gain computation time. Results for this implementation are very great, that it attains 120-150x speedups over LibSVM software on the datasets used.

As the large training data increases the computation time and becomes impractical, this work [6] proposed a GPU version for SVM classification for the detection of high level features in video shots. The modification was the calculation of the kernel matrix elements by the GPU. Comparing by the LIBSVM version, the results proves a significant acceleration in the training phase. It provides a speed development that abruptly increases with the size of the input data, aiding to solve big problems.

The potent parallel processing capability of GPU has competently improved the network anomaly detection technology based on support vector machine (SVM) for large-scale intrusion scenarios [7].

Thanks to the Graphics Processing Units (GPU) implementations, the performances of support vector machines algorithms have been improved. However, caution has to be taken because GPU is a fixed hardware and the interconnection cannot outfit the calculation or data flow of some applications.

C. A digital signal processor (DSP)

It is a specific microprocessor with optimized architecture for the operational requirements of digital signal processing. Its goal is usually to measure, filter and trait continuous real analog signals . also it can execute digital signal processing algorithms successfully. In this context, the authors of this work [8] exploit the characteristics of a DSP for speech recognition by support vector machine classifier.

The SVM was employed in recognition tasks. First, it learns the model parameters by a group of samples with known predicted values for the process. Therefore, the model will be iteratively optimized to obtain the best performance parameters. Then the model will be tested for validation. The speech recognition using support vector machine is used for feature matching. It proves the efficiency and speed up of SVM in speech identification with minimum error rates. Its accuracy is more than 80% for speech recognition.

D. Field Programmable Gate Array (FPGA)

It is digital integrated circuit that contains programmable blocks with configurable interconnections. It allows a great flexibility and efficiency for different applications. The most important advantage of this tool is that it can execute codes in parallel while Digital Signal Processors (DSPs) execute codes in series. Therefore, FPGAs cannot store as much data as DSPs . Other advantage of using FPGAs is their capacity to work

Proceedings of Engineering & Technology (PET)
Copyright IPCO-2016
pp. 902 - 907

with any word length the designer chooses. Whereas, DSP processors must be selected to hold the longest word length that exists in the code.

This reference [9] presents a entirely scalable heterogeneous FPGA architecture for the acceleration of the SVM classification. By exploiting of the parallel computational power offered by the FPGA, the proposed architecture reach to speed-up the CPU classification execution time by 2-3 orders of magnitude, while outperforming over 7× other works on FPGAs and GPUs.

Whereas, this work [10] present a successful design for a high performance and low resource consuming hardware for support vector classification and regression. The proposed architecture has been conceived as general use for embedded applications, where the number of support vectors and the resolution of the parameters can be configured. In addition, there is not a limit to the dimension of the input vectors and the number of support vectors. It is only limited by the size of the FPGA. The performance of this design was tested for a multi classification problem on a simplified COIL database and for regression problem on sinus cardinal function. In both cases, the average error rate for the hardware is between 0% and 0.02 %, which means that the SVM performs better when using the hardware then MATLAB.

IV. METHODS OF FPGA DESIGN

The success key of a FPGA design is the adequate tool used which influences the computation time and the cost.

A. Hand coding with hardware description language HDL

By using hardware description languages, such as VHDL and Verilog HDL, FPGA can be simply designed in hierarchical modules and synthesized into the FPGA. But an our case, the SVM algorithm contains two main phases that are hard to program due to the feed forward and optimization task. So this method cannot lead to design solution.

B. MATLAB HDL coder application

It is an integrated application in MATLAB for generating, Optimising and Verifying HDL Code. By giving the program file or simulink model, this application generate automatically Design for the hardware chosen. This figure resumes the C and HDL generation:

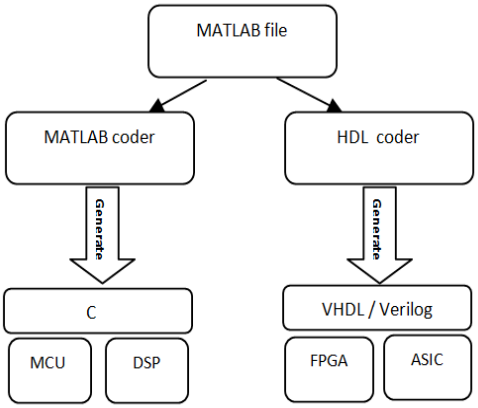

Fig. 1. C and HDL code generation

Comparing with the handwritten code this method is faster and generate VHDL directly and prototype an FPGA. Whereas it demands simple designs or programs not like the SVM algorithm which have to convert also the toolbox containing many functions files. Fortunately there is a solution which will be explained in next paragraph.

C. Xilinx System Generator (XSG)

XSG is a great graphical programming tool for hardware design working with MATLAB Simulink and cibling FPGA platforms. Therefore, the system generator enables Simulink model based design to generate the synthesizable HDL code. which dramatically reduced the design time and quickly evaluate new algorithms in hardware comparing with traditional HDL hand coding.

System Generator provides many proposes to well profitate from FPGA resources, Hardware Co-Simulation and accelerated simulation through hardware in the loop co-simulation [11]. Figure 2 presents the design flow of XSG :the

Proceedings of Engineering & Technology (PET)
Copyright IPCO-2016
pp. 902 - 907

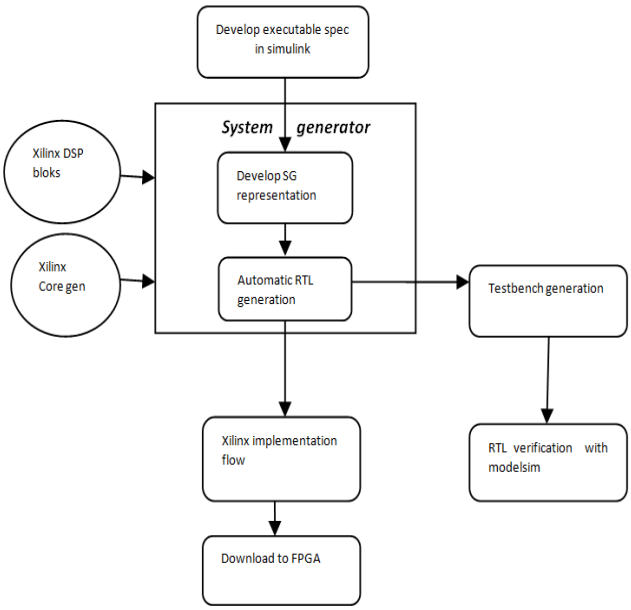

Fig. 2. The system generator design flow

D. Our proposed design for SVM algorithm

We propose to find a model for this non linear system by SVM regression:

$$y(i) = (0.8 - 0.5\exp(-(y(i-1))^2))y(i-1) - (0.3 + 0.9\exp(-(y(i-1))^2))y(i-2) + 0.1\sin(\pi.y(i-1)) + e(i)(26)$$

This system is attenuated by a Gaussian noise e(i) having zero mean and variance equal to 0.4.

To obtain the RKHS model, we employ 82 observations for the training phase and 150 new observations for the validation phase. The kernel used is of type: polynomial. The optimal parameters λ and σ of the machine learning are obtained by a cross validation technique.

We divide the SVM algorithm in two parts: training and testing. Each phase putted in subsystem calling the function files in svm toolbox thanks to M.code block in XSG it is possible to insert a function file in model design. We profit also from the various library blocks: Math Operation blocks, Logic and Bit Operation blocks, Signal Routing blocks. This is the generic model of SVM regression model:

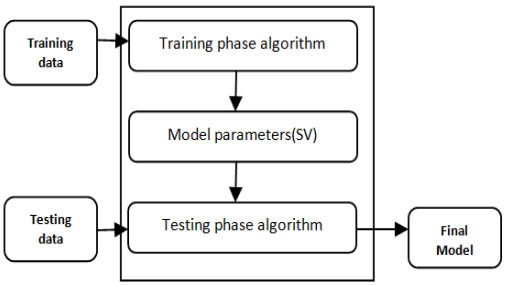

Fig. 3. The generic model of SVM

The model based design provides a mathematical and interactive method for solving problems related on designing complex control, signal processing and communication systems.

The Limitations of System Generator that not all blocks from Simulink are accessible in Xilinx library. And we cannot to generate hardware from models using the basic Simulink Blocks.

V. CONCLUSION

The present work resumes the main embedded handheld devices for SVM designing. The difference in the implementation platforms depends on the application domain.

The FPGA is the better platform due its technical characteristics and also due its efficient tools for modeling, simulation and synthesis making it a greatly useful platform. It has been shown that Xilinx System Generator is a system level modeling tool that facilitates FPGA hardware design by extending Simulink/Matlab in numerous ways in order to provide a powerful modeling environment. This study after illustrating the different implementations of SVM algorithms and its advantages and limitations, we clearly demonstrates the effeciency of System Generator comparing with other method, which make it an excellent high level design tool for FPGA design platform. It offer a decrease in time for testing and verification. Future research should develop System Generator as an efficient design tool valuable for large fields of application.

References

[1] John C. Platt. Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines. Technical report, Advances in kernel methods - support vector learning, 1998.

[2] V.Vapnik, “An Overview of Statistical Learning Theory” , IEEE TRANSACTIONS ON NEURAL NETWORKS, VOL. 10, NO. 5, SEPTEMBER 1999.

[3] Thomas Nowak, "Implementation and Evaluation of a Support Vector Machine on an 8-bit Microcontroller," Bachelor of Science. Faculty of computer science at Technical University of Vienna.

[4] A. Kerhet, M.Hu, F.Leonardi, A.Boni, D.Petri, "SVM-Like Algorithms and Architectures for Embedded Computational Intelligence", Technical Report March 2008, university of Trento, department of information and communication technology.

[5] B.Catanzaro, N.Sundaram, K.Keutzer, "Fast Support Vector Machine Training and Classification on Graphics Processors", Technical Report No. UCB/EECS-2008-11, Electrical Engineering and Computer Sciences University of California at Berkeley.

[6] A.Athanasopoulos,A.Dimou,V.Mezaris,I.Kompatsiaris:"GPU ACCELERATION FOR SUPPORT VECTOR MACHINES" Proc. 12th International Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS 2011), Delft, The Netherlands, April 2011.

[7] X.Zhang, Y.Zhang: " GPU Implementation of Parallel Support Vector Machine Algorithm with Applications to Intruder Detection" JOURNAL OF COMPUTERS, VOL. 9, NO. 5, MAY 2014

[8] U.J.Suryawanshi, S.R.Ganorkar: "Isolated Speech Recognition Using Digital Signal Processor" , Umarani J Suryawanshi et al, Int.J.Computer Technology & Applications,Vol 5 (4),1405-1408.

[9] M.Papadonikolakis,C-S.Bouganis: "A Novel FPGA-based SVM Classifier" Electrical and Electronic Engineering Department, Imperial College London.

[10] M.Ruiz-Llata, G.Guarnizo and M.Yébenes-Calvino: "FPGA Implementation of a Support Vector Machine for Classification and Regression" WCCI 2010 IEEE World Congress on Computational Intelligence July, 18-23, 2010 - CCIB, Barcelona, Spain.

[11] "System Generator for DSP": Getting Started Guide.