Universität Paderborn — Fakultät EIM-I
Fachgebiet Technische Informatik

Proposal

Master study

Incremental Learning with Support Vector Machines on Embedded Platforms

Institut für Informatik
Fakultät für Elektrotechnik,
Informatik und Mathematik
Universität Paderborn

Submitted by: Shankar Kumar, Matrikelnummer: 6809396

shanks@mail.upb.de

Created on: February 6, 2019

Supervisors: Dr. Hassan Ghasemzadeh Mohammadi, Prof. Dr. Marco Platzner

Problem

In domain of classification problems, *Support Vector Machines* (SVMs) are considered as one of the most common and wieldy used algorithms that can handle datasets with high dimensionality. It can also deal well with non-linear structures. The training process heavily depends upon the number of sample data points and become very challenging for the large practical datasets in the big-data era. Therefore, accelerating the training process for such an algorithm becomes of high interest. Moreover, online training of the algorithm, which refers to the updating of the weights, becomes important for the embedded platforms. In embedded systems, it is necessary to be able to implement the learning process which is the part of an optimization algorithm. For the streaming data, updating the weight of the trained model to capture the properties of the newly upcoming data points is a challenging task. This implementation has its own unique set of challenges and requirements. In this master thesis, some of the challenges of training SVMs as well as online updating of the trained model on embedded systems are addressed via providing the appropriate accelerator and approximation of the underlying algorithms.

The effective implementation on embedded platforms can provide several advantages for mobile applications. These include the followings:

- Fulfill with the energy constraints of embedded platforms
- Offering the capability of processing high dimensional inputs without exploiting high-latency access of cloud services
- Preventing security challenges owing to local data processing

To meet the requirements of embedded platforms, complex training algorithms should be modified in a more compact way since most of them having many redundant parameters. Moreover, an efficient sampling technique can be employed to reduce the number of required data points during training by dropping the similar data points. In this case similarity measure should be defined carefully to not impact the performance of the classification. Besides, sparsity models can be used to simplify the training and inference phase of the SVMs. Sparsity can be applied both on input data and the model computations, as both contributes to the final results of the algorithms.

Existing platforms, such as CPU, DSP, or GPU, are not efficient to run the sparse models owing to the weak support for sparse data utilization. Moreover, they suffer from high energy consumption and cost, which doesn't meet power and cost constraints of embedded devices. However, FPGAs are very promising to be exploited for the implementation of the DNNs as they offer the amazing opportunity of reconfiguration which provides the possibility of efficient hardware utilization for sparse DNNs.

In this master thesis, the acceleration and approximation of SVM training algorithms via offering appropriate hardware accelerators and applying sparsity techniques respectively. The approximation techniques are investigated for both inputs and the model.

Then, the performance of the obtained models is examined in term of accuracy and runtime. Finally, the best obtained model is employed on an FPGA board (e.g., PYNQ), and the characteristics are studied as well.

Objective

The objective of the master thesis are as follows:

- Understanding SVM training algorithms (e.g., SMO) in term of computational complexity
- Explore the appropriate architectures for accelerating the SVM training process
- Learning approximation technique to simplify training in SVM
- Learning how to efficiently model the algorithm on an embedded SoC board

Method

First work phase: In this phase the details of the most common available training algorithms are studied for a general multi-class classification problem. Beside, a software implementation of the selected algorithm is exercised to find the computational bottleneck of the target algorithm.

Second work phase: Here, the accelerator architectures for the SVM training are explored and a new architecture is proposed to speed up the training process based on the required extracted in the previous phase.

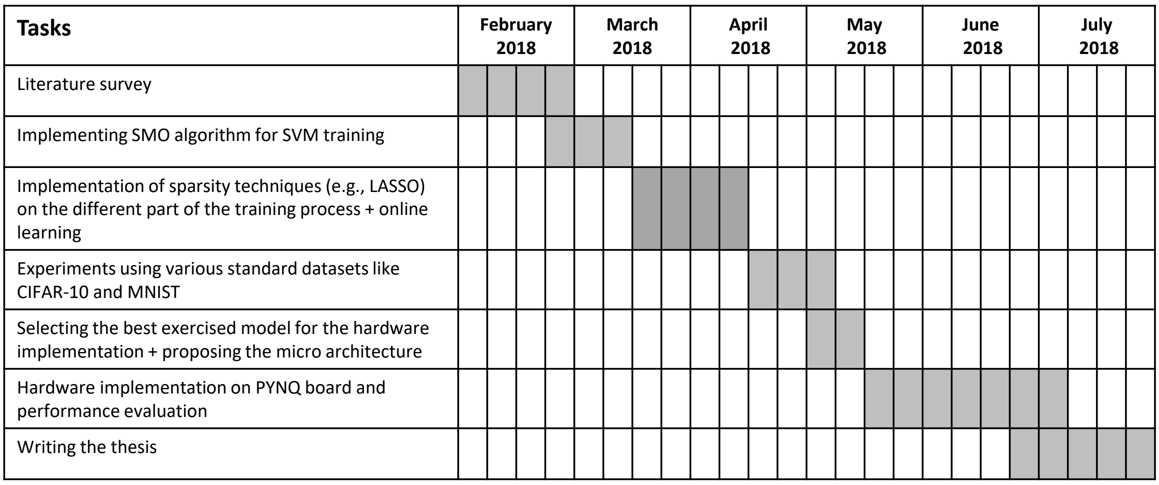
Third work phase: In this phase, approximation techniques like sparsity methods (e.g., LASSO) is applied on the input data and also on the various parts of the target algorithm and the impact of the various amount of approximations is evaluated on the performance and quality of the training step. Improvements on the runtime, and required hardware resources are also exercised.

Fourth work phase: After finding the most appropriate sparse model for the target SVM, the selected model is conveyed to the target FPGA board. In this step, an appropriate architecture for the target algorithm is proposed that can efficiently perform the computation of the both training and inferences for a given test datasets. The architecture should take to account the required needs of the SVM training and inference.

The documentation and the writing of the master thesis should take place parallel to all other work. In particular, the remaining X weeks should be used for this.

Schedule

The schedule shown in Fig. 1 contains the steps detailed in the procedure section, with an overview of the timing.



Tasks	February 2018				March 2018				April 2018				May 2018				June 2018				July 2018			
Literature survey	▨	▨	▨	▨																				
Implementing SMO algorithm for SVM training				▨	▨	▨																		
Implementation of sparsity techniques (e.g., LASSO) on the different part of the training process + online learning							▨	▨	▨	▨														
Experiments using various standard datasets like CIFAR-10 and MNIST											▨	▨	▨											
Selecting the best exercised model for the hardware implementation + proposing the micro architecture													▨	▨										
Hardware implementation on PYNQ board and performance evaluation															▨	▨	▨	▨	▨	▨	▨			
Writing the thesis																				▨	▨	▨	▨	▨

Figure 1: The schedule for the master thesis