A Hardware Efficient Support Vector Machine Architecture for FPGA

Kevin M. Irick, Michael DeBole, Vijaykrishnan Narayanan, Aman Gayasen
Department of Computer Science and Engineering
The Pennsylvania State University, PA, USA
{irick,debole,vijay}@cse.psu.edu, aman.psu@gmail.com

Abstract

In real-time video mining applications it is desirable to extract information about human subjects, such as gender, ethnicity, and age, from grayscale frontal face images. Many algorithms have been developed in the Machine Learning, Statistical Data Mining, and Pattern Classification communities that perform such tasks with remarkable accuracy. Many of these algorithms, however, when implemented in software, suffer poor frame rates due to the amount and complexity of the computation involved. This paper presents an FPGA friendly implementation of a Gaussian Radial Basis SVM well suited to classification of grayscale images. We identify a novel optimization of the SVM formulation that dramatically reduces the computational inefficiency of the algorithm. The implementation achieves 88.6% detection accuracy in gender classification which is to the same degree of accuracy of software implementations using the same classification mechanism.

1. SVM Classification

The SVM classifier utilizing the Gaussian Kernel is given by equation 1.

$$Classification(x) = \sum_{i=0}^{NUM\_SV} a_i e^{-\left(\frac{(x_0 - z_{i,0})^2 + \ldots (x_{n-1} - z_{i,n-1})^2}{2o^2}\right)}$$

Equation 1

1.1. Input Representation

In the case of gender classification we apply the SVM to a fixed 30x30 image window represented as a 900 element pixel vector. Each element in the vector is an 8-bit unsigned grayscale pixel value with magnitude ranging from 0 to 255. Support vectors are defined equivalently.

1.2. Kernel Optimization

First we analyze the range of values that result from the element-wise difference of the input image vector and a support vector when calculating the Euclidean Norm. Since the elements in the input and support vectors are 8-bit values, their difference is in the range of 255 to -255 which can be encoded as a 9-bit signed integer. Note that in equation 1 we are interested in the squared differences. Thus we can neglect negative differences and encode the absolute differences as 8-bit magnitudes.

$$D_i(x,z) = (x_i - z_i)^2$$

$$S_i(x,z) = \frac{D_i(x,z)}{2o^2}$$

$$Gaussian(x,z) = e^{-\sum_{i=0}^{n-1} S_i(x,z)} = \prod_{i=0}^{n-1} e^{-S_i(x,z)}$$

$$Classification(x) = \sum_{j=0}^{NUM\_SV} a_j \prod_{i=0}^{n-1} e^{-S_i(x,z_j)}$$

Equations 2-5

$$Classification(x) = LogSum_{j=0,NUM\_SV}\left(a_j + \sum_{i=0}^{n-1} SLNS\left(e^{-S_i(x,z_j)}\right)\right)$$

Equation 6

Equations 2 through 5 describe the Gaussian kernel as the product of the Exponential function evaluated separately for each of the S_i terms. The kernel is now factored into a series of functions of the absolute difference of x and z. The resulting function is efficiently stored in a 256 entry lookup-table.

To further reduce the complexity of the implementation we perform the series multiplication in the Signed Logarithm Number System, SLNS[3], which allows us to replace multiplication with addition as shown in Equation 6. The efficiency gained from operating in the SLNS, however, is almost always

978-0-7695-3307-0/08 $25.00 © 2008 IEEE
DOI 10.1109/FCCM.2008.40

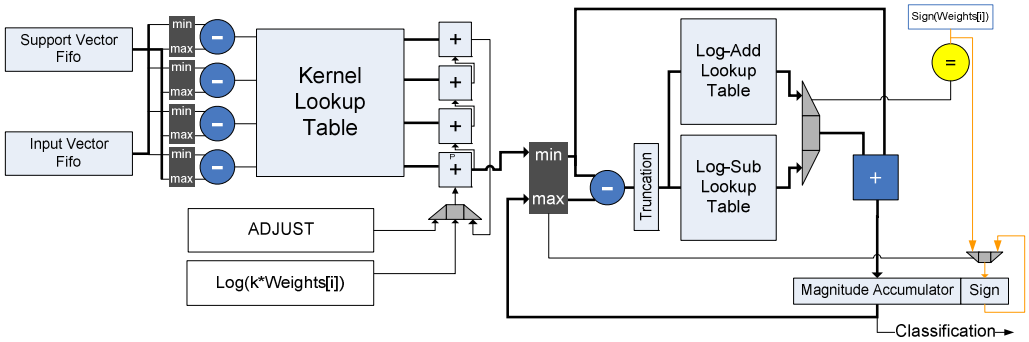

Figure 1

outweighed by the cost in converting to and from the logarithm domain: evaluating Log(x) and AntiLog(x) respectively. Our architecture circumvents the need to evaluate these functions by storing Signed-Log(Gaussian(x)) in the kernel lookup table.

2. System Architecture

Figure 1 is a block diagram of our hardware efficient SVM utilizing lookup-tables for the Gaussian kernel evaluation and SLNS arithmetic units for backend computation. The lookup-table outputs are delivered into the inputs of a cascade adder. Prior to completion of kernel evaluation the primary adder, denoted with a P in figure 1, accumulates the four partial sums, adds the weight term, and adds an adjustment term to account for SLNS conversion.

Once kernel evaluation completes, the current positive weighted result is added to the accumulation of previous weighted kernel evaluations. Because both the newly computed result is signed -as determined by the sign of the weight associated with the current iteration- and the current accumulation is signed -as determined by its sign bit - it will be necessary to perform either sign-magnitude addition or subtraction in the SNLS domain.

3. Experimental Results

We have tested our implementation of the SVM algorithm for the problem of gender classification against a commercial database of 3,454 male/female annotated images. The SVM contains 629 support vectors with each support vector containing 900 elements. Table 1 summarizes the results for double precision and fixed-precision of varying fractional bit-widths.

Table 1

	Misclassifications	Accuracy
DP	393	88.6%
FP 1.0.31	393	88.6%
FP 1.0.15	399	88.4%
FP 1.0.12	567	83.5%

Table 2

	DSP48s	BlockRAMs	Logic Slices
1.0.31	11	11	592
1.0.15	11	9	564
1.0.12	11	9	540

The SVM architecture can perform classification of 1,100 30x30 images, each consisting of 629 support vectors in one second when operating at 100 MHz.

4. References

[1] C.J.C. Burges, "A Tutorial on Support Vector Machines for Pattern Recognition", Data Mining and Knowledge Discovery, 1998, pp. 121-167.

[2] J.E. Volder, "The CORDIC Trigonometric Computing Technique", IRE Trans. On Electronic Computers, Sept. 1959.

[3] E.E. Swartzlander, and A.G. Alexopoulos, "The Sign/Logarithm Number System", IEEE Trans. On Computers, December 1975, pp. 1238-1242.

[4] D. Anguita,, A. Boni, and S. Ridella, "A Digital Architecture for Support Vector Machines: Theory, Algorithm, and FPGA Implementation", IEEE Trans. on Neural Networks, VOL. 14, 2003, pp. 993-1009.

[5] S.J. Melnikoff, and S.F. Quigley, "Implementing log-add algorithm in hardware", IEE Electronics Letters, VOL. 39, 31 March 2003, pp. 939-940.