A Comparison of Artificial Neural Network(ANN) and Support Vector Machine(SVM) Classifiers for Neural Seizure Detection

Mohamed A. Elgammal*, Hassan Mostafa*‡, Khaled N. Salama§ and Ahmed Nader Mohieldin*
*Electronics and Communications Engineering Department, Cairo University, Gize 12613, Egypt
‡Nanotechnology Department, Zewail City for Science and Technology, Egypt
§King Abdullah University of Science and Technology (KAUST), Saudi Arabia
{mohamed.adel@cu.edu.eg, hmostafa@uwaterlo.ca, khaled.salama@kaust.edu.sa, anader2000@yahoo.com}

***Abstract*—In this paper, two different classifiers are software and hardware implemented for neural seizure detection. The two techniques are support vector machine(SVM) and artificial neural networks(ANN). The two techniques are pretrained on software and only the classifiers are hardware implemented and tested. A comparison of the two techniques is performed on the levels of performance, energy consumption and area. The SVM is pretrained using gradient ascent (GA) algorithm, while the neural network is implemented with single hidden layer. It is found that the ANN consumes more power than the SVM by a factor of 4 with almost the same performance. However, the ANN finishes classification in much less number of clock cycles than the SVM by a factor of 34.**

***Index Terms*—support vector machine (SVM), artificial neural network (ANN), neural seizure detection.**

I. Introduction

Epilepsy is the name of a brain disorder characterized by recurrent and unpredictable interruptions of normal brain function, called epileptic seizures. Two-thirds of the patients achieve sufficient seizure control from anti-convulsive medication which is called Anti-epileptic drugs (AEDs) [1], and another 8–10% could benefit from respective surgery. For the remaining 25% of patients, no sufficient treatment is currently available [2]. For those who are untreatable using AEDs or surgery, an electrical stimulation is used to reduce the effect of the epileptic seizure. Hence, automatic seizure detection systems are proposed. Omar et al. [3] proposed a low-power implantable seizure detection processor.

Automatic seizure detection algorithms mainly consists of four stages. The first stage is measuring EEG signals through electrodes. Several research work is done in this field to improve the EEG measurement efficiency [4]. The second stage is preprocessing stage in which the signal is cleaned up from unwanted noise. The third stage is the feature extraction stage. Different features are extracted from EEG and the choice of these features greatly influence the overall efficiency of seizure detection. Features are extracted from time domain [5], frequency domain [6] or time-frequency (Wavelet) domain. The fourth stage is the classification block which is responsible of detecting seizure occurrence based on the extracted features. Many machine learning techniques are used in classification. In this paper, two common techniques are implemented and compared for neural seizure detection. These two techniques are support vector machine (SVM) with linear kernel and artificial neural network (ANN) with single hidden layer.

Section II provides a background on the support vector machine algorithm and the detailed hardware architecture proposed of the SVM classifier. Section III provides a background on the artificial neural network (ANN) and the detailed hardware architecture proposed. Section IV describes the simulation setup used in this paper, the simulation results, and provides analysis and discussion of the obtained results. Finally, some conclusions are drawn in Section V.

II. Support Vector Machine (SVM) Learning

A. Algorithm

Support vector machine (SVM) is a supervised machine learning technique that is one of the most popular classification techniques [7]. Training in SVM is performed using many algorithms like gradient ascent (GA) or sequential minimal optimization (SMO). SVM is searching for the hyperplane that gives the largest margin between the two sets of data. Finding this hyperplane is a problem of solving a quadratic programming problem.

The hyperplane is defined by the following equation :

$$w.\varPhi(x) + b = 0 \tag{1}$$

where w is the normal to the hyperplane, $\varPhi(x)$ is the mapping function used to map each input vector to the feature space and b is the bias.

The optimization problem of finding the hyperplane with largest margin is formulated as follows:

$$\underset{\alpha}{min}\psi\left(\alpha\right) = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} y_i y_j K\left(x_i, x_j\right)\alpha_i\alpha_j - \sum_{i=1}^{N}\alpha_i \tag{2}$$

subject to $\sum_{i=1}^{N} y_i\alpha_i = 0$, $0 \leq \alpha_i \leq C$, and $i = 1, ..., n$

where x_i is the ith input vector, y_i is the corresponding class, α is Lagrange multiplier. Kernel functions K might be linear, polynomial, exponential or any other type. The penalty parameter C should be selected carefully for each data set. There is a trade-off in choosing the value of C. If C is selected large, the weight of any wrong classified point is very large so the convergence of the problem takes large number of iterations. If C is selected small, some errors are allowed to maximize the margin and get the solution in fewer number of iterations than the large C scenario.

The kernel used in the implementation is linear kernel which is calculated as follows:

$$K(x_i, x_j) = x_i.x_j \tag{3}$$

To solve this minimization problem, gradient ascent uses an iterative technique. It depends on taking steps towards the optimum solution proportional to the value of the gradient of function at that point. The formula of updating α in each iterations as following:

$$\alpha_{\text{i}}^{new} = \alpha_i - step.y_i \left(\sum_{j=1}^{N} \alpha_i y_i K(x_i, x_j) + b \right) \tag{4}$$

,

such that $0 \leq \alpha_i^{new} \leq C$

To get the new bias b_{new}, substitute in the following formula by x_i, y_i of any of the support vector machine points that are correctly classified.

$$b_{new} = y_i - \sum_{j=1}^{N} \alpha_i y_i K(x_i, x_j) \tag{5}$$

After the completion of training phase, the classification phase starts. For any input vector x_{test}, by substituting in the following formula using the final value of $\alpha's$ and b, the corresponding class y_{test} is calculated.

$$y_{test} = \sum_{j=1}^{N} \alpha_j y_j x_{test}.x_j + b \tag{6}$$

B. Hardware Implementation

The training of SVM is done offline. Hence, only the SVM classifier needs to be hardware implemented. Figure 1 shows the architecture of the top level design of the SVM classifier which consists of 6 main block: three ROM blocks, classifier block and inner product block.

The first ROM block is used to save the input vectors of the support vector points. The width of this ROM is the same as the data width, while the depth is the number of support vectors multiplied by the number of the classification problem dimensions.

The second ROM block is used to save the values of non-zero α's. The width of this ROM is the same as the data width, while the depth is the number of support vectors.

The third ROM block is used to save the values of the true labels of the support vector points. The width of this ROM is one bit, while the depth is the number of support vectors.

The finite state machine (FSM) is responsible of generating the addresses of the three ROMs and the enable signal of classifier block.

The classifier block is the main block of the architecture. First, each α is multiplied by its corresponding label y. As the implementation used for negative numbers is sign-magnitude implementation, the multiplication is performed using an XOR gate instead of a multiplier. The value of $\alpha_i.y_i$ is saved in a register. An inner product block of size equal to the number of dimensions is used to multiply the input test vector with the input vector of the i^{th} support vector point. The output of the classifier block is fed to the inner product block to calculate the class.

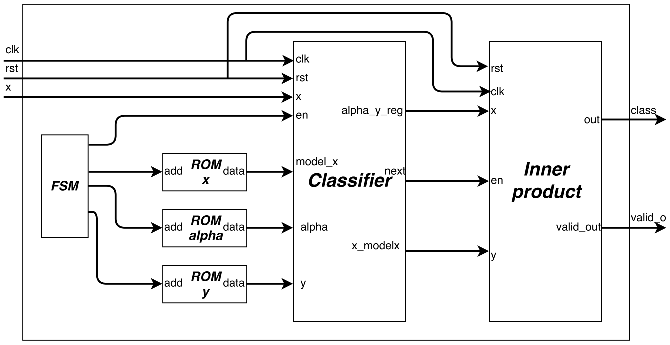

Figure 1. Top level SVM classifier block diagram

The inner product block is a multiple-add block with only one adder and one multiplier that multiply two vectors of size equal to the number of non-zero α's. The output of this block is the class and a valid out signal.

Different approximate computing techniques are used in implementing the proposed SVM classifier. First of all, fixed point is used instead of the computationally expensive floating point. Using software simulation results, a 16-bit word length is enough for achieving the same performance (i.e., accuracy, in SVM). Another technique for energy saving is computation skipping. For example, for non support vectors ,$\alpha = 0$.Hence, these points are skipped in the computations. Moreover, the step is chosen to be multiple of 2, to use a shifter instead of a multiplier.

III. Artificial Neural Network (ANN)

A. Algorithm

Over the past twenty years, many methods inspired by the understanding of the structure and function of the biological neural networks are evolved. One of these methods is the artificial neural network (ANN). Neural networks are used in various applications such as classification, pattern recognition, and data analysis. ANN mainly consists of an input layer, one or more hidden layers and one output layer. Each layer consists of multiple neurons and different weights are given to the connections among these neurons. Each neuron in the input layer takes in one data source. The output of each input layer neuron is input for each of the hidden layer neurons [8]. Finding the weight of each neuron is performed in the training phase. After the neural network is trained, any new input vector is fed to the input layer. The value of each node is calculated by multiplying the input node value by the connection weight and adding all the values entering this node. To detect seizure and differentiate between seizure and non seizure epochs, the architecture of the ANN used is a single

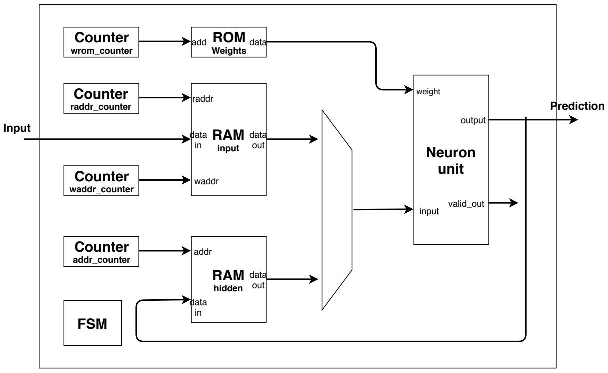

Figure 2. Top level ANN classifier block diagram

hidden layer with 6 neurons. The activation function used is the Sigmoid function.

B. Hardware Implementation

The architecture of the ANN classifier consists of ROM block, two RAM blocks, four counters, neuron block and finite state machine as shown in Figure 2.

A ROM block is used to save the weights of each connection. A single data port RAM is used to save the values of each node (neuron) of the hidden layer. A double data port RAM is used to save the values of each node of the input layer. Four counters are used to generate the addresses of the ROM, single data port RAM and double data port RAM. The neuron block is an multiply-accumulate block that consists of multiplier, adder, register and activation function block. The activation function used is the Sigmoid function and is implemented as a combinational circuit. The FSM is responsible for controlling the overall system.

Different approximate computing techniques are used in implementing the proposed ANN. First of all, fixed point is used instead of the computationally expensive floating point. Using software simulation results, a 16-bit word length is enough for achieving the same performance (i.e., accuracy, in ANNs). Reducing the word length less than 16 bits achieves more power saving with the cost of performance degradation. Another technique for energy saving is the adoption of approximate implementation of the activation functions. For example, instead of implementing the exponential function for calculating the Sigmoid function, a Piece-Wise Linear (PWL) approximation is used to reduce the power consumption.

IV. Simulation setup, Results, and Discussion

The classification techniques implemented in this paper are tested and simulated on neural seizure detection. EEG signals of patients are first processed on 4-sec window as proposed by Aya et al. [9]. Then, different features are extracted. The features used in our test are: Fractal Dimension, Hurst exponent and coastline as proposed by Elgammal et al. [10].

Both ANN and SVM classifiers are applied on the input vectors to detect seizure. The SVM used is the Lagrangian SVM with linear kernel. The ANN used is a three-layers network with six neurons in the hidden layer. The activation function is the Sigmoid function. The architecture of both algorithms is chosen carefully to give almost the same performance.

Table I
Classification performance for seizure detection

Algorithm	sensitivity	specificity	accuracy
SVM	96.23	97.90	97.89
ANN	96.5	97.88	97.88

CHB-MIT Scalp dataset from PhysioNet library was used to verify the implementations. The dataset was collected at the Children's Hospital Boston from subjects with intractable seizures. Recordings were collected from 22 patients (5 males, and 17 females). The age of the subjects was from 3 to 22 in males and from 1.5 to 19 in females. The signals were sampled at 256 sample per second with 16-bit resolution. For each patient, 23 channels were recorded from different electrodes. The dataset comes with labeling on the epileptic sessions for different patients[11]. Data is divided into training and testing data.

The software used for modeling, feature extraction, classification and performance measurements is implemented using MATLAB 2015a. Xilinx ISE 14.2 is utilized to design and develop the VLSI architecture of the algorithms. The design is synthesized on Xilinx Spartan6 FPGA. For the implementation on ASIC, Synopsys Design Compiler(DC) B-2008.09 with UMC 130nm library is adopted.

Results are collected in two main phases. The first phase is evaluating the performance simulation results. The second phase is calculating the hardware implementation metrics such as area, power and maximum frequency for both ASIC and FPGA implementations as illustrated in the next two subsections.

A. Simulation Results

Both of the pretrained SVM and ANN are used to classify any new input vector. The performance for each algorithm is evaluated through three different metrics which are commonly used in neural seizure detection namely: sensitivity, specificity and accuracy. Sensitivity is the algorithm ability to detect seizures correctly, whereas specificity is the algorithm ability to avoid false alarms. Accuracy is a combining matrix between both of them [12].

B. Implementation Results

The hardware implementations of SVM and ANN classifiers are presented on both FPGA and ASIC platforms. Table II lists the resources in Xilinx Spartan6 FPGA such as LUTs and registers slices. Table II also tabulates the maximum frequency and the dynamic power consumption of each algorithm. It is found that the proposed ANN implementation uses less utilization than that used by Liu et al. at [13].

Table III shows the area, power of each algorithm implemented in ASIC platform using UMC 130nm. It also includes the number of clock cycles that each algorithm takes to finish classification and the power delay product (PDP). The PDP is calculated as the multiplication result of power consumption

Table II
FPGA IMPLEMENTATION RESULTS

Algorithm	Area		Max frequency MHz	Power mW
	LUT	Registers		
SVM	293	137	13	1
ANN	401	256	139	3

Table III
ASIC UMC 130NM IMPLEMENTATION RESULTS

Algorithm	Area (nm^2)	Power (μW)	no. of cycles	PDP
SVM	3963	2.15	1020	2193
ANN	16040	8.08	30	242.4

and the number of clock cycles required to finish classification of one input vector. The clock frequency used is 1 MHz.

As shown in Table I, the two algorithms with the chosen parameters give almost the same performance. This makes the comparison of the power, area and energy as fair as possible. The appropriate choice of the applied features helps in achieving very high sensitivity using linear kernel in the SVM and using only one hidden layer with only 10 neurons in the hidden layer. This performance exceeds that obtained by Yuan et al. by using SVM with radial basis function (RBF). Yuan et al. got sensitivity ranging from 73.5% to 95% using different features.

In Table II, It is obvious that the SVM algorithm has the advantage of less utilization, higher maximum frequency and less power consumption than the ANN algorithm. However, the main disadvantage of the SVM algorithm is the required large number of clock cycles to classify every new data point, which reaches up to 1020 clock cycle compared to 30 clock cycle only for the ANN algorithm. This very large number of clock cycle is due to the fact that neural seizure detection problem is a very complex one. Hence, the SVM technique has many support vectors and the inner product occurs for every testing point is very large. However in the case of ANN, only the output of each node is calculated through an add-multiply block. Table III shows the comparison between the implementation of both algorithms on ASIC platform in area and power consumption. As the throughput of each algorithm is different, power consumption is not a good comparison metric. Hence, power delay product is calculated. Although SVM algorithms consumes less power than the ANN algorithm, the power delay product is much larger.

V. Conclusion

Many algorithms are used in classification. Both ANN and SVM are used in neural seizure detection classification efficiently. For the same performance, the ANN classifier consumes less energy than the SVM classifier for each input vector. However, the instantaneous power consumed in the ANN classifier is more than that of the SVM classifier.

Acknowledgment

This work was partially funded by ONE Lab at Zewail City of Science and Technology and Cairo University, NTRA, ITIDA, ASRT, Mentor Graphics, NSERC.

References

[1] A. Varsavsky, I. Mareels, and M. Cook, *Epileptic seizures and the EEG: measurement, models, detection and prediction*. CRC Press, 2016.

[2] F. Mormann, R. G. Andrzejak, C. E. Elger, and K. Lehnertz, "Seizure prediction: the long and winding road," *Brain*, vol. 130, no. 2, pp. 314–333, 2006.

[3] S. Omar, H. Mostafa, T. Ismail, and S. Gabran, "Low-power implantable seizure detection processor," in *2015 IEEE International Conference on Electronics, Circuits, and Systems (ICECS)*. IEEE, 2015, pp. 496–497.

[4] S. F. Cogan, "Neural stimulation and recording electrodes," *Annu. Rev. Biomed. Eng.*, vol. 10, pp. 275–309, 2008.

[5] B. Boashash, *Time-frequency signal analysis and processing: a comprehensive reference*. Academic Press, 2015.

[6] I. Mporas, V. Tsirka, E. Zacharaki, M. Koutroumanidis, and V. Megalooikonomou, "Evaluation of time and frequency domain features for seizure detection from combined eeg and ecg signals," in *Proceedings of the 7th International Conference on PErvasive Technologies Related to Assistive Environments*. ACM, 2014, p. 28.

[7] P. Bhuvaneswari and J. S. Kumar, "Support vector machine technique for eeg signals," *International Journal of Computer Applications*, vol. 63, no. 13, 2013.

[8] D. OâĂŹLeary and J. Kubby, "Feature selection and ann solar power prediction," *Journal of Renewable Energy*, vol. 2017, 2017.

[9] A. Elzahy, M. A. Elgammal, H. S. Mohammed, and H. Mostafa, "Optimal eeg window size for neural seizure detection," in *2019 8th International Conference on Modern Circuits and Systems Technologies (MOCAST)*. IEEE, 2019, pp. 1–4.

[10] M. A. Elgammal, O. A. Elkhouly, H. Elhosary, M. Elsayed, A. N. Mohieldin, K. N. Salama, and H. Mostafa, "Linear and nonlinear feature extraction for neural seizure detection," in *2018 IEEE 61st International Midwest Symposium on Circuits and Systems (MWSCAS)*. IEEE, 2018, pp. 795–798.

[11] A. H. Shoeb, "Application of machine learning to epileptic seizure onset detection and treatment," Ph.D. dissertation, Massachusetts Institute of Technology, 2009.

[12] G. S. Maximous, A. M. El-Gunidy, H. Mostafa, T. Ismail, and S. Gabran, "A new sensitivity-specificity product-based automatic seizure detection algorithm," in *2017 Japan-Africa Conference on Electronics, Communications and Computers (JAC-ECC)*. IEEE, 2017, pp. 107–110.

[13] J. Liu and D. Liang, "A survey of fpga-based hardware implementation of anns," in *Neural Networks and Brain, 2005. ICNN&B'05. International Conference on*, vol. 2. IEEE, 2005, pp. 915–918.

[14] Q. Yuan, W. Zhou, S. Li, and D. Cai, "Epileptic eeg classification based on extreme learning machine and nonlinear features," *Epilepsy research*, vol. 96, no. 1-2, pp. 29–38, 2011.

[15] B. Greene, S. Faul, W. Marnane, G. Lightbody, I. Korotchikova, and G. Boylan, "A comparison of quantitative eeg features for neonatal seizure detection," *Clinical Neurophysiology*, vol. 119, no. 6, pp. 1248–1261, 2008.

[16] C. Chen, Z. Liu, H. Li, R. Zhou, Y. Zhang, and R. Liu, "EEG Detection Based on Wavelet Transform and SVM Method," *IEEE International Conference on Smart Cloud*, pp. 241–247, 2016.

[17] I. A. Basheer and M. Hajmeer, "Artificial neural networks: fundamentals, computing, design, and application," *Journal of microbiological methods*, vol. 43, no. 1, pp. 3–31, 2000.