

Chapter 5

Performance Improvement of SOM by using Low Bit-Width Resolution

One of the prime reasons for NNs popularity in last decade is due to their ability to detect **patterns** **pattern** accurately. Compared to conventional **solutions** **solution** they scale exceptionally well because of their **inherit** parallelism **and are faster**. NNs are also **very power efficient**, since their power consumption during the inference (recall) mode is very low. Apart from the NN benefits that we describe before, the main strength of the NNs comes from their ability to learn the classification criteria directly from the input data. In essence **an** NN trained with genomic data compresses its inputs and encodes it in its structure. The NN then can act as a predictor that can be queried about specific features in a given genome, instead of attempting to output linear DNA sequences as done in the conventional assembly. In other words, the **ANN** **??** can invent an algorithm automatically that otherwise would have to be developed by the bioinformatics engineers. The algorithm is defined by the parameters of the networks, as they are developed during its training. That also provides with the possibility of **retaining** the network in order to update the identification features such as mutations and other genomic alterations that were not known before, without changing the algorithm itself. **do you mean re-training?**

Yang et al. **Prior work in [47] has** introduced Self-organizing maps (SOM) for rapid genome identification. **[47]** SOM uses a type of unsupervised learning called competitive ANN learning model. The model reduces the data dimensions and it clusters similar data together [25]. A trained SOM network does not require to go through the whole DNA sequence to recognize the pathogen, but only requires a small part of its DNA. SOM can be highly parallelized and such parallel implementation have been proposed for synchoros VLSI design **[??]** custom FPGA **[??]** and GPUs **[30, 35, 47]**. Another important aspect of SOM and other NNs is their robustness. NNs **have been proven to work with low bit resolution without sacrificing much of their**

you can't make such a general statement

accuracy [24]. In this work, we explore the limits of the SOM using different bit resolutions and the effect that it has on the accuracy of the SOM, as well as the benefits that this low resolution can provide for a hardware architecture.

5.1 Introduction

An emerging design paradigm that is able to achieve better energy efficiency by trading off the quality (e.g., accuracy) and effort (e.g., energy) of computation is approximate computing [51]. Many modern applications, such as machine learning and signal processing, are able to produce results with acceptable quality despite most of the calculations being computed imprecisely [48]. The tolerance of imprecise computation in approximate computing to acquire substantial performance gains is the basis for a wide range of architectural innovations [13]. It has been demonstrated that high-precision computations are often unnecessary in the presence of statistical algorithms [31, 50]. Zhang et.al. [50] report less than 5% of quality loss obtained by simulation of the real hardware implemented in a 45nm CMOS technology.

Representing data with reduced bit-widths by trading off accuracy is one of the popular low-power strategy. The benefit of using reduced bit-width is improved energy performance. This is because there is a reduction in the energy cost consumption for data transfers, which usually dominates the total energy consumption for such systems.

Gupta et al. present results where they train deep networks with 16 bits fixed-point number representations and stochastic rounding [19]. Talathi et al. show that the best performance with reduced precision can be achieved with 8 bits weights and 16 bits activation, which, if reduced to 8 bits, results in a 2% drop in accuracy [29]. Hashemi et al. look at a broad range of numerical representations applied to ANNs in both inputs and network parameters and analyze the trade-off between accuracy and hardware implementation metrics, and conclude that a wide range of representations is feasible with negligible degradation in performance [21].

We present a design space exploration of a self-organizing map (SOM) to analyze the impact of different bit resolutions on the accuracy, as well as its benefits. SOM uses a type of unsupervised learning called the competitive ANN learning model. To lower the energy consumption, we exploit the robustness of SOM by successively lowering the resolution to gain efficiency and lower the implementation cost. We do an in-depth analysis of the reduction in resolution vs. loss in accuracy. We present an FPGA implementation of SOM aimed at a bacterial recognition system for battery-operated clinical use where the area, power, and performance are of critical importance. Using this implementation, we demonstrate that a 1%

loss in accuracy with 16-bit representation can yield significant savings in energy and area.

5.2 Background

In this section, we briefly introduce the SOM algorithm in [47] which has been used to recognize bacterial genomes.

As shown in figure 5.1, a SOM network is organized in a circle to better match genomic data. Each SOM network hosts N neurons. Each neuron has a weight vector which has the same size as the training vector, assume it's M . Therefore, the weight matrix for the whole SOM network would be $W = M \times N$. Input sequences, either for training or for inference, are vectors of size M . Each element in those input vectors represents a nucleotide.

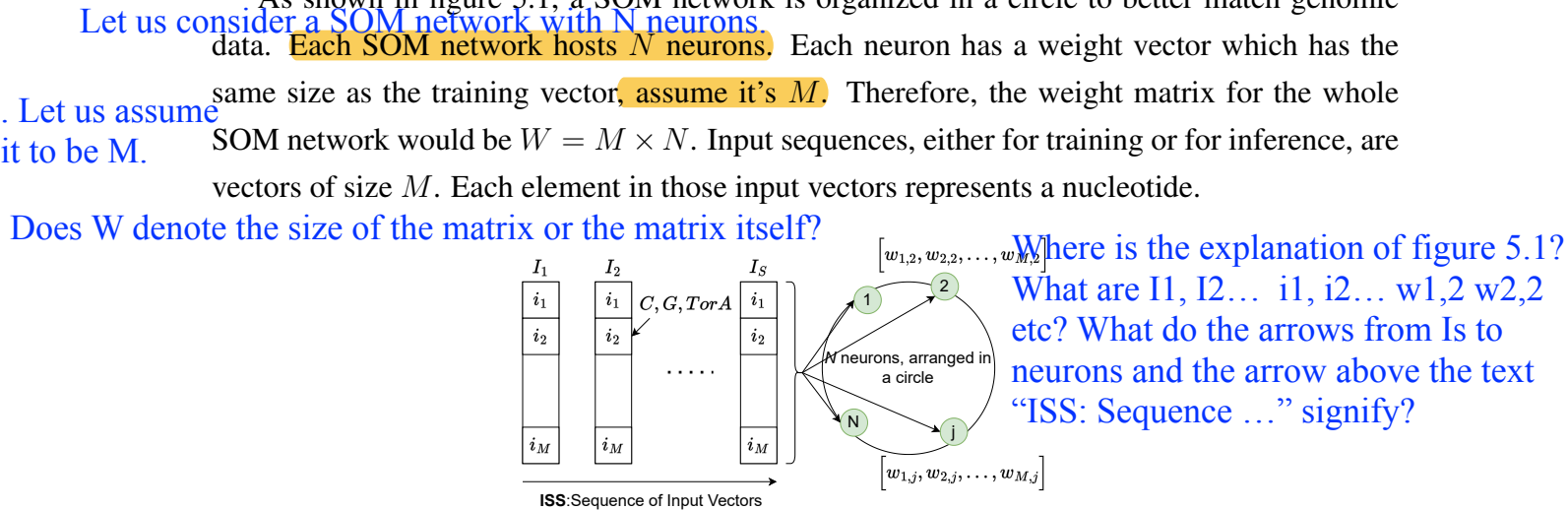


Figure 5.1: SOM based Genomic Identification

The actual training and inference processes are described in algorithm 4. The objective is to embed genomic features of a specific bacteria into a SOM. When DNA sequences of unknown bacteria are compared with trained SOM networks, the SOM network trained with the same bacteria as the unknown test bacteria, must have the highest correlation. Based on the correlation, we can identify the unknown bacteria.

Before giving line by line description of the algorithm, give the basic idea and an overview.

The training phase is for a specific bacterial genome. The complete bacterial DNA sequence is chopped randomly into a set of fixed size training sequence, the IS in algorithm 4 line 0. The weight matrix is initialized by random weights. On the line 1, a parameter β , initialized to 1.0, is used to control the converging process of SOM. In line 3 and 4, the actual training process first tries to find the neuron that best matches the particular input vector. Then, weights of the neighbourhood of the winning neuron are updated based on the distance between the target neuron and the winning neuron, as shown in line 6 and 7. Finally in line 8, the parameter β is decreased by a factor. After repeating the training process for all training vectors, the SOM will learn the features of that particular bacterial genomes. Each SOM network can be trained to recognize only one bacteria. In order to recognize a range of bacteria, many SOM networks need to be trained.

You have not explained all the variables, e.g. beta_min, decay_factor etc.

would have learnt

During test phase, some test DNA fragments from one unknown bacteria are send to each trained SOM network. The SOM network that correlates the test input sequence best, is chosen as the winning SOM and the bacteria it represents reveals the identity of the unknown test bacteria. The correlation measurement is marked by the score shown in line 10. The smaller the score is, the better it correlates with the input test vectors.

Algorithm 4 Pseudo code SOM learning and inference for genome identification

Algorithm Part 1: SOM training for 1 bacterial genome

Input N : Number of neurons

Input $I=[i_1, i_2, \dots, i_M]$: Input-Vector

Input $IS=I_1, I_2, \dots, I_S$: Sequence of Input-Vectors, each IS represents one bacterial genome

Input $W_{i,j}$: $M \times N$ weight matrix for 1 bacteria

1: $\beta_{min} = 0.01$; $decay_factor = 0.99$; $\beta = 1.0$

2: **for** $I_k \in IS$ **do**

3: $dist_{min} = \min_{j=1 \dots N} (\sum_{i=1}^M |I_{k,i} - W_{i,j}|)$

4: $j_{min} = j$ **where** $dist_j = dist_{min}$

5: **for** $j \in 1 \dots N$ **do**

6: $dist = \frac{N}{2} - ||j - j_{min}| - \frac{N}{2}|$

▷ toroid distance

7: $W_j = W_j - \frac{\beta}{2^{dist}} (W_j - I_k)$

8: $\beta = \min(\beta * decay_factor, \beta_{min})$

▷ decay β

Algoritm Part 2: SOM inference

Input $TIS=I_1, I_2, \dots, I_S$: Test Input Sequence of Input-Vectors for which the bacteria is to be identified

Input $W_{r,j,i}=R \times N \times M$: Weights for R bacteria

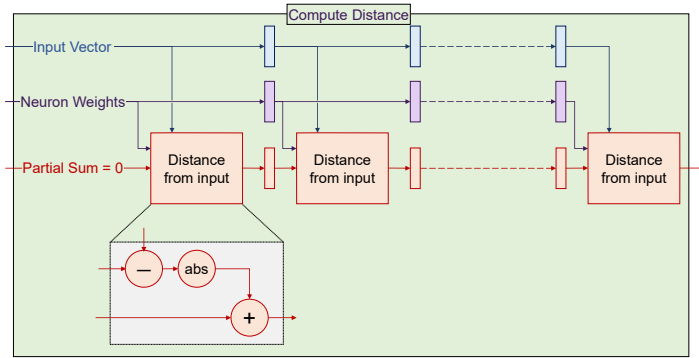
9: $Inferred_r$ is r with

10: $score = \min_{r=1 \dots R} \left[\sum_{k=1}^S \min_{j=1 \dots N} (\sum_{i=1}^M |I_{k,j} - W_{r,i,j}|) \right]$

5.3 Low bit-width FPGA Design of SOM

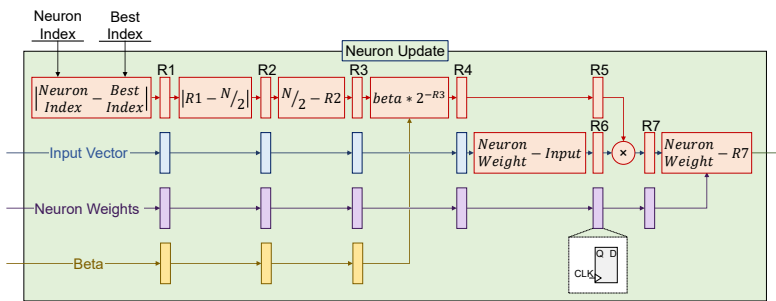
In this section, we present the FPGA implementations that were used to implement SOM for identification of bacterial genomes. The FPGA implementation is done on Xilinx Virtex7 485t chip for the identification of bacterial genomes. *each of the*

A custom semi systolic array was hand crafted, for *different* bit width implementations, to analyze the area versus energy trade off. The design takes a vector of weights as input and finds the neuron in the network closest to *that* input vector. This neuron is referred as *the* winning neuron of the network. The design outputs the distance between the input vector and *the* closest



explanation of the figure?

Figure 5.3: Compute Distance Module.



explanation of the figure?

Figure 5.4: Neuron Update Module.

Distance can start its operations for next $Input\ Vector=i+1$. *Compute Distance* and *Neuron Update* components can overlap their execution, processing different input vectors at same time. This design has overall $II = 23 + N$ cycles, where N is number of neurons in the network.

how do you get this?

is it latency or initiation interval?

5.4 Experimental Results And Analysis

In this section we present the results from our experiments. In subsection 5.4.1 we give the details of our experimental setup used for measuring the accuracy of the SOM. Subsections 5.4.2 describe the experimental setup and presents the results of the FPGA implementation.

5.4.1 Accuracy experimental setup

The SOM has been implemented with a range of fixed-point formats. With fewer bits, one naturally expect that the SOM network for bacterial identification to suffer from accuracy degradation. A MATLAB simulation model was created to analyze the accuracy loss when using fixed-point implementation. The experiment used a scaled down version of bacterial identification problem, to reduce the training time of the network. We consider that this does not compromise the resulting accuracy of the SOM. We trained 10 SOMs with 10 different bacteria DNA sequences. Each SOM network has 100 neurons inside, and each neuron has 20 weights. We trained the networks by two independent training processes running in parallel. One is implemented using double precision floating point and the other is implemented with fixed-point weights. After training, we used the trained networks to identify the unknown sequence and record their scores.

Two metrics are defined to evaluate the implementation accuracy. The *quantization error* is defined as the relative error between fixed-point format score (S_{fixed}) and double precision floating-point format score (S_{float}):

$$quantization_error = \frac{|S_{fixed} - S_{float}|}{S_{float}} \times 100\%$$

The *classification error* is defined as the ratio of the number of times the network classified (C_{false}) the bacterial strain to the total number of tests performed (C_{total}):

$$classification_error = \frac{C_{false}}{C_{total}} \times 100\%$$

What is the reference for this? Are you assuming that with floating point there is no classification error?

Classification error and quantization error are not completely independent, but they reveal the different aspects of fixed-point implementation accuracy. Quantization error shows the difference between fixed-point representation and golden double point reference. It doesn't change with the problem. **Classification error determines the final impact of the accuracy loss introduced by the fixed-point implementation, and is problem dependent.** Figure 5.5 presents the quantization error and the classification error depending on the bit resolution. From the figure, 8-bit representation completely fails the experiment, 12-bit has 39% classification error, 16-bit has <1%, 24-bit and 32-bit successfully pass the experiment with 0% of classification error. However, we should notice that though 16-bit pass the test, it still has about 10% quantization error.

We like to note that each SOM was trained with the specific fixed-point resolution and that the quantization error shows the difference of the *score* variable, see algorithm 4, between the

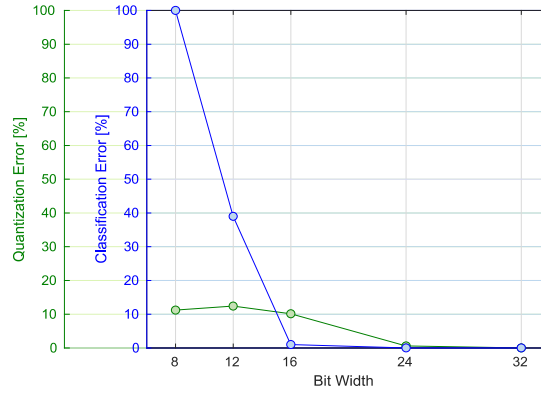


Figure 5.5: Quantization error compare to the identification error

fixed-point and the double precision. The quantization error is very important for the *score* variable of the SOM, since this variable is the one that decides which is the winning SOM, i.e. which SOM matches the input DNA sequence. It is important to guarantee sufficient bits for the *score* of each SOM. If the resolution is too low, the *score* of the SOMs after quantization will be identical even with low quantization error which makes it impossible to identify the bacteria correctly. This explains the fact that in Figure 5.5 the quantization error is low but the classification error is high in 8 to 12-bit resolution region.

5.4.2 FPGA experimental setup

In this section we present the experimental results of the FPGA implementation presented in section 5.3. The FPGA design is implemented with Vivado v.2016.4 used for synthesis and analysis of the HDL Designs. Our design is implemented in VHDL and validated using the Vivado simulator. Experimentation is done for different fixed point representations of weights by modifying parameters in VHDL code.

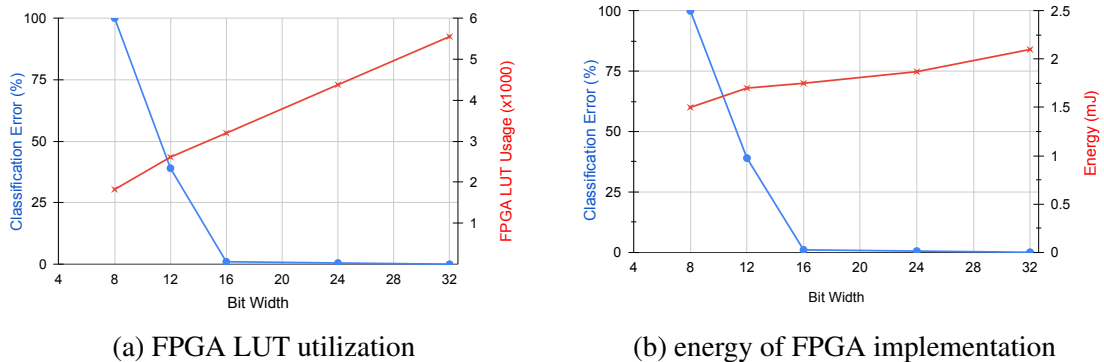


Figure 5.6: Area and energy comparison for different fixed-point format.

The area and power numbers for different weight resolutions are extracted from the reports generated by Vivado tool post placement and routing with a working frequency of 100 MHz. Table 5.1 compares the resources and area for 8, 12, 16, 24, and 32 bits fixed point formats, for a SOM network with 512 neurons. The second part of the table compares the average power in the different fixed point formats, for the same SOM.

Table 5.1: Resource Comparison of different fixed point formats

Resource	8b	12b	16b	24b	32b
LUTs	1823	2611	3196	4375	5549
Registers	3481	4679	5871	8255	10639
Slice	854	1158	1369	1809	2395
LUT FF Pairs	1007	1369	1750	2372	3043
B-RAM	4	6	8	11	15
DSP48E1	17	17	17	17	33
Bonded IOB	57	61	65	73	81
Power(W)	8b	12b	16b	24b	32b
Total Power	0.295	0.314	0.332	0.356	0.392
Dynamic	0.052	0.071	0.089	0.113	0.148
Device Static	0.243	0.243	0.243	0.244	0.244

Should these be linear wrt bit-width?

These figures should appear after table 5.1

The results are summarized in the Figure 5.6a and 5.6b. Both the amount of utilized LUTs and total energy in Joule is presented against the classification error. From the figures, we can easily conclude that, we can substantially reduce the resources used and the energy by using a 16-bit fixed-point representation, without losing accuracy. We can reduce the resources even further by moving to the 12-bit representation, by sacrificing 39% of the SOM accuracy.

5.5 Summary

In this work we explore the design space of a self-organizing map (SOM) used for rapid and accurate identification of bacterial genomes. This is an important health care problem because even in Europe, 70% of prescriptions for antibiotics is wrong. The SOM is trained on Next Generation Sequencing (NGS) data and is able to identify the exact strain of bacteria. This is in contrast to conventional methods that require genome assembly to identify the bacterial strain. SOM has been implemented on FPGA and shown to have better computational efficiency compared to GPUs. To further lower the energy consumption, we exploit the robustness of SOM by successively lowering the resolution to gain further improvements in efficiency and lower the

reference?

I don't see where it has been done

implementation cost without substantially sacrificing the accuracy. We do an in depth analysis of the reduction in resolution vs. loss in accuracy as the basis for designing a system with the lowest cost and acceptable accuracy using NGS data from samples containing multiple bacteria from the labs of one of the co-authors. The objective of this method is to design a bacterial recognition system for battery operated clinical use where the area, power and performance are of critical importance. We demonstrate that with 39% loss in accuracy in 12 bits and 1% in 16 bit representation can yield significant savings in energy and area.

In this work, ^{already said?} we implemented FPGA design for SOMs. The mapping support different bit width to enable trade-offs between accuracy and computation cost. Experiments has shown the big design space, resulting in different cost in terms of timing, area as well as energy, all of which are affected significantly by the fixed-point format representation. Through the experiment, we conclude that the SOM network with 16-bit fixed-point representation implemented on FPGA, ^{you are moving back and forth} has better benefits compared to other fixed-point formats with more bits. And 16-bit has acceptable classification error. Format with bits more than 16 no longer add any benefit to lower the classification error.

In future works, we plan to apply more sophisticated methods to scale down the bit width without losing too much accuracy. For example, the weights can be dynamically scaled after several epochs of training when the current fixed point format is not suitable anymore. We are confident that, with such approximate computing techniques, we could ^{possible} reduce the resolution to 8 bits with acceptable loss of accuracy and, by extension, ^{possibly} the implementation cost of SOM networks on hardware.