TOTEM: A HIGLY PARALLEL CHIP FOR TRIGGERING APPLICATIONS WITH INDUCTIVE LEARNING BASED ON THE REACTIVE TABU SEARCH

To appear in: International Journal of Modern Physics C, 6(4):555–560, 1995

G. ANZELLOTTI, R. BATTITI, I. LAZZIZZERA, G. SONCINI, A. ZORAT
Università di Trento
Trento, Italy, I-38100

A. SARTORI, G. TECCHIOLLI
Istituto per la Ricerca Scientifica e Tecnologica
Trento, Italy, I-38100

and

P. LEE
University of Kent
Canterbury, United Kingdom, CT2 7NQ

The training of a Multi-Layer Perceptron (MLP) classifier is considered as a Combinatorial Optimization task and solved with the Reactive Tabu Search (RTS) method. RTS needs only forward passes (no derivatives), and it does not need high precision in the network parameters. A special-purpose VLSI architecture has been developed to take advantage of the limited memory and processing requirements of RTS: the final system realizes a very close coupling of hardware and training algorithm. The RTS algorithm and the design of the VLSI chip are discussed, together with the operational characteristics of TOTEM and some preliminary training and generalization tests on triggering tasks.

Keywords: high energy physics, triggering, inductive learning, special purpose processors

1. INTRODUCTION

Training algorithms for neural nets often impose serious constraints on the architectural layout of neural processors [3]. Derivative-based training algorithms such as backpropagation tend to require high-precision computation because of numerical problems related to the ill-conditioning of the Hessian matrix [4,5]. As a consequence, highly complex computational structures are often necessary and these may require microprogrammed devices with their related long development phases.

This paper presents a novel and VLSI friendly approach to the training problem: first the task is transformed into a *combinatorial optimization* problem, then it is solved with a heuristic method based on the construction of a *search trajectory*, the *Reactive Tabu Search* (RTS) [6].

RTS escapes rapidly from local minima, is applicable to non-differentiable and

even discontinuous functions and is very robust with respect to the choice of the initial configuration. In addition, by fine-tuning the number of bits for each parameter one can decrease the size of the search space, increase the expected generalization and realize cost-effective VLSI.

In the following sections we summarize the RTS algorithm (Sec. 2), present a chip that has been designed to implement RTS-trainable neural nets (Sec. 3), and discuss the results of a selected application in HEP (Sec. 4).

2. The Reactive Tabu Search

RTS is a "local search" algorithm in the framework of Glover's Tabu Search (TS)[7]. TS optimizes a function f by using an iterative "greedy" component (*modified local search*) to bias the search toward points with low f values and by incorporating *prohibition strategies* to avoid the occurrence of cycles.

A *search trajectory* $X^{(t)}$ is generated in the admissible search space $\mathcal{X}$, where the successor of a point X is selected from a *neighborhood*. For the following discussion $\mathcal{X}$ is the set of all binary strings with a finite length L: $\mathcal{X} = \{0,1\}^L$ and the neighborhood is obtained by applying the *elementary moves* $\mu_i (i = 1, ..., L)$ that negate the i-th bit of the string. At each step of the iterative process, the selected move is the one that produces the lowest value of the cost function f in the neighborhood among the non-prohibited moves. As soon as a move is applied, the inverse move (that is equal to the move in the case of binary strings) is temporarily prohibited for T steps.

The parameter T regulates the amount of *diversification*. RTS [6] [8] uses a simple mechanism to deal with cycles that are not avoided by using the basic "prohibition" scheme and a way to change T during the search so that the value $T^{(t)}$ is appropriate to the local structure of the problem (therefore the term "reactive"). The *prohibition period* T is equal to one at the beginning (the inverse of a given move is prohibited only at the next step), it increases only when there is *evidence* that diversification is needed, it decreases when this evidence disappears. In detail: the evidence that diversification is needed is signaled by the repetition of previously-visited configurations. All configurations found during the search are stored in memory. After a move is executed the algorithm checks whether the current configuration has already been found and it reacts accordingly (T increases if a configuration is repeated, T decreases if no repetitions occurred during a sufficiently long period).

It is important to remark that the overhead caused by the use of memory is negligible: the memory storage and access can be executed through the well-known *hashing* techniques with an average CPU time per iteration that is approximately *constant*.

3. The TOTEM chip

The digital data stream SIMD computational structure was used as the paradigm for the development of two architectures [9] specially tailored for the RTS algorithm,

based on the concepts of simplicity in the basic processor structures, operation on integer numbers, low number of bits in the representation of the inputs and of the weights, and balanced processing versus I/O throughputs. The digital realization is particularly attractive in the present case because word widths of the operands are limited. A bit-parallel architecture was integrated in TOTEM, a full-custom test chip designed to operate as a co-processor in a host system, carrying out the most compute-intensive operations for RTS. The chip comprises an array of 32 parallel processors with associated on-chip weight memory and control logic with broadcast and output buses (Figure 1). Word widths in the architecture were optimized for learning with RTS: the 16-bit width of the broadcast bus is adequate to represent signals from transducers and intermediate results between layers, the memory word width of $B_w = 8$ is sufficient for many classification tasks and allows economical chip layout, while the 32-word width of the output channel permits high accumulation capacity.

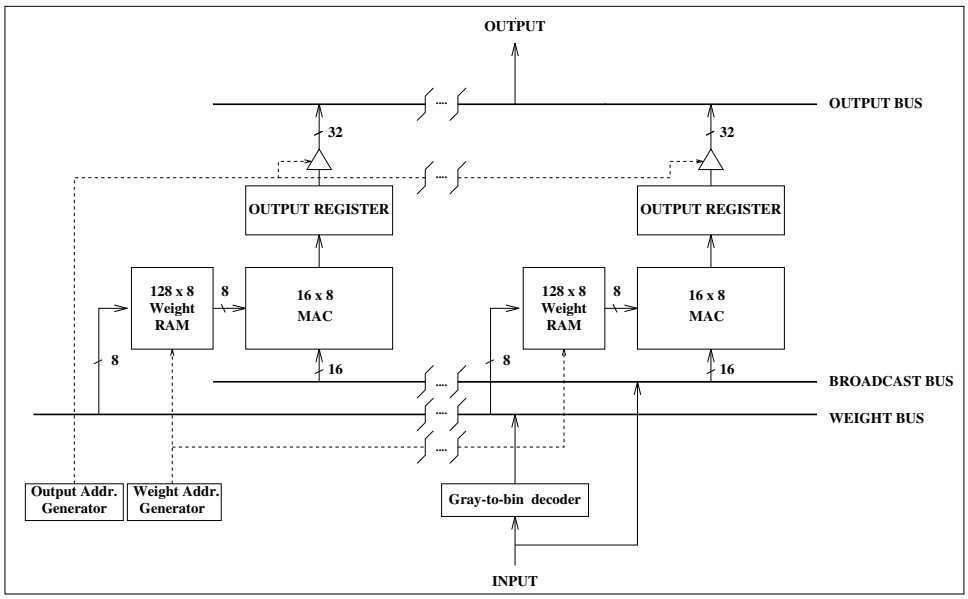

Fig. 1. Simplified block diagram of TOTEM chip.

The key design constraints of very compact memory and multiplier cells was satisfied by full-custom design of these elements, and by careful physical matching of the form factors of the memory and MAC blocks. The MACs operate on two's complement signed integers. The 16 x 8 parallel multiplier uses a Baugh-Wooley algorithm [10], which provides a more regular layout than the widely used Booth multiplier. The basic multiplier cell employs fully static 28-transistor CMOS adders. For high-speed operation, one level of pipelining is included in the multiplier array. Two additional levels of pipelining are included in the accumulator based on a

ripple-carry adder. The larger delay with respect to carry-lookahead or carry-save configurations is counterbalanced by the pipelining. A 32-bit static storage register on the output of the MAC permits an output transfer from a neuron to be performed concurrently with a parallel input-multiply-accumulate operation on all processors for optimal implementation of MLP nets.

For speed optimization reasons, the weight memory was partitioned into 1-Kbit blocks closely coupled to their related processors. RTS requires random write-access for fast modification of single weights during training, and sequential read-access during network evaluation. Shift registers can be more compact than RAMs as they avoid the significant area overhead of decoder circuits [11]. In the present architecture, however, they would impose an unacceptable delay in the write phase. Therefore random-access memory was used, and decoder area minimized by sharing the row decoders among a stacked array of memory blocks. Limited RAM area was achieved by the use of three-transistor, n-MOS dynamic cells, with separate data input, precharged data output, read and write strobe lines, which require very simple static sense amplifiers. A write-after-read scheme ensures transparent refresh in a single clock cycle during operation for uninterrupted processing of long input strings. A one-level pipeline register between RAM and MAC allows the data to be presented to the MAC during the entire clock cycle for optimum speed performance. The memory depth of 128 8-bit words allows neurons with up to 128 inputs to be implemented. Because of the sequential access to the weights, the chip can realize different MLP topologies with a high degree of flexibility: the memory bank can either be assigned to a single neuron or be partitioned among neurons on different layers. The sigmoid function is implemented off-chip (by a RAM-based look-up table). Up to four chips can be paralleled in each layer of the network to achieve high balance between number of inputs and neurons.

The chip was fabricated in a 1.2 μm CMOS process. The distribution of limited-size memory blocks physically adjacent to the related MACs boosts access performance to the 5 ns range. Measured cycle time is under 30 ns, for a sustained performance of 1 Giga multiply-accumulate operations/s and 4-cycle latency. Areas are 220 μm^2 for the RAM cell, 0.48 mm^2 for the multiplier, 0.78 mm^2 for the MAC (including interconnections) and only 70 mm^2 for the chip, with about 30% of the core chip area taken up by the memory. The total transistor count is close to 250,000.

Table 1. Characteristics of the parallel processor chip.

Number of processors	32	Total on-chip RAM	32 Kbit
Cycle time	30 ns	Performance	1 GMACs
Technology	1.2 μm CMOS	Die size	70 mm^2
Transistor count	250.000	Package	132-pin CPGA

The layout of the complete chip is shown in Figure 2 and its main characteristics summarized in Table 1. A doubling in the processor density is expected by implementing the circuit in a 0.8 μm process.

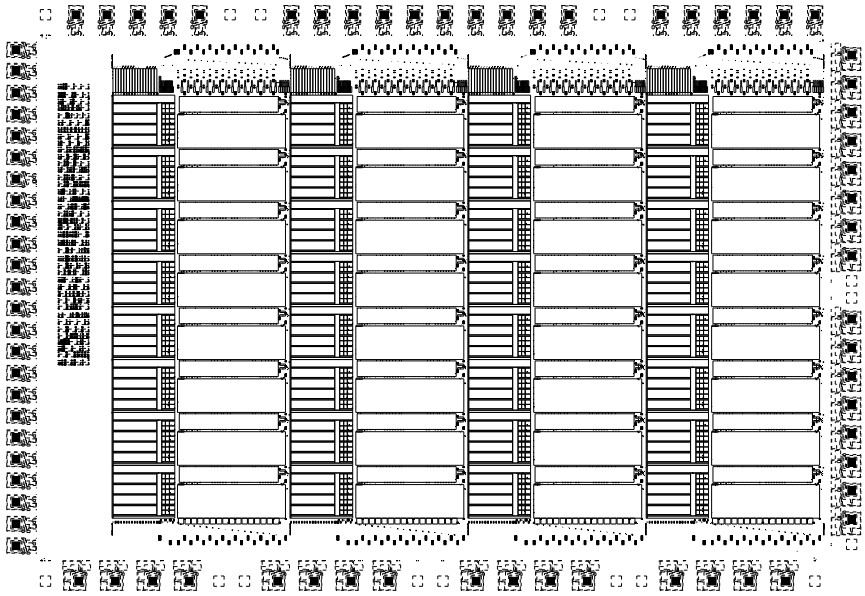

Fig. 2. Layout of TOTEM chip: view of 32 processors, each containing the RAM cells (left), decoders (center), the MAC (bottom-right) and the output register (top-right). The control block is at the left of the layout.

4. A selected application in HEP

Time-critical applications of pattern recognition with neural nets are being considered for High Energy Physics (HEP) experiments [1] [2]. The support of the INFN (Italian National Institute for Nuclear Physics) in the development of the TOTEM chip was motivated by its possible use in the "triggering" step, and the suitability of the chip for HEP applications was thoroughly investigated. In this section we present a brief comparison of the results that can be obtained with a net with real-valued weights (represented with double-precision floating-point numbers) trained with the One Step Secant (OSS) algorithm [5], and networks with limited-size integer weights (from 4 to 8 bits) trained with the RTS algorithm. The task is that of identifying the relevant events (with a "C3 vertex") and separating them from a background of uninteresting events [2].

Event samples used for the training were derived from the WA92 [2] production run and analyzed with an event-reconstruction program. A total of 3,112 C3 events and 33,760 background events were divided into two equal-size sets, one used for training, the other one for testing (validation). The network is a feed-forward MLP with 16 input, 16 hidden, and one output unit. The target output for the background events is zero, that for the relevant C3 events is 1 for the real-valued net, $2^{B_w} - 1$ for the RTS network (where B_w is the number of bits per weight).

The on-line operational tests [2] give an enrichment of about 6.5 for an acceptance of 0.18. Because a different test set is used for our tests (the original on-line data

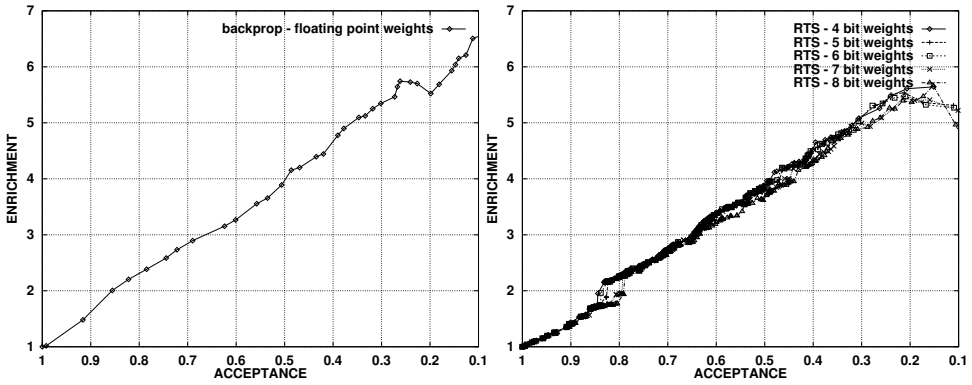

Fig. 3. C3 event enrichment versus acceptance. Network with real-valued weights trained with the OSS method (left) and network with integer-valued weights trained with RTS (right).

were not saved), to obtain a "fair" comparison, a real-valued net was trained with the OSS optimization technique for a number of iterations corresponding to the maximum generalization results (200 batch iterations). The C3 event enrichment versus acceptance obtained when varying the cutoff on the neural trigger output, for the WA92 1993 data, is shown in Figure 3 (left), where the test data are fed through the network.

Figure 3 (right) shows the enrichment-efficiency plot obtained on the test set at 400 RTS iterations, for different numbers of bits per weight. One can observe that 4 bits are sufficient to approximately reproduce the best test results in the operating range of interest. A larger number of bits does not produce significantly better results. The real-valued net performs better at very low acceptance rates, although in that range the comparison is not very significant (the statistical error becomes large for low acceptance values: at 0.1 acceptance, only about 150 of the C3 test patterns survive).

It is useful to report the performance of two existing systems on this task. The 16-16-1 topology MLP considered in [2] using two commercial neural processors is characterized by a total response time of 5.8 μs for the ETANN chip (to complete a network evaluation), and about 5 μs for the MA16 chip [12]. The analog chip requires a high environmental stability: in [2] the supply voltage is stabilized within 5 mV and the temperature within 1^0 C. In addition, the training phase is laborious especially if the functionality has to remain stable in time.

Acknowledgements

The work was partially supported by INFN, the Special Project of the University of Trento and by EU Esprit Project 7101 MInOSS.

1. B. Denby. (1993) The Use of Neural Networks in High-Energy Physics. *Neural Computation* **4**(5):505–549.

2. C. Baldanza, F. Bisi, A. Cotta-Ramusino, I. D'Antone, L. Malferrari, P. Mazzanti, F. Odorici, R. Odorico, M. Zuffa and WA92 Collaboration. (1994) Results from an on-line neural trigger within a fixed target experiment for the production of beauty particles. *Proceeding of the III Int. Workshop on Software Engineering, Artificial Intelligence and Expert Systems for High Energy and Nuclear Physics, October 1993 Oberammergau*, pp. 391–409, World Scientific, Singapore.
3. T. Nordström and B. Svensson. (1992) Using and Designing Massively Parallel Computers for Artificial Neural Networks. *Journal of Parallel and Distributed Computing* **14**(3):260–285.
4. S. Saarinen, R. Bramley and G. Cybenko. (1993) The numerical solution of neural network training problems SIAM Journal of Statistical and Scientific Computing, in press.
5. R. Battiti and G. Tecchiolli. (1994) Learning with first, second, and no derivatives: A case study in high energy physics. *Neurocomputing* **6**: 181–206.
6. R. Battiti and G. Tecchiolli. (1992) The Reactive Tabu Search. *ORSA Journal on Computing*, Vol. 6, N. 2 (1994), pp. 126-140.
7. F. Glover. (1989) Tabu Search - part I. *ORSA Journal on Computing* **1**(3):190–206.
8. R. Battiti and G. Tecchiolli. (1993) Training Neural Nets with the Reactive Tabu Search. Tech. Rep. UTM 421, Dip. di Matematica, Univ. di Trento - Italy. *IEEE Transactions on Neural Networks*, in press.
9. R. Battiti, P. Lee, A. Sartori, G. Tecchiolli, "TOTEM: a Digital Processor For Neural Networks and Reactive Tabu Search," in: *Proc. of the Fourth International Conference on Microelectronics for Neural Networks and Fuzzy Systems, MICRONEURO 94*, pages 17–25, Torino, IT, 1994. IEEE Computer Society Press.
10. C. R. Baugh and B. A. Wooley. (1973) A two's Complement Parallel Array Multiplication Algorithm. *IEEE Transactions on Computers* **C-22** (12):1045-1047.
11. M. Pelgrom and H. Termeer. (1986) A 32 KBit Variable Length Shift Register for Digital Audio Application. *Proc. ESSCIRC-86 Delft*:38–40
12. U. Ramacher. (1992) SYNAPSE – A Neurocomputer That Synthesizes Neural Algorithms on a Parallel Systolic Engine *Journal of Parallel and Distributed Computing* **14**(3):306–318