When will general purpose micro-processors simulate neural networks in real time for HEP applications ?

Bertrand Granado Lionel Lacassagne
Patrick Garda
Université Pierre et Marie Curie - Laboratoire des Instruments et Systèmes
Boîte 252 - 4, place Jussieu - 75252 Paris Cedex 05
Tel.: (33) 01 44.27.75.07 - Fax.: (33) 01 44.27.75.09
email:Bertrand.Granado@lis.jussieu.fr

1 Introduction

By their universal character, general purpose micro-processors may be used to simulate artificial neural networks. However, until now, they were not capable to perform these simulations in real-time. On the other hand, the computational power of these processors has tremendously increased recently. Thus, one may wonder whether up-to-date general purpose micro processors can simulate neural networks in real-time.

To answer this question, we need to evaluate the performances of these architectures for the simulation of neural networks.

To realize this evaluation we have developed an original methodology [7] which can predict the simulation time of a neural network on an electronic architecture. This prediction is based on an analytic model of the architecture performances.

2 Neural Nets models

In this article we consider the two most used kind of neural networks, which are the Multi-Layer Perceptrons (MLP) and the Radial Basis Function networks (RBF).

To determine if general purpose micro-processor can perform real-time simulation of artificial neural networks, we simulated two neural nets: a MLP called MLPPHYS, a RBF called RBFPHYSMANH and a RBF called RBFPHYSMAHA.

- MLPPHYS is a Multi-Layer Perceptron with 3 layers, it's topology is 64x64x1. It was designed for physics experiments and described in [10]. The simulation time of this network has to be less than 8μs.
- RBFPHYSMANH [7, 6] is a Radial Basis Function network. Its 3 layers include respectively 8, 8 and 1 neurons, and it uses the Manhattan distance. this network was designed for physics experiments and described in [9].
- RBFPHYSMANH [7, 6] is a Radial Basis Function network. Its 3 layers include respectively 8, 8 and 1 neurons. This network is similar to RBFPHYSMANH, the difference is that it uses the Mahalanobis distance. This distance may increase drastically the classification rate of the network [5].

3 Evaluation

To determine the interest of general purpose micro-processors for the real-time simulation of neural networks, we have developed an original methodology for the evaluation and prediction of the processors performances [7].

3.1 Method

The usual method to predict the simulation time of neural networks on an electronic architecture is based on the measure of an average speed S for the connections processing. Then the simulation time of a MLP or a RBF with C connections is simply taken as $S * C$. We have demonstrated in [8] that this method cannot be applied for a general neural network architecture because it leads to very high predictions errors. Thus we introduced a new method for this prediction.

3.1.1 Description

This methodology is based on the extraction of an analytical model for the computational primitives of the neural network model. These primitives are the basic mathematical operations that define the model.

The extracted analytical model is a mathematical function that provides the simulation time of a neural network depending on some neural network parameters like the number of neurons or the kind of connections (local or full). It also depends on some hardware parameters like the cache size or the clock frequency.

To get the total simulation time of the a neural network, we simply accumulate the simulation times given by the analytical model for all the primitives.

3.1.2 Primitives for MLP

The equations 1 and 2 give the primitives associated to the MLP model.

$$g\left(X_{j(j\in E_i)}\right) = \sum_{j\in E_i} X_j * W_{ij} \quad (1)$$

$$f(V_i) = m\frac{1-e^{-\lambda V_i}}{1+e^{-\lambda V_i}} \quad (2)$$

m determines the range of the neuron state, included in $[-1:1]$, and λ is the slope of f.

3.1.3 Primitives for RBF

The equations 3 and 4 give the primitives associated to the RBF model, when the Mahalanobis distance is used.

$$g\left(X_{j(j\in E_i)}\right) = \sum_{j\in E_i} (W_{ji}-X_j)^t \Sigma_i^{-1}(W_{ji}-X_j) \quad (3)$$

$$f(V_i) = e^{\frac{-V_i}{\lambda}} \quad (4)$$

λ characterizes the influence zone of the neuron. Σ_i^{-1} is the inverse of the covariance matrix associated to the neuron i.

3.1.4 Number of primitives

In a general purpose micro-processor, there are two major kinds of computation units, an integer unit and a floating-point unit. Thus we have evaluated these two units and we have programmed the four primitives described above in an integer and in a floating-point version: this leads to eight primitives.

3.2 Analytical models for a general purpose micro-processor

To determine the execution time of a primitive P, we first determine the total NumBer of Instructions needed to simulate this primitive $NBI_P(k,l,\ldots,m,n)$, as a function of the sizes $(k,l,\ldots,m,n)$ of the layers $(w,x,\ldots,y,z)$. This step can be realized by an analysis of the assembler code of the programmed primitive.

Thanks to this function, we can estimate the number of Cycles Per Instruction CPI_P for this primitive, with the formula:

$$CPI_P(k,l,\ldots,m,n) = \frac{T*F}{NBI_P(k,l,\ldots,m,n)} \quad (5)$$

where F is the CPU frequency and T is the simulation time measured for this primitive. To approximate the CPI_P, we have made numerous simulations of the primitives, measured the simulation time and determined the CPI_P with the formula 5.

At this point we have two functions: a function NBI_P which provides the number of instuctions for the primitive executed as a function of the sizes of the layers of a neural network, and a function CPI_P which provides the number of cycles per instruction for the primitives as a function of the sizes of the layers of a neural network.

Let us take now a neural network characterized by the primitives $p \in \Pi$ and by the layers $(a_p,b_p,\ldots,c_p,d_p)$ for the primitive p. We can compute the simulation time TS of this neural network with the formula:

$$TS = 1/F * \sum_{p\in\Pi}(NBI_p(a_p,b_p,\ldots,c_p,d_p) * CPI_p(a_p,b_p,\ldots,c_p,d_p)) \quad (6)$$

With the equation 7, we can predict the simulation time of any neural network without programming it on the architecture. Moreover this analytical model depends on the size of the layers but also on the parameters of the architecture like the clock frequency, the cache size, etc .... Then if we change the value of a parameter, for example the clock frequency, we can compute the simulation time of a neural network on a new architecture which is a minor modification of the architecture originaly evaluated. Thus we can forecast now the performances of a processor which will be introduced in the future.

But it's hard to give a deterministic analytical model for the architecture of a general purpose micro-processor, because it includes complex mechanisms. Such mechanisms are:

- **Memory management** including two or three memory cache levels.
- **Instruction flow sequencing mechanism** with branch prediction.
- **Out of order execution of the instructions**.

These mechanisms introduce non deterministic execution times of the instructions flow, because they depend on the values and the nature of the data. The consequences of these features are that the estimation of the CPI_p given by equation 5 show a very large dispersion.

To overcome this problem we estimate the range of CPI_p thanks to two extrema, CPI_p^{min} and CPI_p^{max}. These two values are defined such as for any network:

$$CPI_p^{min} \leq CPI_p(k,l,\ldots,m,n) \leq CPI_p^{max}$$

With these two extrema, our methodology gives two predicted times, a maximum predicted time and a minimum predicted time. Then if the maximum predicted time is smaller than the real time constraint we can say that the neural network is simulated in real-time.

4 Evaluation of SPARC and X86 family processors

To determine the analytic models of the SPARC and X86 processors, we used two commercial C language compilers: Sun Microsystem CC-4.2 compiler for SPARC and Microsoft Visual C++ 5 for X86.

4.1 The processors : ULTRASPARCII

Firstly we evaluated a processor of the SPARC family: the ULTRASPARCII.

4.1.1 Hardware

In this section, we describe the hardware architecture of the evaluated processors. These descriptions are derived from [11, 2, 1].

The SPARC [1] architecture is derived from the Berkeley university studies between 1984 and 1987. It's a RISC architecture owned by Sun microsystems. The evaluated processor characteristics are:

- ULTRASPARCII complies to the SPARC V9 norm. It is a four degree superscalar processor. It has one integer unit with two ALU, one floating-point and VIS [2] graphic unit with 5 processing units, one memory management unit, a 16 KB L1 instructions cache and 16 KB of L1 data cache. It has a L2 cache, its size is in the range [512 KB ,16 MB]. Its clock frequency is 250 Mhz, it has 3.8 millions of transistors in a 0,29 μm CMOS technology.

4.2 The processors X86

The X86 processors family is derived from an Intel seventies CISC architecture. But to compete with other micro-processors in scientific applications, there is with PENTIUM micro-processors an evolution towards a RISC internal micro-architecture.

4.2.1 Hardware

- PENTIUM II is a CISC-RISC micro-processor, the first stage of the pipeline is dedicated to translate CISC instructions into 118 bits RISC-like micro-instructions. This micro-processor has an integer unit with two ALU, a floating-point unit, a memory management unit, 16 KB of L1 instructions cache and 16 KB of L1 data cache. There are MMX [3] graphic units, the L2 cache is running at 2/3 of the CPU clock with and its size is not limited to 512 KB. There are 7.5 millions of transistors in a 0,28 μm CMOS technology and a CPU clock of 266 MHz.

[1] Scalable Processor ARChitecture
[2] Visual Instructions Set
[3] MultiMedia eXtension

4.3 Analytical models

We extracted the analytical models for the eight primitives and for the four processors. We cannot give in this article all the models, but we give the example of the PENTIUMII processor for the interger Mahalanobis distance primitive in table 1. The range of CPI_{mahai} is:
$CPI^{min}_{mahai} = 1.1311$ and $CPI^{max}_{mahai} = 3.5772$.
The function h is defined as:

$$h(x) = 1 \; if \; x > 0 \; else \; h(x) = 0$$

Primitives	Analytical model
Mahalanobis Distance Integer version for CPI^{min}_{mahai}	$(1.1311/F) * (39 + h(size] * (9 + 12 * size) + h(size - 3) * (11 + \lfloor \frac{size}{4} \rfloor * (9 + 19 * size)) + h(size\%4)) * (7 + (size\%4) * (8 + 9 * size)))$
Mahalanobis Distance Integer version for CPI^{max}_{mahai}	$(3.5772/F) * (39 + h(size] * (9 + 12 * size) + h(size - 3) * (11 + \lfloor \frac{size}{4} \rfloor * (9 + 19 * size)) + h(size\%4)) * (7 + (size\%4) * (8 + 9 * size)))$

Table 1: Example of PENTIUMII analytical models, where F is the clock frequency

With all the analytical models we can perform both evaluation and prediction.

4.4 Evalution and Prediction

We present here the predicted and measured simulation time of the three neural networks, MLPPHYS, RBFPHYSMANH and RBFPHYSMAHA.

4.4.1 SPARC family

Processor	Neural Network	Minimum Predicted Time (in μs)	Maximum Predicted Time (in μs)
ULTRASPARCII	MLPPHYS integer	124.74	354.65
	MLPPHYS float	103.44	241.04
	RBFPHYSMANH integer	2.29	5.89
	RBFPHYSMANH float	2.58	4.87
	RBFPHYSMAHA integer	24.07	51.82
	RBFPHYSMAHA float	18.00	38.14

Table 2: Predicted simulation time for MLPPHYS, RBFPHYSMANH and RBFPHYSMAHA on ULTRASPARCII processor at 250 MHz

The table 2 shows that minimum predicted times are one order larger than the required 8μs latency for the MLPPHYS network in its two versions. But for the RBFPHYSMANH network the required 8μs latency is guaranteed and for the RBFPHYSMAHA there is only a factor of 2 between the required 8μs latency and the simulation time of the network.

4.4.2 X86 family

Processor	Neural Network	Minimum Predicted Time (in μs)	Maximum Predicted Time (in μs)
PentiumII	Mlpphys integer	62.46	187.35
	Mlpphys float	61.82	290.19
	Rbfphysmanh integer	1.48	5.59
	Rbfphysmanh float	2.83	6.97
	Rbfphysmaha integer	10.26	37.84
	Rbfphysmaha float	11.74	39.58

Table 3: Predicted simulation time on PentiumII at 266 MHz

Similarly to the Sparc family, the table 3 shows that Mlpphys and Rbfphysmaha networks cannot be simulated in 8μs by the current Intel micro-processors, but Rbfphysmanh can be.

5 Predicted performances for future electronic architectures

Our methodology can evaluate actual electronic architectures, but it can also predict the simulation time of future evolutions of these architectures. We used it to predict the simulation times of the neural networks Mlpphys, Rbfphysmanh and Rbfphysmaha on four possible future evolutions of the UltraSparcII and PentiumII. For the sake of simplicity, we modified only a single parameter: the clock frequency. The prediction will be pessimistic, because progress in microelectronics technology may lead to a speedup larger than the ratio of the clock frequencies as we saw when we compared the SuperSparc and the UltraSparc.

The four evolutions for which we predict the simulation time of Mlpphys, Rbfphysmanh and Rbfphysmaha networks are:

- an UltraSparcII with a 400 MHz clock frequency,
- an UltraSparcII with a 1 GHz clock frequency,
- a PentiumII with a 400 MHz clock frequency,
- a PentiumII with a 1 GHz clock frequency.

The clock frequency of 400 MHz is up-to-date as the current generation of PentiumII have a frequency of 450 MHz, and the UltraSparcIIi a frequency of 360 MHz.

The 1 GHz frequency will be available before year 2002. This is not a dream, as said Peter Bannon of Compaq at the MicroProcessor Forum on October 1, 1998. The Alpha EV7 micro-processor, the next generation of Alpha processors will be operated at more than 1 GHz [4]. Sun announces in its roadmap [3] a new generation of UltraSparc processor with a frequency of 1.5 GHz in 2002.

The prediction results are shown in table 4.

Neural Network	UltraSparcII 400 Mhz Minimum Time (in μs)	UltraSparcII 400 Mhz Maximum Time (in μs)	PentiumII 400 Mhz Minimum Time (in μs)	PentiumII 400 Mhz Maximum Time (in μs)
MLPphys integer	77.97	221.66	41.54	124.59
MLPphys float	64.65	150.65	41.11	192.98
Rbfphysmanh integer	1.43	3.68	0.98	3.72
Rbfphysmanh float	1.61	3.05	1.89	4.64
Rbfphysmaha integer	15.04	32.39	6.83	25.17
Rbfphysmaha float	11.25	23.84	7.81	27.33

Table 4: Predicted time for UltraSparcII and PentiumII with 400 MHz clock frequency

The table 4 shows that the time simulation of Rbfphysmaha will be very close to the 8μs required latency, and that the simulation time of Rbfphysmanh will be lower than 5μs.

Neural Network	UltraSparcII 1 Ghz Minimum Time (in μs)	UltraSparcII 1 Ghz Maximum Time (in μs)	PentiumII 1 Ghz Minimum Time (in μs)	PentiumII 1 Ghz Maximum Time (in μs)
MLPphys integer	31.19	88.67	16.62	49.84
MLPphys float	25.86	60.26	16.45	77.19
Rbfphysmanh integer	0.58	1.48	0.40	1.49
Rbfphysmanh float	0.65	1.22	0.76	1.86
Rbfphysmaha integer	6.02	12.96	2.73	10.07
Rbfphysmaha float	4.5	9.54	3.13	10.53

Table 5: Predicted time for UltraSparcII and PentiumII with 1 GHz clock frequencies

The table 5 shows that with 1 GHz clock frequencies, simulations of the Rbfphysmaha network could take place in less than 11μs but the simulation of the Mlpphys network could not: so specialized hardware will be needed to stand a 8μs simulation time latency.

6 Conclusion

In this article we propose a new methodology to evaluate and predict the simulation time of MLP and RBF neural networks on general purpose micro-processors.

With this methodology we evaluated two processors family, Sparc and X86 and we demonstrated that the general purpose micro-processors can not now simulate MultiLayer Perceptrons with a 8μs real time constraint.

We used also our methodology to predict the simulation time of neural networks on two future possible evolutions of Sparc and X86 family, and we showed that these architectures would simulate Radial Basis Function networks with Mahalanobis distance in real time with a 8μs time constraint. They could be available in the next three years. But these architectures would not simulate Multi-Layer Perceptron in real time with a 8μs constraint.

References

[1] Ultrasparc user's manual - ultrai - ultraii. Technical report, Sun Microsystems. http://www.sun.com/microelectronics/manual/ultrasparc/802-7220-02.pdf.

[2] Intel architecture software developer's manual, volume 1: Basic architecture. Technical report, Intel Corporation, 1997. http://developer.intel.com/design/pentium/manuals/24319001.pdf.

[3] 1999. http://www.sun.com/microelectronics/roadmap/.

[4] Peter Bannon. Alpha 21364: A scalable single-chip smp. Compaq Computer Corporation, Shrewsbury, MA, October 1998.

[5] Larbi Beheim and Maurice Milgram. Comparison between different distances in character recognition. In *RFIA'98*, pages 229 – 232, Clermont-Ferrand - France, January 1998.

[6] R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis*. John Wiley, New York, United States, 1973.

[7] Bertrand Granado. *Architecture des systèmes électroniques pour les réseaux de neurones - Conception d'une rétine connexionniste*. PhD thesis, Université Paris XI, November 1998.

[8] Bertrand Granado and Patrick Garda. Evaluation of cnaps neuro-computer for the simulation of mlps with recptive fields. In *Proceedings of IWANN'97*, Lanzarote - Canary Islands, Spain, June 1997.

[9] Clark S. Lindsey, Thomas Lindblad, Givi Sekhniaidze, G. Székely, and M. Minersklöld. Experience with the ibm zisc036 neural network chip. In B. Denby and D. Perret-Gallix, editors, *New Computing Techniques In Physics Research IV*, pages 371 – 376, Pisa, Italy, April 1995. World Scientific.

[10] J. Möck, J. Fent, W. Fröchtenicht, F. Gaede, A. Gruber, J. Huber, C. Kiesling, T. Kobler, J. Köhne, P. Ribarics, S. Udluft, D. Westner, T. Zobel, H. Getta, D. Goldner, M. Kolander, T. Krämerkämper, and H. Kolanoski. Artificial neural networks as a second-level trigger at the h1 experiment - performance analysis and results. In B. Denby and D. Perret-Gallix, editors, *New Computing Techniques In Physics Research IV*, pages 465 – 471, Pisa, Italy, April 1995. World Scientific.

[11] André Seznec and Thierry Lafage. Evolution des gammes de processeurs mips, dec alpha, powerpc, sparc, x86 et pa-risc. Technical Report 1110, Institut de Recherche en Informatique et Systè mes Aléatoires, 1996.