Can general purpose micro-processors simulate neural networks in real-time ?

Bertrand Granado, Lionel Lacassagne, and 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 [6] 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 Real time

Neural networks are often used in real-time applications. Such applications are for example the recognition of the amount of a bank check or of a zip postal code. In these applications the simulation time is hard limited. In this article we have taken a time constraint of 40 ms, which correspond to the CCIR video rate.

3 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 LeNet and a RBF called Rbf3.

3.1 LeNet

LeNet is a TDNN Multi-Layer Perceptron with 96522 local connections, 1920 full connections and 4365 neurons. Its function is to recognize handwritten digits. It was developed by Y. Lecun in the AT&T laboratories [8].

3.2 Rbf3

Rbf3 [6, 5] is a Radial Basis Function network. Its 3 layers include respectively 256, 10 and 4 neurons, and it uses the Mahalanobis distance. This distance is a very hard benchmark because the number of computations increases as the square of the number of neurons in the input layer.

4 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 [6].

4.1 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 [7] 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.

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.

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

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

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

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

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

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

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

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

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.

4.2 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:

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

With the equation 6, 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.

5 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.

5.1 The processors : SuperSparc, UltraSparcII

Firstly we evaluated two processors of the Sparc family: the SuperSparc and the UltraSparcII.

Hardware In this section, we describe the hardware architecture of the evaluated processors. These descriptions are derived from [9, 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 two evaluated processors characteristics are:

- SuperSparc complies to the Sparc V8 norm. It is a three degree superscalar processor. It has one integer unit with two ALU, one floating-point unit, one memory management unit, a 16 KB L1 data cache and a 20 KB L1 instructions cache. Its clock frequency is 50 Mhz, it has 3.1 millions of transistors in a BiCMOS 0,6 μm technology.
- 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.

5.2 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.

Hardware

- PentiumPro 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, 8 KB of L1 instructions cache and 8 KB of L1 data cache. There is, at the same CPU clock frequency, a 256 or 512 L2 unified cache. CPU clock is 200 MHz.
- Pentium II is an improvement of the PentiumPro. There are MMX [3] graphic units, the sizes of L1 caches are increased up to 16 KB, the L2 cache is only 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

5.3 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.

5.4 Evalution and Prediction

We present here the predicted and measured simulation time of the two neural networks, LeNet and Rbf3.

Sparc **family** The table 2 shows that measured times are smaller than maximum predicted time and larger than minimum predicted time: this confirms the validity of our methodology.

For the real time simulation of the neural networks, this table shows that the SuperSparc processor can not satisfy the 40 ms time constraint.

But on the other hand, the UltraSparcII can manage the real time simulation of LeNet. We have a maximum time of 8.3 ms for the integer version and a maximum time of 14.621 ms for the floating-point version. Because LeNet is one of the biggest MLP ever designed, we can state that current MLPs can be simulated in real-time on general purpose micro-processors, when the time constraint is 40 ms.

However, the table 2 shows that the real-time simulation of Rbf3 cannot always be achieved

Processor	Neural Network	Measured Time (in ms)	Minimum Predicted Time (in ms)	Maximum Predicted Time (in ms)
SuperSparc	Lenet integer	37,424	22,939	46,005
	Lenet float	51,199	24,465	56,593
	rbf3 integer	230,697	144,903	259,369
	rbf3 float	211,703	190,944	255,853
UltraSparcII	Lenet integer	4,578	2,728	8,359
	Lenet float	11,709	7,380	14,621
	rbf3 integer	43,206	30,500	65,395
	rbf3 float	37,619	21,821	46,244

Table 2. Predicted and mesured simulation time for LeNet and Rbf3 on SuperSparc and UltraSparcII processors

The results shown in table 2, demonstrate the impressive evolution of general purpose micro-processors. The SuperSparc, introduced in 1992, is seven times less powerful than the UltraSparcII introduced in 1997. This evolution is not only a consequence of the increase of the clock frequency, as the ratio between the two clock frequencies is only equal to five, but also a consequence of architecture improvements like memory cache management or duplication of computional units.

If this evolution continues, the integer version of the Rbf3 network could be simulated in 9.34 ms in year 2002 on a Sparc processor which would be 7 times more powerfull than the UltraSparcII. Then general purpose micro-processors could be used for the real-time simulation of RBF with the Mahalanobis distance

Processor	Neural Network	Measured Time (in ms)	Minimum Predicted Time (in ms)	Maximum Predicted Time (in ms)
PentiumPro	Lenet integer	3,019	2,751	8,086
	Lenet float	37,869	10,853	41,523
	rbf3 integer	51,404	17,816	56,346
	rbf3 float	54,094	20,886	75,583
PentiumII	Lenet integer	2,134	2,113	21,252
	Lenet float	24,378	7,933	39,046
	rbf3 integer	42,800	13,033	48,442
	rbf3 float	43,238	16,149	54,198

Table 3. Predicted and measured time on PentiumPro et PentiumII

X86 **family** Similarly to the SPARC family, the table 3 shows that our methodology is valid, and that MLP, can be simulated in real time on these architectures.

6 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 time of the neural networks LENET and RBF3 on four possible future evolutions of the ULTRASPARCII and PENTIUMII. For the sake of simplicity, we modified only a parameter: the clock frequency. The prediction will be pessimistic, because progress in microelectronics technology may lead to speedup larger than the ratio of the clock frequency as we saw when we compared the SUPERSPARC and the ULTRASPARC.

The four evolutions for which we predict the simulation time of LENET and RBF3 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 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 operates at more than 1 GHz [4]. Sun announces is in 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 Maximum Time (in ms)	PentiumII 400 Mhz Maximum Time (in ms)	UltraSparcIII 1 Ghz Maximum Time (in ms)	PentiumII 1 Ghz Maximum Time (in ms)
LeNet float	9,138	25,965	3,655	10,386
Rbf3 integer	40,871	32,214	16,348	12,885
Rbf3 float	28,902	36,042	11,561	14,416

Table 4. Predicted time for ULTRASPARCII and PENTIUMII with 400 MHz and 1 GHz clock frequencies

This table shows that with 400 MHz and 1 GHz clock frequencies, simulations of neural networks will possible in real time for the two kind of neural networks when the time constraint is 40 ms.

7 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 now simulate MultiLayer Perceptrons with a 40 ms 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 40 ms time constraint. They could be available in the next three years.

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. R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis*. John Wiley, New York, United States, 1973.
6. Bertrand Granado. *Architecture des systèmes électroniques pour les réseaux de neurones - Conception d'une rétine connexionniste*. PhD thesis, Université Paris XI, November 1998.
7. 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.
8. Y. LeCun, B. Boser, J.S. Denker, D.henderson, R.E. Howard, W. hubbard, and L.J. Jackel. Handwritten digit recognition with a back-propagation network. In *Neural Information Process and System*, pages 396–404, 1990.
9. 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.