Implementing very high-speed hierarchical MLP-based classification systems in real-time industrial environments

Riccardo Parenti (*), Carla Penno (*)
Daniela Baratta (**), Francesco Masulli (***)

(*) Ansaldo Ricerche S.r.l., Corso F.M. Perrone, 25, 16161- Genova (IT) - E-mail: parenti@ari.ansaldo.it
(**) DIBE Università di Genova, via all'Opera Pia, 11A, 16145 – Genova (IT) – E-mail: dany@dibe.unige.it
(***) DISI Università di Genova, via Dodecaneso, 33, 16146 – Genova (IT) – E-mail: masulli@ge.infm.it

Abstract

Using a proper combination of special HW and SW, it is possible to exploit the capabilities of the Multi-Layer-Perceptron-based "tree architecture" even in very high-speed industrial classification problems. In particular, the paper shows as, adopting a specially developed 128 MCPS ASIC Neural co-processor (able to synthesize a whole 64 input – 128 hidden – 64 output MLP on-chip), it has been possible to build an industrial board, suitable for both PCI and VME bus, capable of more than 50000 highly detailed pattern classification per second. The board aims to become a widely diffused tool for the industrial implementation of data treatment systems. It is now entering in the commercialization phase. The paper gives many details about the chip + board + SW kit developed and shows an example of application.

Introduction

It is widely known that Neural Networks (NN) constitute a cost-efficient solution for classification problems [1][2][3][4]. It is also assessed that they can handle a large class of classification problems yielding the reliability needed in the industrial environment [7], even if the availability of industrial-graded development environments is today not so diffused.

A typical industrial-graded development kit should:

(i) be able to achieve the performance typically required in industrial applications;
(ii) be easy to be operated, both during the set-up and the installation phase;
(iii) be equipped with tools useful to validate the system performances both in speed and classification quality;
(iv) be implemented in such a way to be easily integrated in the industrial data treatment systems, e.g. by means of a standard industrial bus interface.

Of course, in order to be able to assure a good commercial exploitation, a really friendly human interface of the development kit has to be developed, in order to allow the user to apply the NN-based classification system, also without a specific NN skill. In particular, the SW development toolkit should hide as much as possible to the industrial operator the complexity related to the set-up and training of the NN system.

Many of the problems of coupling advanced data treatment techniques to real industrial applications are quite complex; they have been here solved thanks to a deep experience matured by some of the authors in the implementation of Advanced Information Technologies in the industrial field [5...14].

To meet the real-time constraints of a very high-speed application, a dedicated HW implementation has been designed and manufactured. It is based on an ASIC neural co-processor [5][6] (called MLP_chip) whose architecture has been optimized to execute in real-time a MLP-based hierarchical network.

An application board, containing two MLP_chips, some weight RAMs and a glue logic chip, has been designed and manufactured according to the IP bus ANSI standard specifications (hence compatible with many commercial carrier boards). Both a VME bus based system, and a PCI bus version, compliant with the emerging standard for low cost industrial systems, have been developed and tested.

A Windows based user interface has been implemented for the development kit, able to let the operator be in condition to obtain a fully working system simply feeding the system with a good database, collected on the plant.

The system has been designed in order to implement in HW the TMLP (Tree of Multi-Layer Perceptron) architecture (see later), a hierarchical NN topology able to yield the same performances of any MLP application, requiring at the same time very much less synapses, obtaining a very much faster training phase [6].

For instance in the sample application presented, the TMLP required about 5 times less connections than its equivalent MLP, performed about 5 times faster, and required about 25 times less training time.

The structure of this paper is as follows. Section 2 introduces the TMLP architecture and its capability.

Section 3 deals with the basic neural HW developed and gives some information related to the Windows NT driver developed for the commercial carrier board used. Section 4 describes the development toolkit, that is the SW developed in order to allow the non-specialist to manage in details the NN system. Section 5 shows a VME-based sample application developed for the steel industry. Section 6 reports conclusions and final observations.

The TMLP Architecture

Multi Layer Perceptron (MLP) based architectures have been proved, among others, reliable and they are widely adopted in the real-world applications. The TMLP architecture introduces a particular hierarchical architecture.

In order to explain more clearly the TMLP architecture let us consider the application example reported. Since such an application concerns the classification of samples belonging to hierarchically organized classes, it seemed natural to use a hierarchical network, reflecting the same hierarchy of the data to be classified. Anyway the TMLP architecture can be suitable also for problems where the hierarchical structure is not so clearly defined.

In principle, the hierarchical classification scheme, likewise any associated TMLP network, could span over many levels, yielding very complex topology.

In practice, it is extremely difficult to find real-world problems requiring more than two levels in order to be well solved. For that reason the system developed refers to a generic second order tree, composed by a maximum of a single root MLP driving a maximum of 63 leaf MLP. Each of the MLP involved (root, leaves) can have as much as 64 input, 128 hidden units, 64 output nodes.

The overall problem complexity solvable by such a rich structure is very high and, very probably, able to well cover all the main industrial applications. It can be considered that such a structure can classify as much as 8192 different objects described using a 64 features input data set.

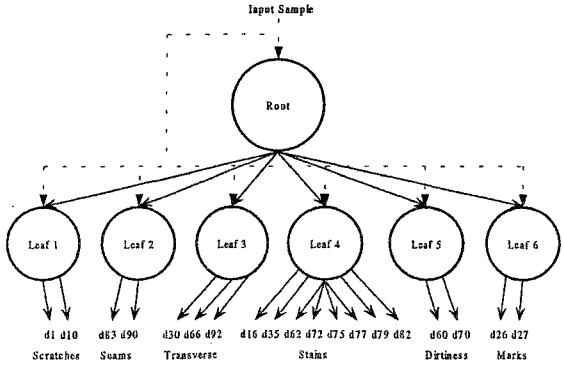

Figure 1. Structure of the TMLP network

In order to clarify the TMLP network use, the reader can look (Figure 1) at the hierarchical structure used for the sample problem reported in Section 5. Each node shown in the picture is a single-hidden-layer, multiple-output MLP. The MLP at the top level (root MLP) has to classify the input data with respect to superclasses (i.e. defect families, e. g. seams, marks, stains, etc.). It selects only one leaf MLP at the second level in charge of classifying the input sample, within the chosen family, into detailed classes (e.g. d1, d30, d92 etc.). Both the first and the second level MLPs refer to the same input data set.

Concerning training, each MLP is trained independently from the others (e.g. using the Back Propagation algorithm [1]) on the corresponding data sub-set.

Generally speaking, the main advantages of using TMLPs with respect to MLPs implementation in real-world problems are at least the following [15]:

(i) a significant reduction of the training complexity respect to MLPs;
(ii) a significant increment of the classification accuracy, thanks to the exploitation of the "inherent data structure" of the problem;
(iii) a very much easier determination of the single MLP topology (hidden neurons required, overall number of weights, etc.).

The Basic Neural HW Developed

In this section they will be briefly shown the main components of the HW system developed: the custom VLSI MLP_chip, the Industry pack bus based mezzanine board, both in its block structure and in its layout. Moreover a technical summary of the mezzanine board is reported.

The MLP_chip

The MLP_chip is devoted to be used as a basic module in hardware networks implementing MLP based neural networks. The topology of the MLP architecture implemented by the MLP_chip is programmable; the weight memory (each weight is an 8-bit data) is implemented off chip. To implement a TMLP network then each MLP must be run-time configurable into a wide variety of different topologies [6]. Possible network topology definitions are down-loaded from a host computer into internal RAMs. Weights are stored in external weight memories directly addressed by the MLP_chip. The MLP_chip has a 20-bit weight addressing space.

Some examples of hierarchical MLP-based architectures are shown in Figure 2: TMLP, CMLP (Coupled MLP) and TCMLP (Tree of CMLP).

The MLP_chip implements a two-layers MLP using two different matrix-vector multipliers (FIRST LAYER and SECOND LAYER). The FIRST LAYER and SECOND LAYER blocks implement

the matrix-vector multiplication, inherent to the MLP execution, and the neuron activation function and they have been optimized to speed-up the execution of the whole MLP. The computation performed by the two layers has been decoupled making them work independently from each other and following two different computational schemes [5],[6]. The two layers work in parallel. Being the module designed for classification problems, it gives as output a list of the more likely classes in decreasing order (e.g. classification): the first class corresponds to the neuron with the highest output value, the second one to the second neuron highest output value, and so on.

The processing rate obtained is 128 Mega-Connection Per Second (MCPS) @32 Mhz clock frequency (taking into account a single MLP_chip). The chip complexity is about 8000 standard cells. It contains 7040 bits of RAM, organized as two dual port RAMs and ten single port RAMs, and four multipliers. The chip has been designed and manufactured in CMOS 0.7 μm technology, double metal, single poly and its area is 6.7×6.3 mm^2. The main chip features are summarized in Table 2; a magnified photo of the silicon chip of the MLP_chip is shown in Figure 3.

Table 1 – IP BUS interfaces

INDUSTRIAL BUS	Max IP single size modules for each carrier board
VME 3U	2
VME/VXI 6U	4
ISA	4
PCI and Compact PCI	2/4

Technology	ATMEL ES2 CMOS 0.7 μm channel length.
Complexity	8000 standard cells and16 macroblocks
PADs count	148
Total Area	6.7×6.3 mm^2
Power consumption	1 W @ 32 MHz
On-Chip Ram	7040 bits
Package	PGA181
Processing rate	128 MCPS @ 32 MHz.

Table 2 – MLP_chip characteristics

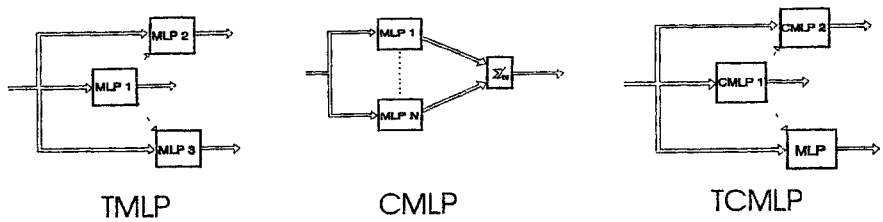

Figure 2: Some examples of hierarchical MLP-based architectures.

The industry pack bus

The Industry Pack Bus is ANSI/VITA 4/1995 standard, defined on standard high density 50 pin connectors. It can be better defined a "meta-bus", as it is possible to easily find on the market carrier board able to interface IP bus modules to the major industrial rack level buses (e.g. Table 1).

The bus main features are:

- 16 bit data bus / 6 bit address bus / Byte or word addressing
- 2 Interrupts for each module
- Main clock frequency: 8 or 32 Mhz
- Peak continuous data rate: > 32 Mbyte/second (single size module at 32 Mhz) > 64 Mbyte/second (double size module at 32 Mhz)

The block structure of the mezzanine module

At the board level the real-time classification system can be considered an application where specialized processors (the MLP_chips) co-operate with standard microprocessors to accomplish application specific number crunching.

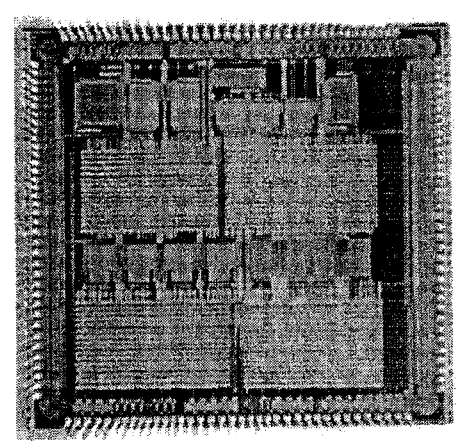

Figure 3: MLP_chip Photograph

the matrix-vector multiplication, inherent to the MLP execution, and the neuron activation function and they have been optimized to speed-up the execution of the whole MLP. The computation performed by the two layers has been decoupled making them work independently from each other and following two different computational schemes [5],[6]. The two layers work in parallel. Being the module designed for classification problems, it gives as output a list of the more likely classes in decreasing order (e.g. classification): the first class corresponds to the neuron with the highest output value, the second one to the second neuron highest output value, and so on.

The processing rate obtained is 128 Mega-Connection Per Second (MCPS) @32 Mhz clock frequency (taking into account a single MLP_chip). The chip complexity is about 8000 standard cells. It contains 7040 bits of RAM, organized as two dual port RAMs and ten single port RAMs, and four multipliers. The chip has been designed and manufactured in CMOS 0.7 μm technology, double metal, single poly and its area is 6.7×6.3 mm^2. The main chip features are summarized in Table 2; a magnified photo of the silicon chip of the MLP_chip is shown in Figure 3.

Table 1 – IP BUS interfaces

INDUSTRIAL BUS	Max IP single size modules for each carrier board
VME 3U	2
VME/VXI 6U	4
ISA	4
PCI and Compact PCI	2/4

Technology	ATMEL ES2 CMOS 0.7 μm channel length.
Complexity	8000 standard cells and16 macroblocks
PADs count	148
Total Area	6.7×6.3 mm^2
Power consumption	1 W @ 32 MHz
On-Chip Ram	7040 bits
Package	PGA181
Processing rate	128 MCPS @ 32 MHz.

Table 2 – MLP_chip characteristics

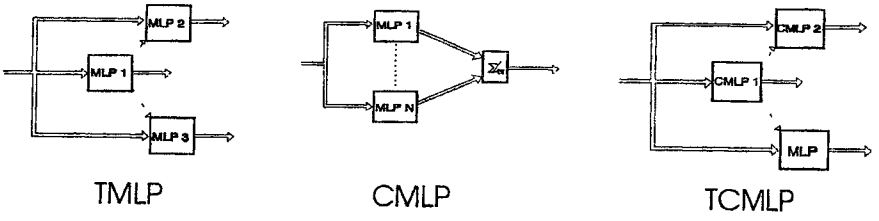

Figure 2: Some examples of hierarchical MLP-based architectures.

The industry pack bus

The Industry Pack Bus is ANSI/VITA 4/1995 standard, defined on standard high density 50 pin connectors. It can be better defined a "meta-bus", as it is possible to easily find on the market carrier board able to interface IP bus modules to the major industrial rack level buses (e.g. Table 1).

The bus main features are:

- 16 bit data bus / 6 bit address bus / Byte or word addressing
- 2 Interrupts for each module
- Main clock frequency: 8 or 32 Mhz
- Peak continuous data rate: > 32 Mbyte/second (single size module at 32 Mhz) > 64 Mbyte/second (double size module at 32 Mhz)

The block structure of the mezzanine module

At the board level the real-time classification system can be considered an application where specialized processors (the MLP_chips) co-operate with standard microprocessors to accomplish application specific number crunching.

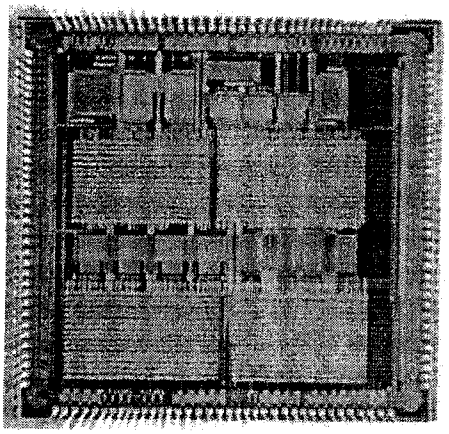

Figure 3: MLP_chip Photograph

Given that the development flow of the NN based real-time classification system can be assumed very similar to the one reported in Figure 6; a consistent user aiding system has been implemented. Using a Windows-like GUI, the development system helps the operator in properly implementing the various phases needed to obtain the full working classification system.

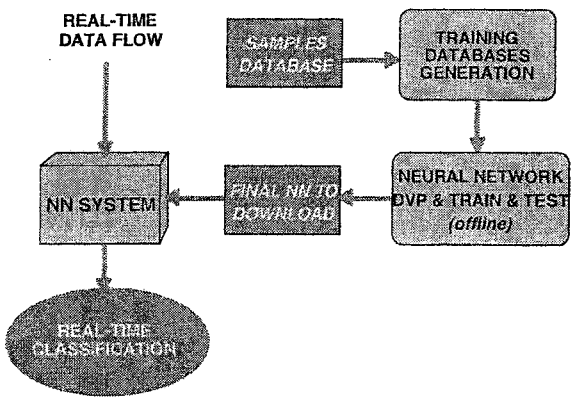

Figure 6: NN based real-time classification system development flow

In order to describe the development system some of the main panels are reported and briefly commented.

In Figure 7, the active desktop implemented for the development environment is shown. A symbolic development flow is traced and the operator is forced to implement the various phases in the correct order, starting from the training database validation and treatment phase up to the NN parameters download to the specialized HW, ending with the aided on-line or off-line testing.

In particular the main steps are:

1. **Data Elaboration phase** *(see Figure 8)*
   - starts from the databases directly collected on the site/plant;
   - organizes the data in a proper hierarchical way in order to make the classification useful (i.e.: to develop the classes by which the incoming data should be divided in)
   - automatically prepares the files required by the NN trainer for all the application NNs.
2. **Training Phase** *(see Figure 9)*
   - starts from the training files worked out by the previous phase;

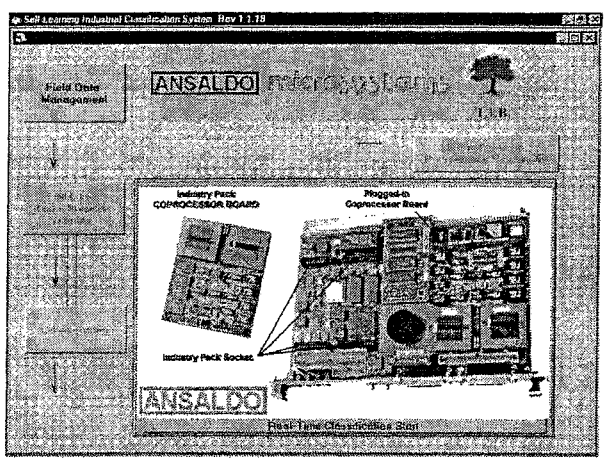

Figure 7: The active desktop

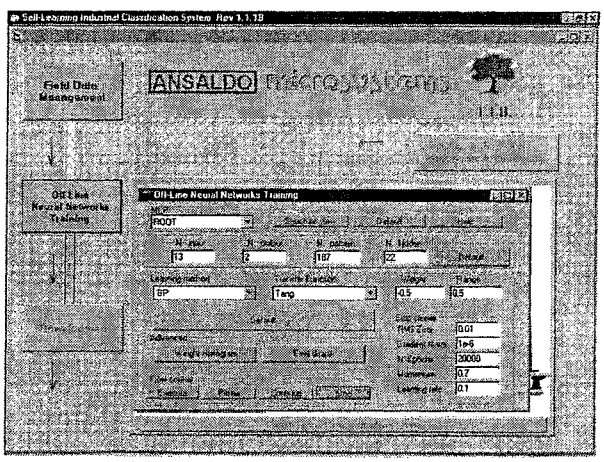

Figure 9: The training tool

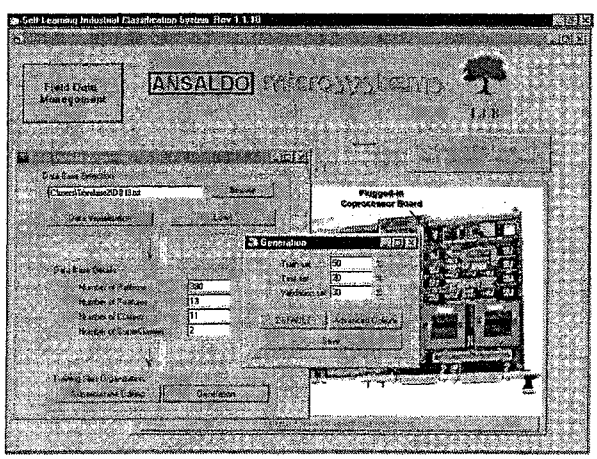

Figure 8: The database elaboration

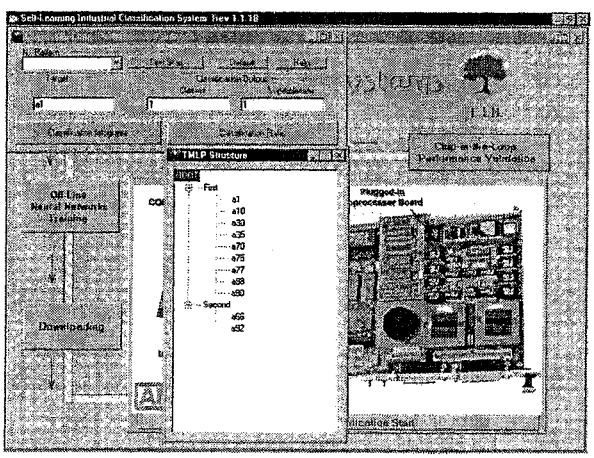

Figure 10: The on-line test

- allows the changing of both the algorithm and the default training parameter set, if required by the operator (otherwise, automatic default choices are selected by the system);
- monitors the training cycle and automatically stops it when some stop criteria are met, in order to assure a satisfactory training for the on-going industrial application;
- tests the developed structure using the specialized NN HW, in order both to be sure of the results and to functionally test the download link.

3. **On-line classification** *(see Figure10)*
   - downloads the parameters worked out in the previous phase to the HW subsystem and switches to the classification 'on-line' mode.

   At this point the development system is no more required and can be quitted. The classification task is then performed in real-time at full speed by the NN HW.

A Sample Application

A system implementing in real time the surface defect classification of steel strips in flat rolling mill has been developed as sample application.

Surface defect recognition of steel strips plays a basic role in quality management of steel makers. Surface defects are one of the most important quality metrics for customers, as they affect heavily the quality (and price) of the products.

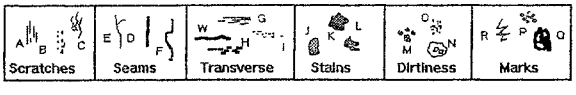

Figure 11: *A sketch of surface defect classes and families in flat rolled mills.*

Better information on defects may provide valuable direct feed-back for process control to reduce costs of quality (internal costs of scrap-rework) and to increase manufacturing productivity and yield. Surface defects are numerous, and may be organized by families such as marks, scratches, stains, etc. The Figure 11 shows a sketch of defect classes and families in flat rolled mills. Defects belonging to different families may have similar visual characteristics, and, as such, they are very difficult to classify. Current approaches rely on human inspectors specifically trained for the job. The inset of Figure 12 shows the surface of a flat rolled mill containing a scratch.

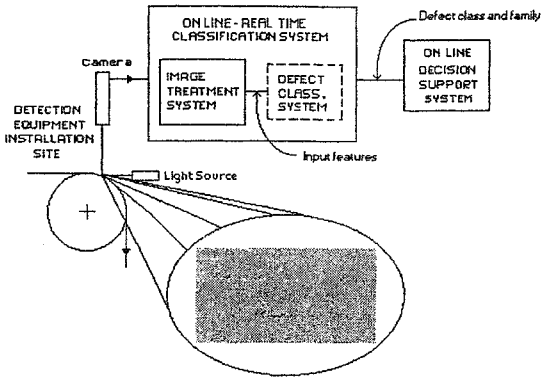

Figure 12: *A schematic of the quality control system.*

The features of samples of steel-ribbon impurities and flaws (i.e. the defects) are obtained with on-line and on-plant measurements. Some features represent the geometrical properties of the imperfections on the strip, while others provide information about illumination, width and thickness of the strip etc. The measurements were collected through an on-line CCD camera coupled with an on-line image acquisition/pre-processing system, which performed filtering and feature extraction tasks on the images collected on the surface of the flat rolled strip.

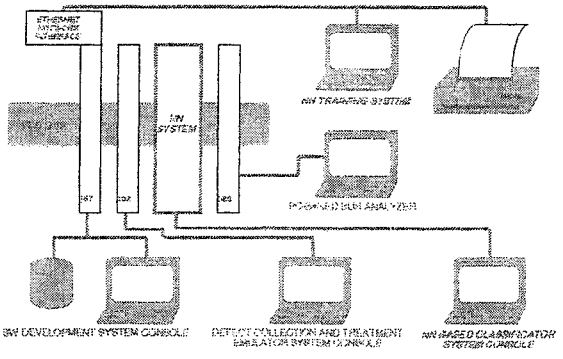

Figure 13: *The performance measurement set-up of the steel defect classification system*

The databases, used for tailoring the application, were obtained with on-line and on-plant measurements; they include samples of steel-ribbon impurities and flaws. Each sample consists of 16 features, some representing the geometrical properties of the imperfections on the strip, others providing information about illumination, width and thickness of the strip etc., and a code number, which identifies the type of defect. Most of the input features come from the on-line image acquisition/pre-processing system.

The image processing system has been tested in a simulated environment (shown Figure), *using real world data collected on the real plant*, in order to

make a reliable assessment of the performances achievable.

It can be mentioned that, on the real plant, the NN system has been implemented using a Motorola MVME162-533 board as carrier board, and it has been integrated in an already existing VME/VMEexec/pSOS+ based defect detection system. The on-line system is shown in Figure 14.

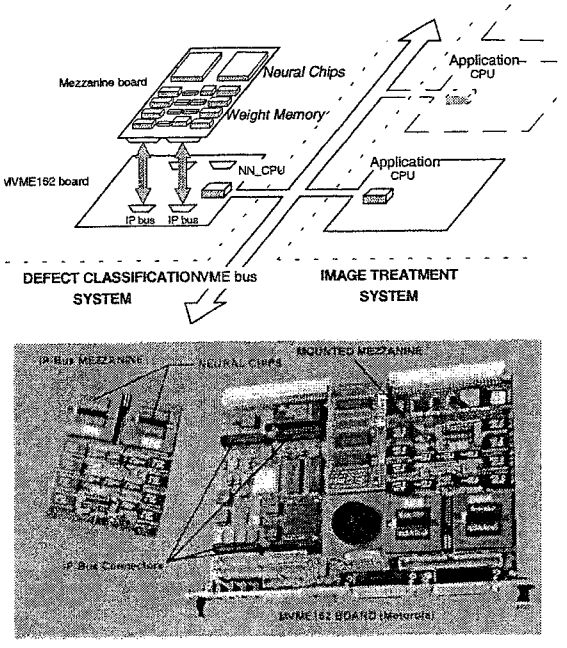

Figure 14: Host CPU: Commercial CPU Motorola MVME162_533 (68040@33 MHz)

Physically MLP_chips are placed on the custom mezzanine board. With reference to the classification real-time requirement, the data transfer between the MLP_chip and the Image Treatment System (ITS) plays a crucial role; in particular:

(i) special care has been taken in developing an optimized custom control logic (Glue Chip) on the mezzanine board;

(ii) the weight data throughput to the MLP_chips has been maintained at 64 Mbyte/s, using special design techniques;

(iii) a MVME162-533 (32MHz 68040 CPU, 16 Mbyte RAM) has been adopted as host for the neural mezzanine, because it guarantees both the data transfer rate and the compliance with the whole application system (ITS) required.

Any communication between the Neural Defect Classification System (DCS) and the ITS modules is done throughout the NN_CPU that executes commands from the ITS System properly interfacing with the mezzanine board. Commands, data exchanges and neural network parameters are carried out through shared memories, accessible from the address space of the VME bus.

Starting from a collection of less than 800 samples of defect collected on a real plant, the system demonstrated to be able to classify the defects better than a skilled operator, with a mean classification time of less than 380ms considering all the computational overhead due to the simple VME-bus communication mechanism adopted for the demonstration unit (one by one classification). The system architecture resulted in a hierarchical neural structure, shown in Figure 15.

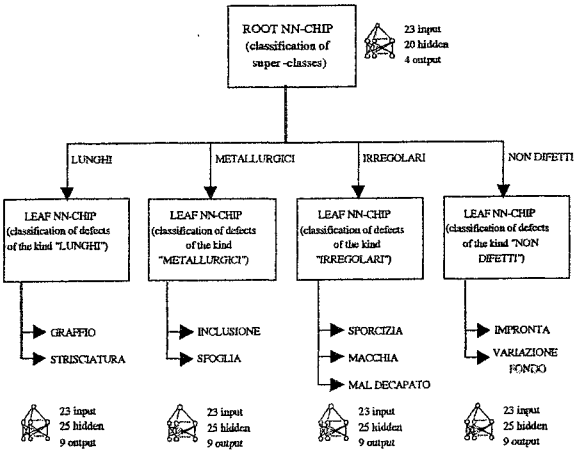

Figure 15: The NN classification system architecture.

The database had 794 samples, collected on the plant by the detection system and classified by the plant operator using 4 main families of objects and 9 particular defect types. From the statistical point of view, the database collected is "poor": the various defect types are very unbalanced. For technical reasons 773 samples out 794 have been used for the test, while the remaining ones have been rejected, because affected by some recording error (missing fields, meaningless "0"s, etc.).

The results obtained have been extremely good, outperforming the mean requirement given by the plant owner (80% of correct classification on a defect of the order of 500 defect per second): the system yields a good 83.6% correct classification ratio at a very good 2611 defect per second classification speed.

Given the poor statistical quality of the database collected from the point of view of the requirement of a good NN training, those results are to be considered even more interesting. As a consequence of this, it is reasonable to expect that the system will outperform *very much more* the specification of the plant managers if a good statistical quality database is used. Anyway, the system is very much faster than specified even if in the test application it is not used the available pipeline subsystem implemented on the chip to fully develop the maximum classification rate achievable (1 classification every 20 μs).

Conclusions

The paper presents a HW system implementing hierarchical MLP_based NNs (e.g.: TMLP) for real industrial applications and the SW toolkit implemented for it. A Neural ASIC co-processor has been developed for the real-time execution of MLP-based NNs: the processing rate of the ASIC is 128 MCPS. An ANSI standard IP bus based neural co-processor board has been developed and tested. Both a VME and a PCI version of the system have been manufactured and carefully tested. The system focuses first of all on the industrial imaging market. It is provided with the completely automatic training wizard, able to:

(i) record directly from the plant imaging system a database suitable for the Neural Network learning phase;

(ii) perform the learning phase starting from that simple Excel-like database;

(iii) validate the learning obtained;

(iv) download to the on-line system the Neural Network architecture realized;

(v) start the on-line classification task.

References

[1] Widrow, B., Lehr, M.A., "30 Years of Adaptive Neural Networks: Perceptron, Madaline, and Backpropagation", Proceedings of the IEEE, Vol. 78, No. 9, pp. 1415 - 1442, Sept. 1990.

[2] Tsoi, C., Pearson, R. A., "Comparison of three classification techniques, CART, C4.5 and Multi-Layer Perceptrons", in R. P. Lippmann, J. E. Moody, D. S. Touretzky (Eds.), Ad. in Neural Information Processing System, Morgan Kaufmann, Vol. 3, pp. 963-969, 1990.

[3] Atlas, L., et al., "A performance comparison of trained multilayer perceptrons and trained classification trees", Proc. of the IEEE, Vol. 78, No 10, pp. 1614 - 1619, Oct. 1990.

[4] Guo, H., Gelfand, S. B., "Classification Trees with Neural Network Feature Extraction", IEEE Transaction on Neural Networks, Vol 3, pp. 923-933, November 1992.

[5] Parenti, R., Penno, C., Caviglia, D.D., Bo, G.M., Baratta, D., Valle, M., Canepa, G., "A Hardware Implementation of Hierarchical Neural Networks for Real-Time Quality Control Systems in Industrial Applications", ICANN'97 – Int. Conf. on ANN, Lausanne (CH), 1997.

[6] Baratta, D., Valle, M., Caviglia, D.D., "Hierarchical Neural Networks for Quality Control in Steel-Industry Plants". Journal of Microelectronic Systems Integration,1997.

[7] Colla, A., Parenti, R., "Neural networks applications in process control", Proc. of Int. Conf. SNN'97 - Neural Network Best Practice in Europe, Amsterdam (NL), 1997.

[8] Caviglia, D., Valle, M., Baratta, D., Baiardo, V., Marchesi, M., "A Neural ASIC Architecture for Real-Time Classification" Proc. of EUROMICRO95 (1995) 632 - 638.

[9] Parenti, R., Bagnasco, A., "Industrial applications of Neural Network Tecnologies", Proc. of the Int. Society for Hybrid Microelectronics Congress, Milano (IT), 1992.

[10] Parenti, R., Bogi, S., Massarani, A., "Industrial Application of Real-Time Neural Networks in Multistage Desalination Plant", Proc. of IDA World Congress on Desalination and Water Sciences, Abu Dhabi, 1995.

[11] Parenti, R., Masulli, F., "Drive Train For Electric/Hybrid Vehicles Improved By Soft-Computing", Proc. of the International ICSC Symposia on Intelligent Industrial Automation - IIA'96, Reading (UK), 1996.

[12] Parenti, R., Penno, C., "A Neural Network Based Approach For The Prediction of Rolling Forces in Steel Hot Rolling Mills", Proc. of the Int. Conf. on Engineering Applications of Neural Networks EANN96, London (UK), 1996.

[13] Parenti, R., Penno, C., Oriati, M., "Set-up System for Hot Rolling based on Neural Networks", Proc. of IIA97 - International ICSC Symposia on Intelligent Industrial Automation 1997, Nîmes (FR), 1997.

[14] Parenti, R., "The Soft-Computing In The Real World: Some Data And Some Considerations", invited paper, pages 298-304 in (A.Bonarini et al. - Ed.s) "New trends in fuzzy logic", Proc. of first Italian Workshop on the Fuzzy Logic WILF'95 - World Scientific, 1996.

[15] Baratta D., Diotalevi F., Valle M. and Caviglia D., "Gradient Descent Learning Algorithm for Hierarchical Neural Networks: a Case Study in Industrial Quality", IWANN99, Brno (CZ), 1999.