USING PVM FOR THE SELECTION OF NEURAL NETWORKS PARAMETERS

R. Baron and H. Paugam-Moisy

LIP - URA CNRS 1398
ENS Lyon, 46 allée d'Italie
69364 Lyon Cedex 07, France
e-mail : rbaron@lip.ens-lyon.fr

11 october 1994

Plan

- Introduction.
- Wavelet networks.
- Learning algorithm.
- Parameter space.
- PVM implementation.
- Conclusion.

Introduction

Given a function f to be approximated by a neural network :

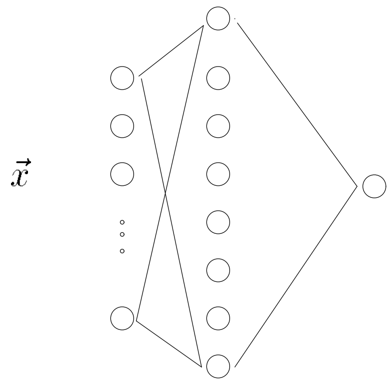
– Which architecture $\begin{cases} \text{number of layers} \\ \text{number of cells per layers} \end{cases}$

– Learning parameters $\begin{cases} \text{learning rate} \\ \text{number of iterations} \end{cases}$

Function approximation using wavelet networks

Given $f : I\!R^n \rightarrow I\!R$ to be approximated

Neural Network with $\begin{cases} \text{n input} \\ \text{one hidden layer} \\ \text{one output} \end{cases}$



$s(\vec{x}) = \sum_{i=1}^{N} w_i \cdot \psi(D_i(\vec{x} - \vec{t_i})) + \overline{g}$

D_i is a dilation matrix (diagonal matrix)
$\vec{t_i}$ is a translation vector
$\overline{g}$ is a parameter for non-zero mean functions

$\boxed{\text{Learning algorithm}}$

Learning set $\mathcal{S}=\{(\vec{x_k}, f(\vec{x_k}))\}$

Cost function to be minimized $c(\vec{x_k}, f(\vec{x_k}) = \frac{1}{2}(s(\vec{x_k}) - f(\vec{x_k}))^2$

Stochastic gradient algorithm :

For each weight X (X stands for $w_i, D_i, \vec{t_i}$ or $\overline{g}$)

$$X_k = X_{k-1} - \gamma \ \cdot \ grad_X \ c(\vec{x_k}, f(\vec{x_k}))$$

Parameter space

architecture with one hidden layer $\Rightarrow C$, number of cells

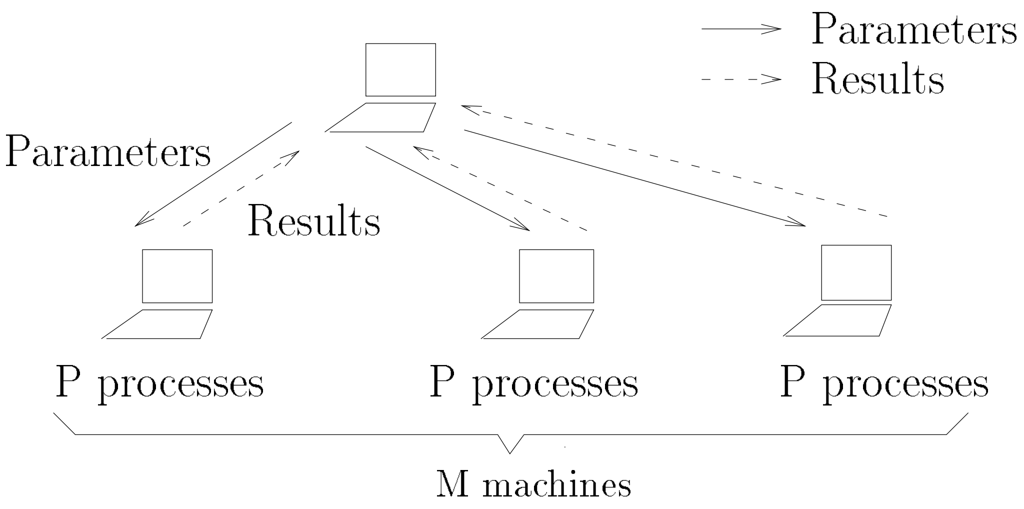
$$\textbf{learning algorithm} \begin{cases} \Rightarrow \text{I number of iterations} \\ \Rightarrow \gamma \text{ learning rate} \end{cases}$$

N_C, N_γ, N_I stand respectively for the numbers of values for C, numbers of values for γ, numbers of values for I

$N = N_C \cdot N_\gamma \cdot N_I$ sets to be tested $\Rightarrow$ Parallel implementation as a solution

PVM implementation

Master-slave architecture



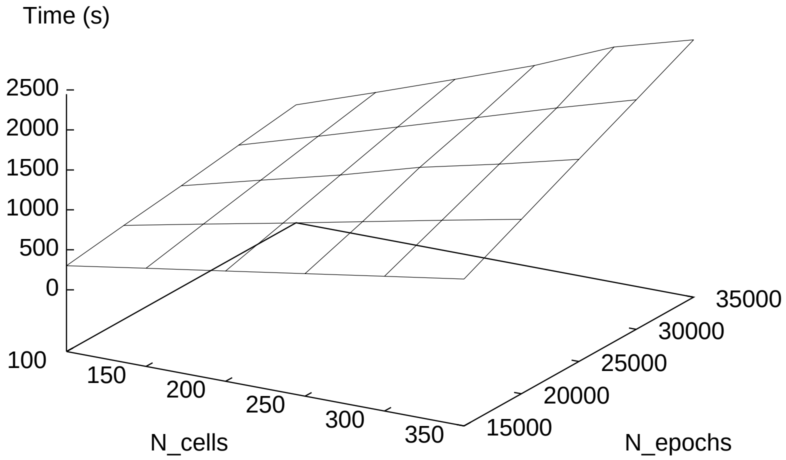
Heterogeneous network of M machines

P processes on each machine

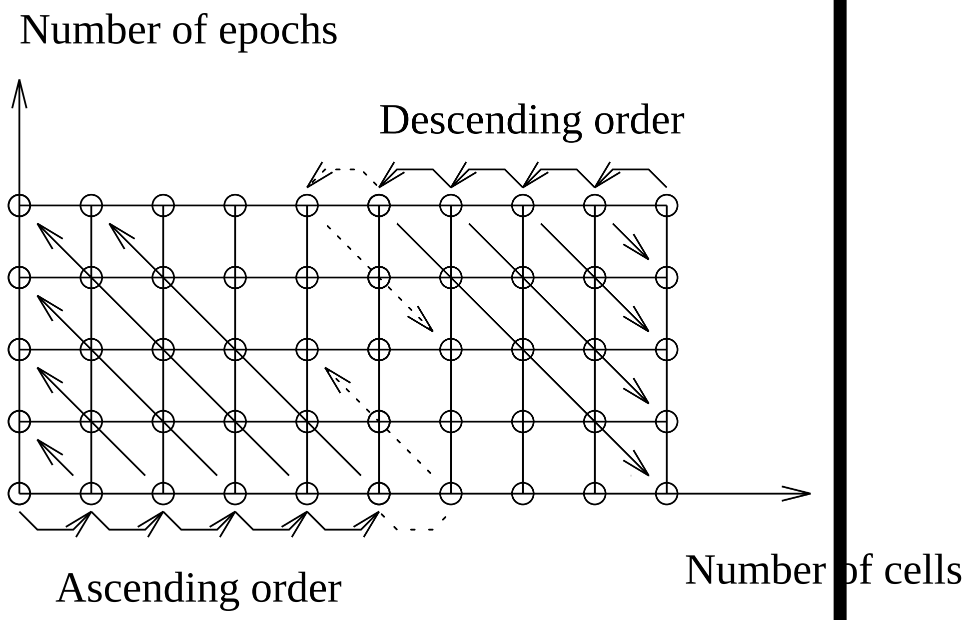
$M.P$ processes $\Rightarrow$ $M.P$ sets tested in parallel

Load balance problem

Computing times increase with I and C



(a) Running time



(b) Parameter space exploration

R. Baron and H. Paugam-Moisy

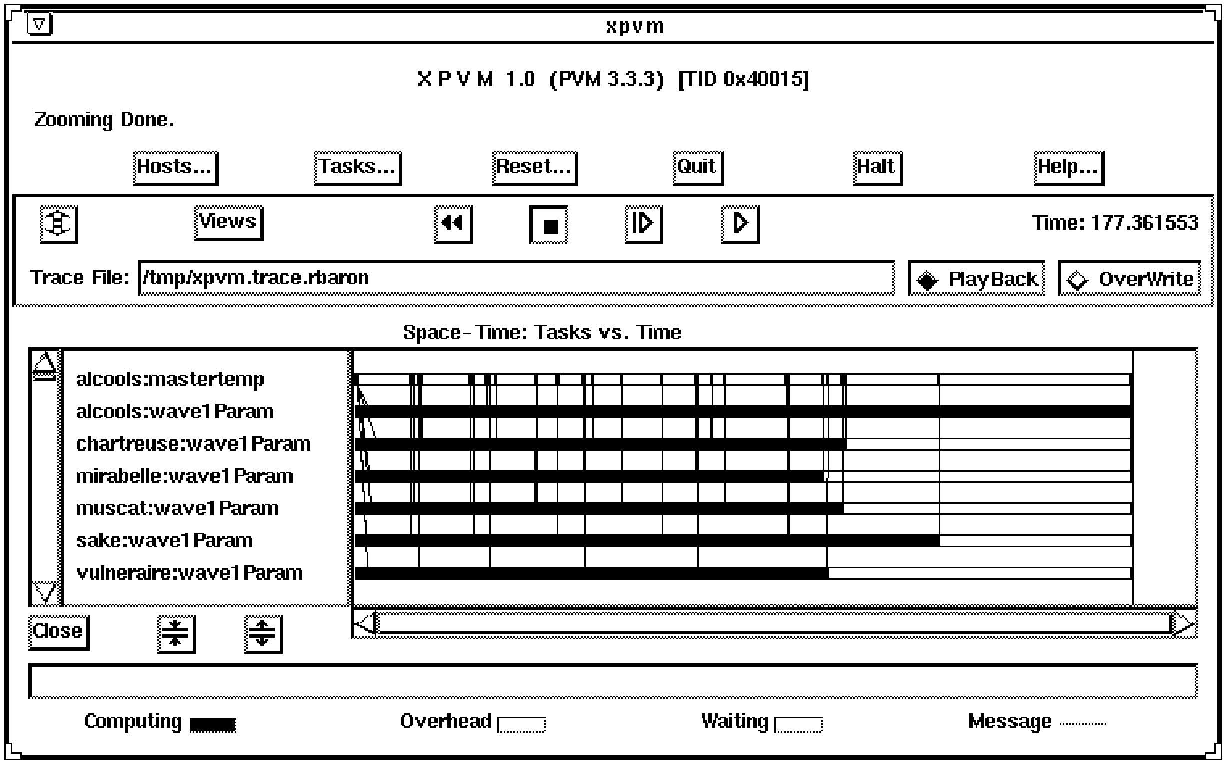


FIG. 1 - *Execution without caution*

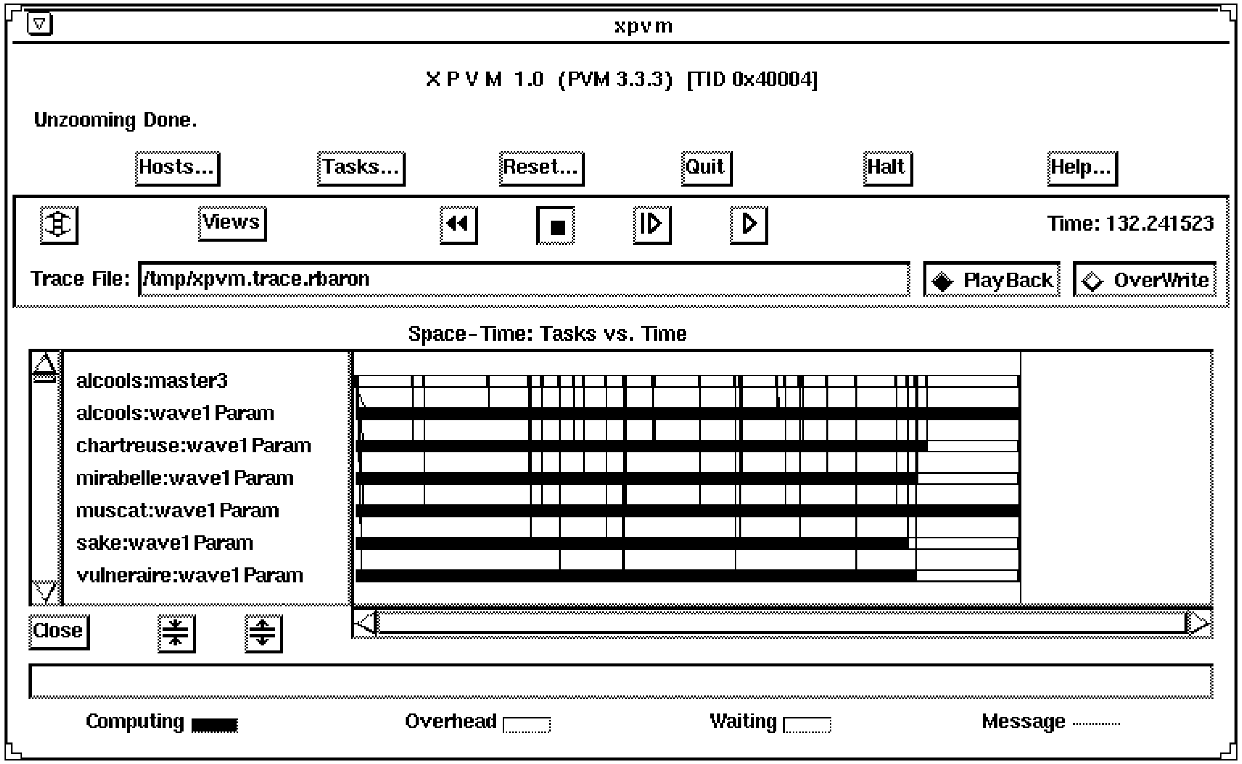


FIG. 2 - *Execution with ordering strategy*

# simultaneous processes	# machines	# processes/machines	time (s)	speed-up s1	speed-up s2
6	1	6	33109	1	0.90
	2	3	22440	1.47	1.33
	3	2	15067	2.19	1.98
	6	1	5837	5.67	5.11
12	12	1	3929	8.42	7.60

Observations

Heterogeneous network $\Rightarrow$ different computing capabilities

Different load $\begin{cases} \text{due to the network} \\ \text{due to the station} \end{cases}$

Conclusion

Allow effective observation (time for test divided by 10)

No special computer requirement

Perspectives

Parallel computer

Automatic selection of parameters :

- finding best set of parameters
- reducing the search around these values