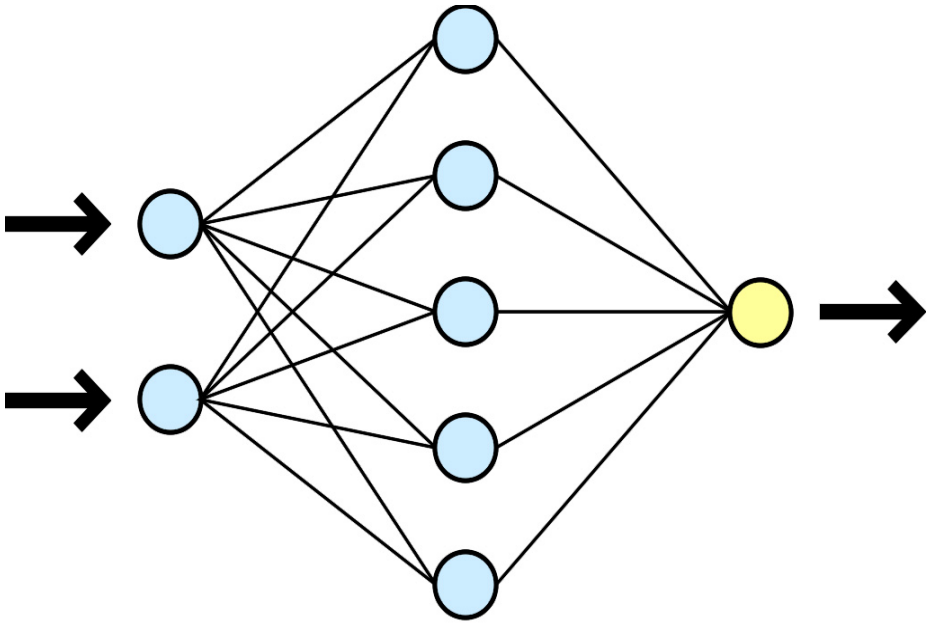
Designing a neural network for forecasting financial time series

29 février 2008

What a Neural Network is ?



Each neurone k is characterized by a transfer function f_k :

$$output_k = f_k\left(\sum_i w_{ik}x_k\right)$$

From a mathematical point of view, a neural network is a function $f : \mathbb{R}^N \rightarrow \mathbb{R}^M$ where the function f is defined as the composition of other function g_i :

$$f = \prod_{i \in I} g_i = g_n \circ g_{n-1} \circ ... \circ g_1$$

Therefore a neural network define a function f_w where w is the vector of weights. The idea is to find the best approximator of a function in the space defined by :

$$C = \{f_{w_1, w_2, .., w_n}\}_{w \in \mathbb{R}^n_+}$$

Where n is the total number of weights.

What a Neural network is not ?

A neural network is not a magic system that takes inputs and find a way of making money by itself ! !

Therefore it is highly important to choose the input data and to calibrate the Neural Net. Nelson and Illingworth outline 8 steps on designing a neural net.

1. Variable Selection
2. Data collection
3. Data processing
4. Training, testing and validation set
5. Neutal network paradigms :
   - Number of hidden layers
   - Number of hidden neurons
   - Number of output neurons
   - transfer functions
6. Evaluation Criteria
7. Neural Network training
   - Number of training iteration
   - learning rate and momentum
8. implementation

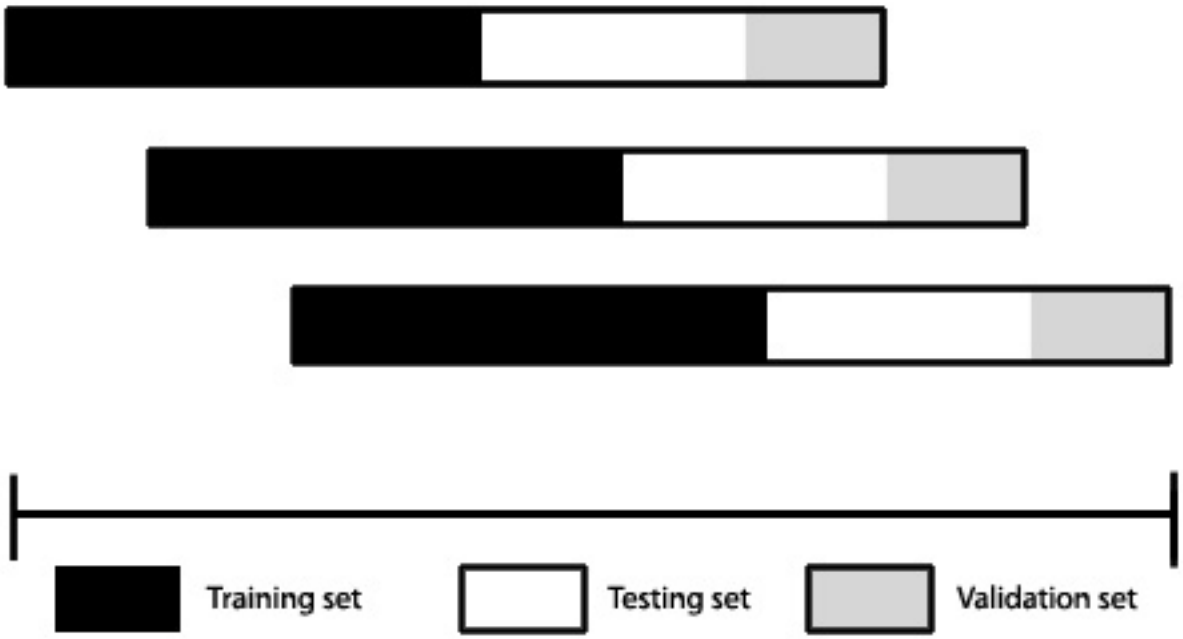
Succes in designing a neural net depends on the clear understanding of the problem.

A neural network can find complex relations between variables, but it is more likely to find them it it is given various technical indicators that are likely to be corralated for economic reasons. For instance one could input :

- Returns of stocks and index.
- Bid/Ask and volumes traded
- Stock price of Microsoft and Apple
- Price of petrol and stock price of GE

One may think to more complicated inputs taking already taking some correlation information into account.

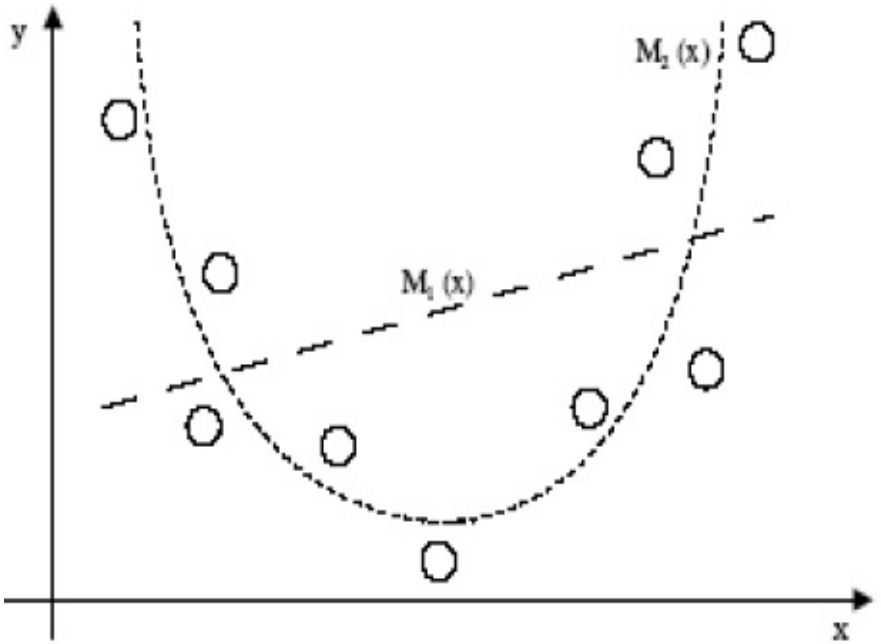
- The researcher would select the NN which performs the best over the testing set.
- The testing setś size is ranging from 10% to 30% of the training set.
- To prevent risk of overfitting, the size of the training set must be at least five times the number of weights.



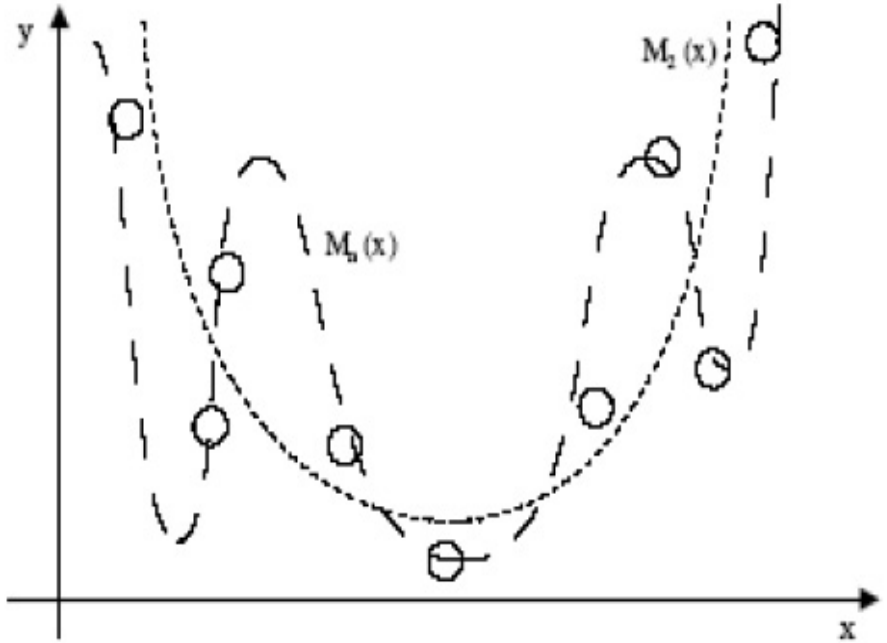
Number of hidden layers

- The hidden layers provide the network with its ability to generalize.
- In theory one layer is enough to approximate any continuous function.

Both theory and empirical work suggest that putting more four layers (one input, one output and two hidden) will not improve the results.
Increasing the number of hidden layers, increases the risk of over-fitting and increases computation time.



(a)



(b)

Number of input and hidden neurons

For a three-layers network it has be suggests that the hidden layer should have approximately :

$$\sqrt{n_{input} \times m_{output}}$$

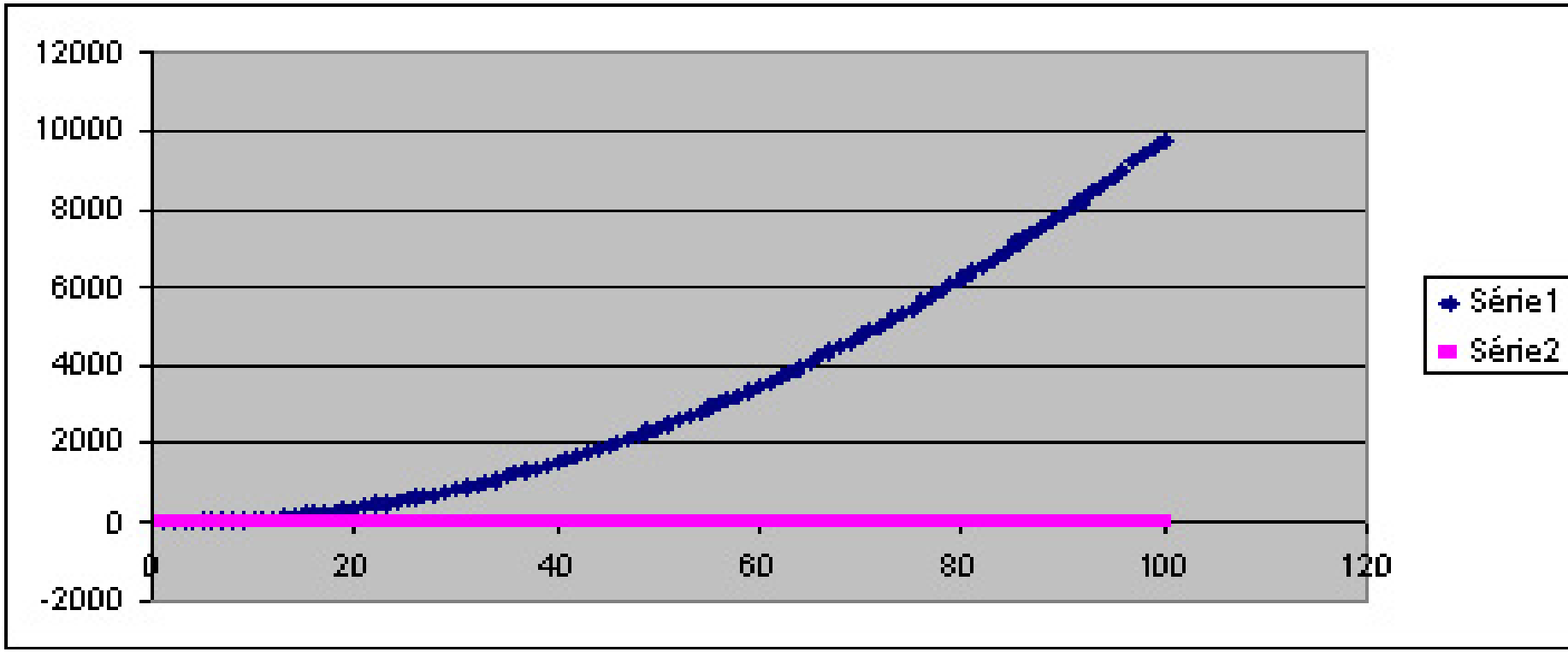
If we use one minutes quotes we have per day : $7 \times 60 = 560$ values divided in 450 in the training set and 110 in the testing set. So we could at most have 90 weights.

We can have approximately 20 hidden neurons...

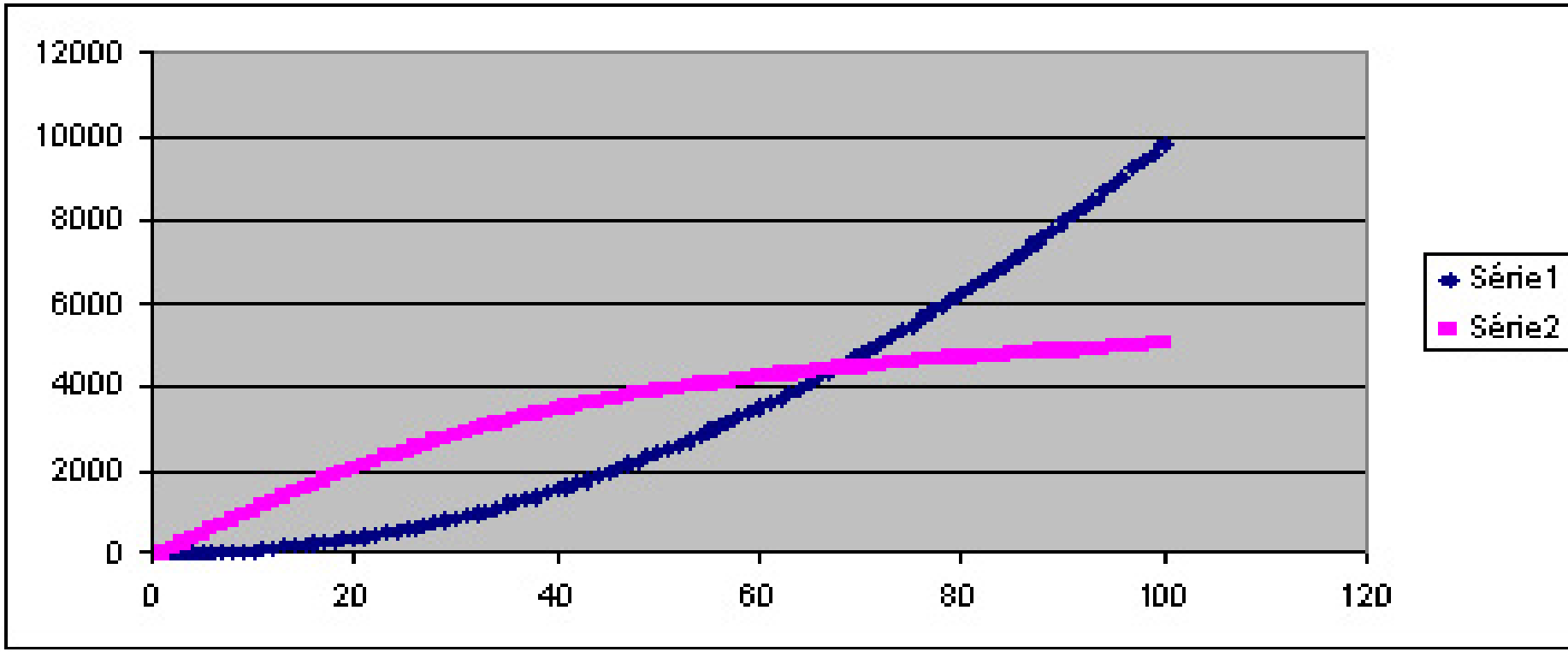
Number of output Neurons

Using multiple outputs will produce inferior results as compared to a network with single output.

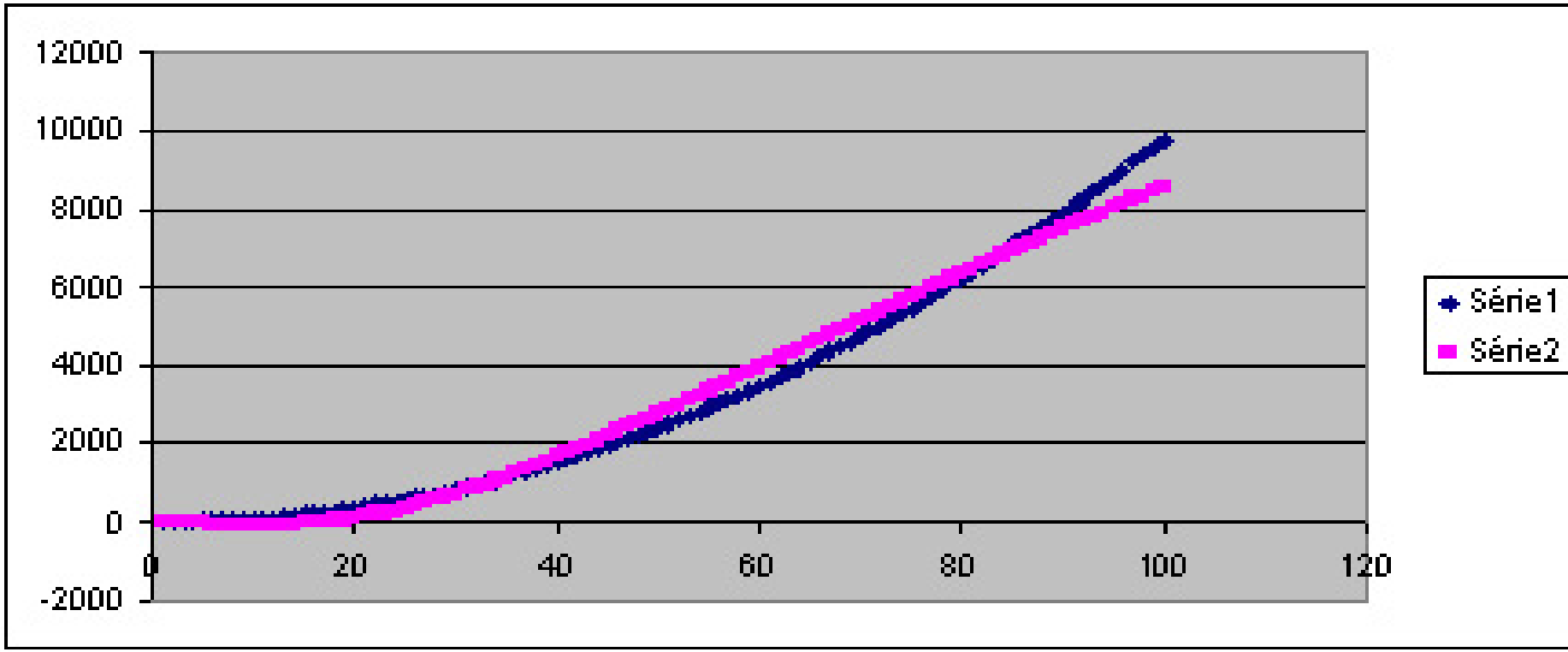
Convergence : 3 Layers, 20 hidden neurons, 50 steps



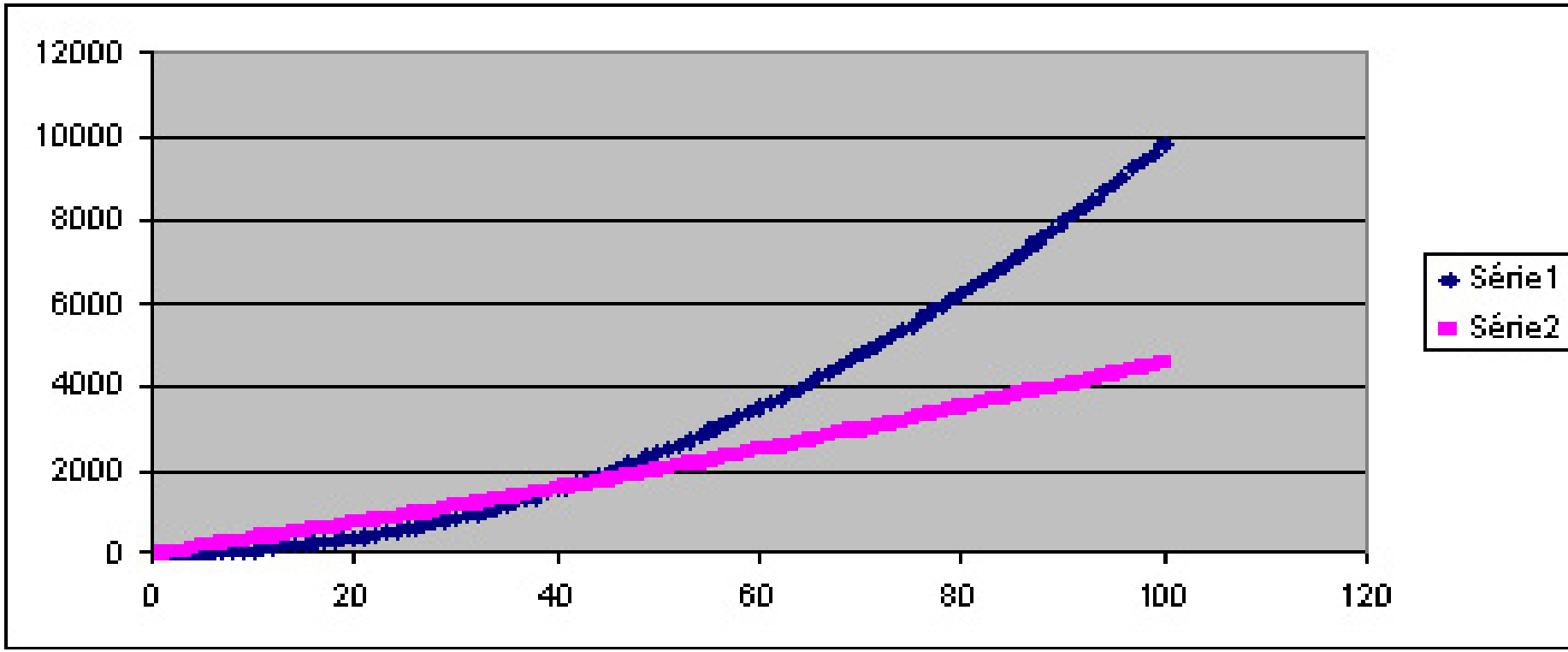
Convergence : 3 Layers, 20 hidden neurons, 100 steps



Convergence : 3 Layers, 20 hidden neurons, 300 steps



Convergence : 3 Layers, 50 hidden neurons, 5 steps



Convergence : 3 Layers, 20 hidden neurons, 50 steps