

Thesis Overview:***Sistema basado en redes neuronales digitales aplicado a la predicción climática en ambientes con microclima controlado***

M. Tosini

Universidad Nacional del Centro de la Provincia de Buenos Aires

Exact Science Faculty

mtosini@exa.unicen.edu.ar

This thesis describes the design, implementation and evaluation of FPGA Architecture of a Digital ANN (Artificial Neural Networks) by using four components: data memory, weights memory, neuron module, and control unit. All these components have a generic orthogonal structure in order to facilitate automatic generation from specific parameters. In a particular ANN design all of its components are generated with specific size, depending on two types of parameters: A) Topological (number of inputs and number of hidden neurons) will determine the size of the memories, the neuron processing time and the microprogram size. B) Architectonic (i.e. word size in bits and circuit granularity) will determine the precision of the results and the final ANN performance.

An equation shows the metric to compare areas of different ANN (number of physical registers) with inputs, hidden neurons, word size and circuit segmentation granularity.

The proposed architecture is programmed by a microprogram with a variable number of microinstructions depending on ANN topology. Each microinstruction configures all datapath signals for each clock cycle, whereas the entire microprogram, cyclically executed, allows obtaining subsequent decisions of the network. The proposed tool uses two levels of abstraction: A) Analyzing the precedence relationships in the computations needed by the network under design. B) Describing if the ANN is Multilayer Perceptrons (MLP) or Hopfield ANN, synchronic and asynchronous.

The synchronic or asynchronous feature of the Hopfield ANN is achieved by the microprogram taking into account that the neurons are all updated in each clock cycle or sequentially, respectively.

A complete set of tools was developed to assist the user in the ANN generation and test. The output of the programming tool is VHDL code to be synthesized to any chip, ASIC or FPGA. NNGen helps the user in the creation and handling of different projects in order to maintain diverse designs. SNN simulates the design allowing analyzing the NN behavior. NNT is a training tool to compute the weights defining the behavior.

- NNGen let to specify the topographic and physical parameters to personalize the circuit, generate the control program, simulate the microprogram operation and, finally, generates VHDL source code for all the components. MLP topographic parameters are three: number of inputs, hidden layer dimension (number of layers and neurons per layer) and number of outputs. These parameters allow the system to generate the control microprogram to drive all datapath signals in each clock cycle. Physical parameters allow the system to adjust all datapath-operators size and to generate data and weight memories (RAM and ROM respectively). The parameters are: a) Weights file, necessary for 'off-line' training; b) Word-size, to determinate internal data size and c) Working precision for current application, in order to establish the Arithmetic to use.
- SNN execute the microprogram with a simulation tool allows the user to make two level simulations: logic simulations to test the correct datapath operation and the architectural simulation to trace data in circuitry. The signals synchronization to handle the systolic datapath is highly complex. For this reason, the simulator allows following the evolution of the calculations in all the points during each clock cycle.
- NNT computes the weight to define the desired behavior off-line.

The proposed design methodology and tools can be used to aid developers making any system; only one project is fully analyzed in this thesis. One of the main problems to solve in temperature forecasting is to achieve the ability of prediction of time series. The ANN approach seems attractive in this task from several points of view. Effectively, there are various ANN architectures, capable to learn the evolutive features of temporal series, and by so doing, to predict future states of these series from past and present information. Perhaps the most used ANN architecture for this kind of prediction is the MLP. For weather forecasting are used stationary temporal series. Then, a reasonable choice is to use some kind of memory to store relevant temporal information, like in Elman neural networks or FIR (Finite Impulse Response) neural networks. The first uses a positive feedback through memory units called “context memory” allowing to add past behaviors of the net to the learning process. On the other hand, in FIR neural networks, FIR filters replace the primary synapses weights, also with the added feature of temporal recalling. This change yields a vectorial and temporal extension to the MLP. Depending on the application, when the prediction horizon is small enough, temporal series may be considered as stationary. However, measuring in time units this prediction horizon for weather forecast applications is still a matter of research and experimental studies.

This thesis presents a methodology with a set of tools to develop different topologies of MLP trained and probed with several temporal series of temperatures with depreciable prediction errors. Hardware architecture for digital ANN design was developed with the main features of an easy configuration from a small set of parameters, a pipelined datapath and a microprogramed control unit. These characteristics have two great advantages: allowing resources reusability and achieving high speeds of operation. Future works includes development of algorithms for automatic generation of other paradigms such as Hopfield networks, Radial Basis Function networks and K-Nearest Neighbor (KNN); and on line MLP training.

The thesis is organized in pipelined flow of chapters. Where chapter one shows an introduction to ANN and presents the objectives. Chapter two presents the state-of-art of the ANN for the calculus of stationary temporal series. Chapter three proposes pipelined microprogramed architecture and its analysis. Chapter four shows the generic design and FPGA implementation, to later be tested in a real case by using Xilinx FPGA in Virtex and Spartan families. Chapter five shows the developing methodology supported by own tools that let the automatic generation of the full circuitry of the architecture. Chapter six shows the conclusions and future works. This document has three appendixes to complete the description of all work.

Nelson ACOSTA
nacosta@exa.unicen.edu.ar