A

Quantum Cellular Neural Networks

A Theoretical Model

By

Najib Saylani

A dissertation submitted to the Graduate Faculty in Computer Science in partial fulfillment of the requirements for the degree of Doctor of Philosophy. The City University of New York

2000

UMI Number: 9959225

UMI®

Bell & Howell Information and Learning Company
300 North Zeeb Road
P.O. Box 1346
Ann Arbor, MI 48106-1346

This manuscript has been read and accepted for the Graduate Faculty in Computer Science in satisfaction of the dissertation requirement for the degree of Doctor of Philosophy.

Jan. 26, 2000 — Date — Chair of Examining Committee

Jan 26, 2000 — Date — Prof. Stan Habib — Executive Officer

Professor Daniel I.A. Cohen

Professor Constantin Negoita

Professor Mohammed Kouar

Dr. Abdelhak Hajjari

Supervisory Committee

THE CITY UNIVERSITY OF NEW YORK

Abstract

Quantum Cellular Neural Networks

A Theoretical Model

By

Najib Saylani

Adviser: Professor Christina M. Zamfirescu

We propose a new theoretical model of Artificial Neural Networks (ANN'S) based on well established ideas in the field of Cellular Automata (CA), the field of Quantum Mechanics (QM) in general, and the field of Quantum Electrodynamic (QED) in particular. The model uses a new architecture where neurons (nodes or units) are located within a 3-D cubic lattice. The connections between the nodes are well defined. The input to the model is a set of patterns encoded as waves packets propagating across the whole set of nodes. The propagation of those waves is analogous to the process of waves scattering across a set of potentials (here potentials exist at the level of each node). Storage and retrieval in this model is based on 3-D scattering. A node behaves as a quantum harmonic oscillator, a feature by which a propagating wave would leave a piece of information it encodes as a set of oscillations at the node level.

Acknowledgements

John Heywood once penned in his Proverbs that it is better to give than to receive.

For the enormity that I have received in the assistance of composing this thesis. I am pleased to give my thanks. Firstly. to my advisors. Professor Christina Zamfirescu and Daniel I.A. Cohen. -- who despite ravages of illness and the obduracy of the author. stuck with me and my idea from the beginning –

I offer my deepest gratitude to Professor Stanley Habib. my thanks for his assistance in this writing and for his agreeableness and open-earedness. and perhaps most importantly. for finding sources of financial support when the going got rough I doff my hat.

I sincerely thank the members of my Examining Committee. Drs. Constantin Negoita. Mohammad Kouar and Abdelhak Hajjari for their input. encouragement and assistance. I thank Professor Michael Anshel for his kindness and consideration.

To Mr. Joseph Driscoll who offered assistance in scaling the Everest of university administrative confusion. I offer my gratitude and an all you can eat buffet of *mdakker*.

To my family. both here and abroad. whose inspiration and encouragement saw me through many an episode of writer's block: especially I would like to recognize Aziz and Malina -- even though they would tut-tut the significance of their contributions toward the completion of this work. I can assure them it was an invaluable asset.

Table of Contents

List of Figures

List of Figures

List of Figures

Introduction

The present work is inspired by current findings in the field of neurobiology and by past and present results in traditional Artificial Neural Networks (ANN). In the light of these domains, this work will take advantage of basic results in the field of Cellular Automata and Quantum Electro Dynamic to explore a new model of ANN, labeled QCNN (Quantum Cellular Neural Network). We are currently witnessing a fair success in designing ANNs which are adequate in solving problems in (and not limited to) computer vision pattern recognition and combinatorics.

Researchers in the past and present have worked on different models of the brain. But in biological systems, modeling many of the brain functions that process, stores and recognizes information is still facing a huge set of obstacles. The problem of consciousness is the most famous one. We have at our disposition a good description of a single neuron at least as far as its internal structure and its molecular composition. We have designed a fairly good map of the brain functions. We can keep track, record and analyze neuronal signals across the cerebral cortex, but we are still far behind in designing a model which will best mimic the human brain.

It is understood that the brain is a set of neurons densely connected as well as being a system capable of massive distributed processing. It exhibits a surprising capability of self-organization and stability by a-not-yet completely known process of self-interactions and interaction with its environment. For most neural researchers, the brain is a non-linear dynamic system capable of stability and equilibrium. Neuro-physiology uses mathematical models to study specific functions of the brain. These models simulate the flow of a signal from one neuron to another across axons and their effect at the surface level of a given neuron. But what has not been fully developed yet is a model which simulates the brain behavior as a whole. Understanding individual parts of the procedures does not explain the collective processes of a collection of interconnected neurons.

Scientists from biology, psychology, computer science, mathematics and physics are all contributing to design a working model of the brain. Together with neuroscientists, they are working on and understanding how the neuron system of the most primitive being (with few neurons) works and interacts with its environment. A single neuron in beings such as the paramecium is still a complex system. It looks like, even a single unit (neuron) is a complex entity but a few are too much to handle. The advantage to partially exploring, even partial the brain functioning, lead others in the fields of Computer Science, Mathematics and Physics to supply neuroscientists with the right tools needed for a fast and efficient analytical processes. At the same time they benefit from their partial results by exploring their application to a variety of problems where other approaches had previously failed.

In any case, all apparently agree that the cerebral cortex is a fast processing system capable of simultaneous parallel and distributed processing. Current results in cellular biology and physiology suggest that a neuron is more than just an on/off switch namely a system where many molecular signals are generated in a more complex structures not fully understood. Recently, the discovery of fine neuron structure called micro-tubules and the theoretical opinion that these micro-tubules are capable of some type of "photonic transportation"-via some type of signaling-encouraged many scientists to explore new ideas in dealing with the brain. The scientists are so optimistic that they started attacking the problem of consciousness and the binding problem, which is the capability of the nervous system to not only recognize a pattern but also to generate or associate with it information or other patterns generated from an other location of the brain.

Our present work will not attempt to explain consciousness (a task of unimaginable difficulty-if not downright impossible). This work will however present a new model and will explore existing techniques claimed to complete techniques applied in other traditional models. The work will propose a model where pattern storage and recognition are done differently with the hope that the closer it gets to ideal implementation, the more powerful a tool it becomes in dealing with different aspects of neuronal processes such as pattern recognition and remote neural communication.

1-Foundations and Background

1-1 Artificial Neural Networks(ANN): Foundations of Traditional ANN.

1-1-1 History

Since the identification of a different type of cell in biology by Ramon Y. Cajal in 1911 (who called them neurons) biologists recognized the need to describe and understand both the structure and functioning of the brain. It was clear to them, as far as structure is concerned, that the brain is a very complex organ with massive interconnections between the neurons. The brain therefore can be said to be a network of neurons. But as far as functionality, little is understood when you compare basics functions to other more complex processes still in research. In parallel with the work of biologists, neuro-psychologists and psychologists after extensive research, proposed various theories in the areas of behavior and consciousness. With the help of mathematics, models were created for a single neuron in order to study its functions and its relation to its environment [1].

It was not until the 1940's that interest in modeling the brain reached its peak with McCullogh and Pitt with their seminar on neural networks. Rosenblatt followed in 1960 with various studies concerning human perception. In 1943 McCullogh and Pitt introduced a two layer concept of perception as a basis for neural modeling. During the same period Minsky and Papert with their work on the logic of threshold networks, realized that their network, as a finate state machine, was capable of universal computations.

The perception of Rosenblatt introduced the idea of "weights" along neural connections. By choosing the appropriate weight, a network can learn how to solve a particular problem. It's the perception of Rosenblatt that was criticized by Minsky and Papert as not being able to solve the XOR problem (1969). Later Rosenblatt tried to overcome that with the development of a theory with multiple layers. Unable to discover a suitable algorithm which maintained neural networks are composed the field of neural network was thrown into a virtual darkness until the 1980's [22].

In 1956, Von Neumann played a part in developing the field of ANN by suggesting the use of redundancy when creating reliable networks, with unreliable parts. In other words, his work suggested that in a system when one part, or a certain number of parts, fail the whole system should in theory be able to continue functioning using the working parts. (Redundancy in structure/function). Von Neumann is most famous contribution to the field was the creation of computers. While Minsky and Papert contributed to freezing research in approaching the creation of intelligent machines with neural network modeling, other researchers continued their efforts in the 70's lead to the introduction of associative contents-addressable memory. It is important to mention that many researchers, not associated with computer science, studied the general problem of learning and contributed indirectly to a comprehensive development of ANN. These developed in parallel different approaches based on their respective main field of study [22].

Marr (1969-1971) biologist approach developed new theories concerning the functioning of the cerebellum and hippo corpus and contributed to locating and assigning specific functions to specific areas in the brain.

Malsburg and Cooper, around 1973, focused their study on the functioning of the visual system. In 1973, I was not trying to understand so "everyone" is wrong. Physics got involved when Cragg and Temperley (1954-1955) first redefined and reconstructed McCulloch-Pitts network this time by constructing a set of interconnected neurons. Each neuron is subject to a certain magnetic orientation which defines its spin. Each neuron's spin would thereby contribute to the spin of the whole network.

The field of ANN's renaissance started in 1981 by incorporating ideas from physics, especially statistical mechanics. Hopfield introduced an energy function and the notion of dynamic local attractors. Then between 1983 and 1986, Hinkron Sejnewski and Peretts used stochastic units in their study of neural networks. Then, still borrowing ideas from statistical mechanics, Amit in 1985 and Al in 1989 developed a system called spin lasses through methods from the field of random magnetic systems. In 1985 Rumelhart and others went back to study the perception with multiple layers and introduced a new method of learning called back propagation, a method widely used and explored in today's applications and research in ANN [22, 30].

The area of ANN is evolving now from a theoretical perspective by accommodating ideas from statistical mechanics. New currents of thought are being explored with sources in the fields of physics and particularly the field of quantum mechanics.

1-1-2 Applications of ANN

Pattern recognition represents the most widely-used and widely-accepted application of ANN. In pattern recognition an ANN will be presented with many variations of the pattern and should be able to converge with the desired one in the retrieval process even if it is represented with a slightly modified one (pattern completion).

Machine vision and speech recognition are also some of the main area of its application. Currently, the applications span other fields such as solving linear systems of equations; mapping parallel programs onto a hyper cube or space mapping; signal processing where ANNs are used to solve partial differential equation; filtering and classifications of numerical parameters ... etc.

In 1983 the neocognitron emerged as one of the most serious application of ANN in visual pattern recognition. This particular ANN is a multi-layered network containing an input layer of photo receptors capable of recognizing handwritten numerals symols. The other layers represent different stages of processing. The last layer is used to retrieve a stored pattern. An input would be processed at the level of each layer using a learning function by which synapses-or interconnections between nodes or neurons-would be represented by a certain set of variable weight [30, 41, 44, 56].

The weight is varied accordingly (actually through the function) until the learning process is accomplished.

The neocognitron is trained and tested with various input patterns. The weights within a given layer are changed adaptively so as to recognize each pattern. The neocognitron is simulated on a computer. The synapses, during learning, are reinforced to form clusters by which the recognition of a given pattern is achieved. The neocognitron is at the source of a true pattern recognition network and can be adapted to recognize not only numeral but also the alphabet or other geometrical shapes. Currently many working (non simulated) pattern recognition systems are based on the neocognitron.

Two good models of ANN lie at the two Polar opposites within the field of ANN: the simple perception theory and the multi-layer perception theory with the back-propagation training algorithms. A simple perception is an ANN with one layer of nodes. The network can learn to recognize a simple pattern by classifying it into one of two sets, A or B.

A node in a perceptron computes the weight which would be passed along by adding the input incident to it from other nodes. The output will be +1 or –1. The output of +1 means the pattern is a class A while the output of –1 represents a pattern of class B. To study this type of network one needs to construct what is called a map of the space of decision which consists of regions created by the input variables. A set of input values would result in a decision for class A and another set of input would result in a decision for class B. This is represented on the ANN's map by two regions separated by a hyper plane. For two inputs, the two regions would be separated by a line. Region A would be on top of the line and region B below the line. The algorithm used will always subtract a threshold (denoted θ in the equation) from the sum of input weights to a give node (here two inputs to one node). The algorithm would initialize the inputs randomly and set the threshold to a random value. Rosenblatt was the first to design a process by which the network would converge to a desired result after adjusting the weights in the network. First a weight W of the i input is set to a value $W_i(0)/0 \le i \le N-1$. N is a random value and θ is set to a small value. $W_i(t)$ is the weight from input i at time t and θ is the threshold in the output node.

Second new continuous valued inputs $X_0.X....X_{N-1}$ are presented with the desired output $d(t)$ then the output is calculated by the following:

$$y(t) = I_n\left(\sum_{i=0}^{N-1} W_i(t)X_i(t) - \theta\right)$$

When the desired output is not attained. the weight should be adapted by setting

$$W_i(t+1) = W_i(t) + \eta[d(t) - y(t)]x_i(t)$$

here η is a positive gain function.

$$\theta \le i \le N-1 \quad , \quad 1 < \eta < d(t)$$

$$d(t) = \begin{cases} +1 & \text{if input in } A \\ -1 & \text{if input in } B \end{cases}$$

when adapting new weight. the weight for a given input at time $t+1$ is equal to the weight at time t plus a fraction of the desired output minus actual output. Rosenblatt showed that the two regions are separable, which means that the output lies in either region (class A or class B).

One of the problems with this model is that if decision boundaries tend to oscillate in a continuous manner then after many trials (or training periods) the two regions may not be separable. The use of Least Mean Square (LMS) avoids this problem by assigning the weight on a connection using a value function of the desired output minus the actual output. The LMS uses 1 for class A and 0 for class B and the threshold above is set to 0.5 when an input is of class A. This method is called the Widrow-Hoft algorithm.

Some classes cannot be separated by a hyper plane. An example would be the representation of exclusive OR (XOR). It requires the map to be separated into 3 regions. This is the problem for which ANN was criticized at the beginning.

The limitation of the single layered perception was overcome by multi-layered perceptions. In these new models three types of layers are used: one input layer. one output layer and a set of intermediate layers called hidden layers. It was shown that linear input can be manipulated by a single layer network and that a multi-layer perceptron takes advantage of non linear inputs. It was shown that a two layer perceptron is needed to separate any convex polygons into disjoint sets which can solve the XOR problem. A three-layer-perception can separate convex from non-convex polygons. These multi-layers networks are easier to study when the regions are polygon shaped. The problem becomes harder when shapes are delimited with curve rather than not straight lines. The above mentioned example of two-or-three-layer networks are characteristics of one type of algorithm called feed forward algorithm.

When decision regions are of more complex shapes, another method is needed. The algorithm is called the back propagation training algorithm. The algorithm feeds forward inputs then backward the output until training is achieved. The algorithm is an interactive gradient algorithm by which the mean square error between the actual output and the desired output is minimized. The algorithm uses a sigmoid function of the form: $l(\alpha) = 1/(1 + e^{-(\alpha-\theta)})$

The output of each layer is fed forward to the next layer. At the end, if desired output is not attained, a recursive process is used to feed back the actual output as input to previous layers and back again until desired output is achieved. A continuous valued input vectors are presented to the network together with the desired output vectors $\{d_0, d_1, ..., d_{M-1}\}$. Samples from a training set make up the input. A sigmoid function is used to calculate the actual output $\{y_0, y_1, ..., y_{M-1}\}$. Weights are adjusted recursively starting back from the output layer by the following equation:

$$W_{ij}(t+1) = W_{ij}(t) + \eta\delta_j x_i$$

where η is a gain term and δ is an error term.

If j is an output node then $\delta j = y_j(1-y_j)(d_j - y_j)$.

If j is an interval node then $\delta j = X_j(1-X'_j)\sum_k \delta_j w_j k$

where k is over all nodes connected to j. This algorithm was used with the XOR problem in speech recognition and pattern recognition. The back propagation algorithm is a gradient search technique which finds a local minimum in the LMS cost function and not the desired global minimum. A global minimum means a convergence to a solution it is sometimes the case that the algorithm gets trapped in a local minimum. In that case no convergence occurs. It was suggested to increase the number of hidden layers be increased in order to increase the chances of getting to a global minimum and to set various random weights at start [1, 22, 30, 41, 44, 56].

The problem with back propagation is that most algorithms based on this technique cannot achieve convergence or global minimum with a number of passes or training steps less than 100. Kolmogorov states in a theorem that a three-layer perception with $N(N+1)$ nodes using continuously increasing non function of N variables. But the theorem does not indicate the choice of weight and internal functions.

1-1-3 Evolution and Limits of Traditional ANN

The search for better algorithms continues. The involvement of scientists from other fields helped in a way by. exploring other methods. Hopfield in 1982 introduced the idea of an energy function. A system of neurons (nodes in an ANN) will evolve from a state (an energy level) to another state (another energy level) according to some dynamical rules. These rules are modeled after the dynamics of a moving particle along an energy surface (represented by an energy function) under the effect of gravity and friction. The particle would slide down the surface to a local minimal called a/the basin of attraction. The set of these attractions are the memorized patterns the ANN can recognize when it is presented with the correct or close to correct pattern. Analogies were found between Hopfield's energy function and some models in magnetic materials. The use of stochastic units in these type of networks makes it necessary to look into statistical mechanics for new ideas to implement in the field of ANN. One of the explored models in statistical mechanics is the Ising Model. The Ising Model is a set of atomic magnets arranged on a regular lattice. The atomic magnets are also called spins. A spin would be oriented up (+1) or down (-1). An active neuron (firing) would have a spin up (+1) and a resting one would have spin down (-1). In the case of an ANN. a neuron is influenced not only by the ones connected to it but also by other at remote locations (with no direct connection to the node but part of the same network). We should note that not all Ising Models are useful when implemented into an ANN.

Models based on statistical mechanics ideas helped in developing better algorithms for applications. such as problems of optimization. pattern recognition in machine vision. Models were built on the ideas of energy function. entropy of a network. stochastic dynamic. ... etc. The Hopfield Model is an example of good use of statistical mechanics in ANN. A large number of properties and results were obtained. Different learning methodologies were developed such as the Hebb Rule. Stochastic Evolution Rule. Mean Field Theory and the Average Activation Rule. Gardner (1987) contributed most to the theory of ANN capacity. Gardner is at the origin of the method used to calculate the capacity (information capacity) of a Hopfield-like recurrent network. Recurrent networks and the Boltzman machine were reformulated using statistical mechanics. Much of the effort was spent on new learning algorithms. Optimal architectures for various networks were attained yet the search remains ongoing for the optimal model architecture and the optimal learning algorithm [42, 44. 86].

As stated previously, a good model of a single function of the brain does not yet exist. A true model should not only present all the capabilities of a single neuron but also all the processes present in a system of multiple neurons (nervous system). Different models target different problems such as associative memory, optimization, pattern recognition and problems in vision. Some of current research effort is concentrated on making existing algorithms perform faster while most tools to do so are imported from statistical mechanics.

Not only does a single algorithm for an ANN capable of solving all problems in the main area of application (optimization, pattern recognition, ... etc.) not exist but, even its implementation beyond simulation (actual physical implementation) is very difficult using currently available hardware. Most ANN are simulated on a digital computer, and only very few simple ones were built on VLSI. There is a limit to how many neurons (nodes) and connections one can build on a circuit. In general, building serious ANN even for the sake of experimentation would depend on how many artificial neurons or units one can use, interconnect and subject to a training algorithm.

There are limitations to simulating ANN on classical computers or the Von Neumann machine. New trends in Optico-electronics are giving new hopes to solving connection problems in the case where the number of nodes to connect increases dramatically. From statistical mechanics optics to new results involving quantum mechanics, some scientists are gaining more insights in the field of neural networks. Recent advances in quantum computation and especially advances in creating physical models to support it are gaining more momentum. A combination of various methodologies is used to approach ANN [47, 54, 79].

1-2 Quantum Electro Dynamic (QED): An Overview

1-2-1 Brief Introduction and Relation to the Model

QED is the quantum field theory of electrons, positrons and photons. Note that here the QED model focuses mainly on weak perturbations during a scattering process. We will consider quantum scattering in 3-D.

All computations are developed from [34].

Key words: *Huygens principle, Green's function, propagator.*

In our proposed model, QED will be used the following way:

- A wave packet would propagate through the units/nodes which are within a 3-D cubic lattice.
- Waves will be detected at each node in given layer (3-D lattice can be divided into a set of 2-D lattices). The wave will interact with local oscillating particles within a given node (we consider each node to represent a set of harmonic oscillators).
- Each node represents those oscillators within a cubic potential whose effect is limited to the node itself and does not interfere with neighboring nodes.
- This potential would cause the wave to scatter to other adjacent nodes.
- Given that scattering happens within a finite macroscopic apparatus, we consider that the waves would vanish at all ends or external 2-D lattices.
- We assume, due to the values of each wave packet, that scattering can diminish or terminate within the 3-D space without reaching the end outside lattice nodes.

- The scattering process is time-dependent; for that reason we would like to present the Schrodinger wave equation [34]:

$$i\eta\frac{\partial(x,t)}{\partial t}=-\frac{\eta^2}{2m}\nabla^2\Psi(x,t)+V(x,t)\Psi(x,t)$$

The equation describes the interaction of a particle of mass *m* with a potential constant in space. (In our case the potential is not fixed-which would cause it to have different equations for different nodes, with different potentials at various given times). Furthermore, in QED, it is well established that:

If Ψ is known at position x, and at time t then Ψ is also know at t_0 <t and t >t.

Note that in our model a good choice of path must be made to analyze the scattering process. Fortunately our model represents positions (nodes/units) within a well defined lattice.

<u>1-2-2 Scattering and the *Green's Function*</u>

QED offers the method by which past and future behavior of the nodes is known (remember that the system here does not exhibit any interference as was stated before. There are no interactions between nodes other than an indirect one).

The approach to finding the behavior of the scattered waves across different nodes consists of the use of the *Green's Function*.

Green's Function describes the probability amplitude associated with the scattering process.

In our model, defining a specific path of scattering will help determine the solution to all *Schrödinger* equations involved. Across all nodes of the path the solutions to a specific *Schrödinger* equation would be linearly superposed on the *Green's Function*. In order to analyze the process of scattering we need to introduce the Huygens Principle:

$\Psi(x,t)$ at time t' position x' is known if $\Psi(x,t)$ is known at source x at time t.

$$(1)\ \left[\Psi(x',t') = i\int d^3x G(x',t':x,t)\Psi(x,t) t' > t\right.$$

The intensity of Ψ at (x',t') is proportional to Ψ at $\Psi(x,t)$

Here $iG(x',t':x,t)$ is the constant of proportionality in (1)

$G(n',t';x,t)$ is the *Green's Function* of the *Hamiltonian* $\hat{H}$ or *Hamiltonian operator* $\hat{H} = -\frac{\eta^2}{2m}\nabla^2 + V(x,t)$

In our care we must consider propagation forward and backward. This assumes that within our model if the path is defined then one can proceed by assuming either a forward or backward propagation.

For that case there are two types of *Green's Function*:

- *Forward or retarded Green's Function*

$$G^{+}(x',t':x,t) = \begin{cases} G(x',t':x,t) & t'>t \\ 0 & t'<t \end{cases}$$

That is a causal backward or evolution of $\Psi(x,t)$ from $\Psi(x',t')$

- *Backward or advanced Green's Function*

$G^{-}(x',t';x,t)=\begin{cases} G(x',t';x,t) & t'<t \\ 0 & t'>t \end{cases}$ This is a causal backward or evolution of

$\Psi(x,t)$ from $\Psi(x',t')$

Determination of a resulting wave packet backward or forward necessitates the introduction of step-function θ.

$$\theta(\tau)=\begin{cases} 1 & for\ \tau>0 \\ 0 & for\ \tau<0 \end{cases}$$

The causal evolution of $\Psi(x',t')$ from $\Psi(x,t)$ where $t' > t$ can be represented by

$$\theta(t-t')\Psi(x',t')=-i\int d^3xG^{-}(x',t':x,t)\Psi(x,t)$$

It is noted that the *Green's Function* is considered only when a potential V of a scattering position x and at time t is not zero. $(V(x,t)\neq 0)$. If $V(x,t)=0$ we speak of a *free Green's Function* noted G_0.

As stated before, when a scattering path is chosen in our model at time t_1 and at position x_1 (a node x, of a given path) the corresponding scattered wave can be found by solving the following *Schrodinger* equation:

$$i\eta\frac{\partial}{\partial t_1}\Psi(x_1,t_1)+\frac{n^2}{2m}\nabla^2\Psi(x_1,t_1)=V(x_1,t_1)\Psi(x_1,t_1)$$

And if $V(x_1,t_1)$ is on during a fraction time Δt_1:

$\Psi(x_1,t_1)=\phi(x_1,t_1)+\Delta\Psi(x_1,t_1)$

ϕ: free wave

$\Delta\Psi$: scattered wave

if $V(x_1,t_1)$ is off after Δt_1:

$$\Delta\Psi(x_2,t_2) = i\int d^3x.\nabla G0(x_2.t_2;x.t.)\Delta\Psi(x_1,t_1)$$

Other computations lead to:

$$\Delta\Psi(x_1,t_1) = \frac{1}{\eta}V(x_1.t_1)0(x_1.t_1)\Delta t_1$$

Before we get lost in QED we must remind ourselves of the following: the *Green's Function* and *Huygen's Principle* will enable us to compute a wave function at a given position at a given time when the wave is known at a former position and time (in the same scattering process).

In our model each neuron or unit

- Works as a scattering center.
- Scattering starts at one side of the 3-D lattice and proceed to one end (or the last 2-D lattice)
- The choice of scattering path within the nodes in the 3-D lattice may be critical in reconstructing the waves as we can have many options to use solving different equation across different paths.

Everything that was introduced before is part of the QED background from [34]. Here we will cover applications to the scattering process in our model. Several assumptions will be presented.

First we assume that scattered waves are originated from the process of photon interaction with the 2-D input lattice. From there, generated waves would propagate across the remaining 2-D lattices. At the level of the first layer, potentials may be turned on/off. We would like here to compute or construct the wave $\Psi(x.t)$ at a node x at time t during a potential V. We note that by $V(x.t)$

Assume $\phi(x,t)$ is the initial wave associated with the incoming particle; it is established that $\phi(x,t)$ is a solution of the *Schrodinger* equation for a free particle. Therefore, if no interaction affected $\phi(x,t)$ then it is stated that $\Psi(x,t) = \phi(x,t)$.

In the future at position $x^{'}$ and time $t^{'}$ we have the following result:

(2) $\Psi^{(+)}(x^{'},t^{'}) = \phi(x^{'}) + \int d^4 x_1 G_0^{+}(x^{'};x_1)V(x_1)\Psi^{(+)}(x_1)$

$(+)$: future scattering

Here $\Psi(x_1)$ is the original wave packet. The second term of (2) is the scattered wave.

We are dealing here with more than one incoming particle. So we consider other conditions on potential switching as well as on its timing.

Within our model, scattering can travel various paths but all the information about the scattered waves is contained in the probability amplitude associated with the particle.

Wave propagation is controlled across the model by limiting the scattering to within our 3-D lattice and by acting on the potentials.

1-2-3 Scattering and the *S-Matrix* in QED

We summarize in the following section all the material from [34] necessary for us to work with.

The most important tool here will be the use of the *S-Matrix* or the *scattering matrix* by which we compute and find the results of a given scattering process.

Here we model the process of wave scattering along the theory presented in [34].

We must remind ourselves that the most critical process here is the problem of normalization.

In [34]:

An initial wave ϕ_i will be scattered to different positions and so to different states ϕ_f (final states).

In our model the scattering is contained within a 3-D cubic lattice. and so potentials are non-existant outside of it. If ϕ is considered a plane wave then

$$\phi_f(x^{'}.t^{'}) = \frac{1}{\sqrt{(2\pi\hbar)^3}} \exp[i(kf \bullet x^{'} - \omega fi^{'})]$$

This equation will be approached using the continuum normalization also called the ∂-function normalization. An alternative to this is necessary box normalization. The particle is within a box of volume V; which will cause the momentum variable to become discrete. and the quantity $\frac{1}{\sqrt{(2\pi\hbar)^3}}$ will be replaced by $\frac{1}{\sqrt{V}}$ fortunately our model benefits from this since it is a 3-D cubic lattice of nodes where a node is an actual cubic volume of a specific height. The potential at the node level increases and get dimmed within this volume.

In [34] it is stated that *Dirac's* ∂ *function* $\partial^3(kj - k_i)$ is replaced *by Kronecker's delta* (k being the momentum).

$$\partial_{kj,ki} = \begin{cases} 1 \ for \ kj = ki \\ 0 \ for \ kj \neq ki \end{cases}$$

The probability amplitudes are elements of the *Heisenberg's scattering matrix* or *S-matrix*.

Let us present the established result, or the *S-matrix*, which is a solution for the aforementioned equation. Again it has been previously established before that a future wave $\Psi(x,t)$ is represented by the following equation:

$$\Psi^{-}(x',t') = \phi(x') + \int d^4x_1 G_0^{-}(x';x_1)V(x_1)\Psi^{(-)}(x_1)$$

where $\int d^4x_1 G_0^{+}(x^1;x_1)V(x_1)\Psi^{(-)}(x_1)$ is the scattered wave solution considered in the following expression of the *S-matrix*

$$S_{\ } = \partial^3(kj - ki) + \lim_{t\to\infty} \int d^3x' d^4x \phi_j^{*}(x',t')G_0^{-}(x',t':x,t)V(x,t)\Psi_i^{(-)}(x,t)$$

In our case t tends to be a finite (albeit sometimes lengthy) period of time. Given the finite dimension of the model, time will be critical. For multiple scattering events we need to insert Ψ^{-} into the expansion of S_{n}

$$S_{\ } = \partial^3(kj - ki) + \lim_{t\to\infty} \int d^3x' d^4x \phi \dot{j}(x',t')G_0^{-}(x',t':x,t)V_0(x,t)\phi_i(x,t)$$
$$+ \lim_{t\to\infty} \int d^3x' d^4x_1 d^4x \phi \dot{f}(x',t')$$
$$x G_0^{-}(x',t':x,t)V(x_1,t_1)G_0^{-}(x_1,t_1:x,t)V(x,t)\phi_i(x,t)$$
$$+ \ldots$$

The first integral represents a single scattering, the second represents double scattering ... etc.

Remember that we should be able to compute future values of a scattered wave and also past values. Here we notice again the need to express things using both the *advanced* and the *retarded Green's function.*

State $\phi_j(x,t)$ corresponds to the past wave function $\Psi^{(-)}(x,t)$ causal of $\phi_j(x,t)$.

$$\lim_{t\to\infty}\Psi_f^{(-)}(x,t)$$

$$\Psi_f^{(-)}(x,t)=\lim_{t\to\infty}-i\int d^3xG^-(x,t;x',t')\phi_f(x',t')$$

In the distant past

$$S_{fi}=\lim_{t\to-\infty}\langle\Psi_f^{(-)}(x,t)\,|\,\Psi_i^{(-)}(x,t)\rangle$$
$$=\lim_{t\to-\infty}\lim_{t\to\infty}i\iint d^3xd^3x'G^{-*}(x,t;x',t')\phi_f^*(x',t')\phi_i(x,t)$$
$$=\lim_{t\to\infty}\langle\phi_f(x',t')|\Psi_i^{(-)}(x',t')\rangle$$
$$G^-(x',t';x,t)=G^{-*}(x,t;x',t')$$

In this present work specific conditions are explored. Those conditions have to do with the choice of potential values at different states. The potentials in the model are real and, as shown in [34], are necessary in order to increase the probability of efficiently monitoring the propagation process of a given particle.

In [34] the *S-matrix* can be represented by

$$S_{fi}=\langle\Psi_f^{(-)}(x,t)|\Psi_i^{(-)}(x,t)\rangle$$

which shows the overlap between $\Psi_i^{(-)}$ of the incoming condition and $\Psi_f^{(-)}$ of the outgoing condition.

It is shown that this result is independent of time.

In the next section a case will be shown where determination of a state for a wave $\Psi(x,t)$ can be done independent of time.

For that case we will present a detailed example using *Green's function* and show the propagation into the future and in the past of a free quantum particle.

We saw before that it is critical to monitor the potential switching at different levels and times we must also study the cases where the process is independent of time. That's the case of 'stationary' or local time-independent scattering.

Note that in our model we will focus on two aspects of information storage and retrieval.

In the next section, we will focus on details of 3-D scattering and we will base our results on books and software published by Brand and Dahmen [8.9].

1-3 Scattering of Waves Packet

1-3-1 3-D Scattering of Spherical Waves

In this section we will focus on another important aspect of our model which is the process of scattering in 3-D. All computations and conclusive results in this chapter are developed in [8.9] and used to relate 3-D scattering ideas to our model. In our case each node would represent step potentials within a set of N regions. Those regions belong to the interval $[0. r_1 .... r]$ within which a region will be chosen. Each region would represent a specific potential from the interval $[V .... V_N]$ such that the potential V at region r can be represented by the following:

$$
\begin{aligned}
V(r) = & \bullet V_1 \quad when \quad 0 \le r < r_1 \\
& \bullet V_2 \quad when \quad r_1 \le r < r_2 \\
& . \\
& . \\
& . \\
& \bullet V_N \quad when \quad r_{N-1} \le r
\end{aligned}
$$

In this case we have continuum states for potential within an infinite range.

Here we will choose only the case where the EigenValues $E < V_q$. In this case the wave number $k_q = i\kappa / \kappa_q = \left|\sqrt{2M(V_q . E)} / \hbar\right|$

Let's present the radial wave function with an angular momentum l with N values

$$R_l(k,r) = \bullet R_{l1}(k_1,r) \quad when \quad 0 \le r < r_1$$

$$\bullet R_{l2}(k_2,r) \quad when \quad r_1 \le r < r_2$$

.

.

.

$$\bullet R_{lN-1}(k_{N-1},r) \quad when \quad r_{N-2} \le r < r_{N-1}$$

$$\bullet R_{lN}(k_{N-1},r) \quad when \quad r_{N-1} \le r$$

$k = k_i$ is the wave number at region i for incident and reflected waves. The R_{l_q} pieces for $q = 1.....N$ each can be written the following way

$$R_{l_q}(k_q,r) = A_{l_q} j_l(k_q r) + B_{l_q} \eta_l(k_q r)$$

j_l and n_l are spherical *Bessel* functions (first and second Kind) j_l and n_l are real and $R_{l_q}(k_q,r)$ is a real when the coefficient $A_{l_q} j_l, B_{l_q} n_l$ are real.

Here:

$$j_l(k_q r) = \sqrt{\frac{\pi}{2k_q r}} \quad j_l + 1/2^{(k_q r)}$$

$$n_l(k_q r) = -\sqrt{\frac{\pi}{2k_q r}} \quad N_{l-1\,2}{}^{(k_q r)} = (-1)^l j_{-l-1}{}^{(k_q r)}$$

using the following Spherical *Hankel* functions (the spherical function of the 3^d kind)

$$h_l^{(+)}(k_q r) = n_l(k_q r) + ij_l(k_q r)$$

$$= i\left[j_l(k_q r) - in_l(k_q r)\right]$$

$$= i\sqrt{\frac{\pi}{2k_q r}} \quad H^{(1)}_{l-1\,2}(k_q r)$$

$$h_l^{(-)}(k_q r) = n_l(k_q r) + i j_l(k_q r)$$
$$= -i\left[j_l(k_q r) + i n_l(k_q r)\right]$$
$$= -i\sqrt{\frac{\pi}{2k_q r}}\; H^{(2)}_{l+1/2}(k_q r)$$

$h_l^{(+)}$ and $h_l^{(-)}$ are 1st and 2nd kind spherical *Hankel* functions. Both $h_l^{(-)}$ and $h_l^{(-)}$ can be written the following way:

$$h_l^{(-)}(k_q r) = C_l^{(-)} \frac{e^{+i(k_q r)}}{k_q r}$$

and $$h_l^{(-)}(k_q r) = C_l^{(-)} \frac{e^{-i(k_q r)}}{k_q r}$$

We can now express $R_{l_q}(k_q r)$ using the spherical *Hankel* function:

$$R_{l_q}(k_q r) D_{l_q} h_l^{(-)}(k_q r) + F_{l_q} h_l^{(+)}(k_q r)$$

in this latter equation

$$D_{l_q} = -\frac{1}{2i}\left(A_{l_q} - B_{l_q}\right)$$

$$F_{l_q} = \frac{1}{2i}\left(A_{l_q} + B_{l_q}\right)$$

if k_q is real

$$h_l^{(-)}(k_q r) = C_l^{(-)} \frac{e^{-i(k_q r)}}{k_q r} \qquad \text{(developed above)}$$

representing an incoming spherical wave from a radial distance r in region q and approaching the origin $(r = 0)$

$$h_l^{(+)}(k_q r) = C_l^{(+)} \frac{e^{i(k_q r)}}{k_q r}$$

represent an outgoing spherical wave with k_q real again wave is a <u>superposition</u> of

incoming wave $h_l^{(-)}$ and an outgoing wave $h_l^{(+)}$ • $(case E > v_q)$ (both complex).

At region N $(E > V_N)$ k_N is real and $h_l^{(-)}(k_N r)$ and $h_l^{(+)}(k_N r)$ are both the

incident and the reflected spherical wave with an angular momentum l.

Now let's develop one aspect we would like to explore in our model when

$k_q = i\kappa$ with the eigenvalues $E < V_q$ (potential at region q with $q \neq N$) in this case

the scattering wave $R_{l_q}(i\kappa_q.r)$ is a *<u>linear superposition</u>* of the real spherical *Hankel*

functions

$$\left.\begin{aligned} h_l^{(-)}(i\kappa_q r) &= C_l^{(-)}\frac{e^{k_q r}}{r} \\ h_l^{(-)}(i\kappa_q r) &= C_l^{(-)}\frac{e^{k_q r}}{r} \end{aligned}\right\} k_q = i\kappa$$

using D_{l_q} and F_{l_q}

$$R_{l_q}(i\kappa_q.r) = \frac{1}{2}\left[(A_{l_q} - iB_{l_q})h_l^{(-)}(i\kappa_q.r) - (A_{l_q} + iB_{l_q})h_l^{(-)}(i\kappa_q r)\right]$$

for real coefficient $i(A_{l_q} + iB_{l_q})/2$

and $i(A_{l_q} - iB_{l_q})/2$

the wave functions $R_{l_q}(i\kappa.r)$ are real functions. This lattice is interpreted as the

<u>tunnel effect</u>.

What it mans is that the eigenvalues E being less than V_q (potential at region q)

would cause the wave representing the propagating particle would penetrate the

potential wall of height V_q.

Lets map this info to our model by defining both the boundary and continuity conditions. The coefficient A_{l_q} and B_{l_q} represent a system of inhomogeneous Linear Equations two conditions arises when at the node center $(r = 0)$ $R_{l_1}(k_1, r)$ has no singularities (physical observation). This result requires that $B_{l_1} = 0$ meaning

$$R_{l_1}(k_1, r) = A_{l_1} j_l(k_1, r) + B_{l_q} \eta_l(k_q r)$$

$$n_l(k_1 r) = -\sqrt{\frac{\pi}{2k_1 r}} \; N_{l-1/2}{}^{(k_1 r)}$$ has a singularity at $r = 0$

$$= (-1)^l j_{-l-1}{}^{(k_1 r)}$$

Away from the center and reaching the region N when A_l and $j_l^{(k_1 r)}$ are assumed known we get an incoming and an outgoing spherical wave.

Note that to get scattering solutions the radial wave function has to be continuous and continuously differentiable at $[r_1 .... r_{N-1}]$.

In this case we have $2(N-1)$ inhomogeneous linear algebraic for $2(N-1)$ coefficient

$A_{l_1}, A_{l_2}, B_{l_2} .... A_{lN-1}, B_{lN-1}, B_{lN}$

The coefficient are unique.

$(E > V_q)$ j_l, n_l real in region where k_q is real $(E < V_q h_l^{(-)})$ and $h_l^{(-)}$ are real in region where $k_q = i\kappa_g$ the solution $A_{L_1}, A_{l_2}, B_{l_2} .... A_{lN-1}, B_{lN-1}, B_{lN-1}, B_{lN-1}$ are real solutions functions of the incoming wave number $k \Rightarrow R_l(k, r)$ is a real function. The main ideas to explore for the radial scattering wave is the existence of continuum states for the potentials, existence of real solution and the presence of the *tunnel effect*.

We will make our choice later, but first let's describe the type of scattering which, indirectly we have being referring to since the beginning which the scattering of a plane harmonic wave.

Let's remind ourselves that we are not treating the case where the incoming particle would actually remotely interact with the boxed particle (a node is a cubic volume within which there is a local particle).

Here the incoming particle has a momentum p assumed to be large enough so that we can represent the phenomenon with a 3-D plane harmonic wave where the wave vector is $k = p / \hbar$. We use polar coordinate along the z axis.

The plane wave is

$$\varphi(k.r) = e^{ik.r} = e^{ikr\cos\vartheta}$$
$$\cos\vartheta = k.r / kr$$
$$p = \hbar k = \hbar k e_z$$

$$e^{ik.r} = e^{ikr\cos\vartheta} = \sum_{l=0}^{\infty} \varphi_l = \sum_{l=0}^{\infty} i^l (2l+1) j_l(kr) P_l(\cos\vartheta)$$

using

$$\varphi(k.r) = e^{ikr\cos\vartheta} = \sum_{l=0}^{\infty} (2l+1) i^l j_l(kr) P_l(\cos\vartheta)$$

So $\varphi(k.r)$ is decomposed into partial waves. In region X, which assumed to be sufficient distance from the center $r = 0$, the incoming plane wave can be described by a set of partial waves with angular momentum l and with a magnetic quantum number $m = 0$ we know we know

$$h_l^{(+)}(kr) = n_l(kr) + ij_l(kr)$$
$$= i\left[j_l(kr) - in_l(kr)\right] \qquad \text{spherical Hankel functions of 1}^{\text{st}} \text{ kind}$$
$$= i\sqrt{\frac{\pi}{2kr}} \; H^{(1)}_{l+1/2}(kr)$$

$$h_l^{(-)}(kr) = n_l(kr) - ij_l(kr)$$
$$= -i\left[j_l(kr) - in_l(kr)\right]$$
$$= -i\sqrt{\frac{\pi}{2kr}}\;H_{l+1/2}^{(2)}(kr)$$

which is the spherical *Hankel* functions of the 2nd kind. Inverting both *SHF* 1st and 2nd we get

$$j_l(kr) = \frac{1}{2i}\left[h_l^{(+)}(kr) - h_l^{(-)}(x)\right]$$

This inversion result presents the superposition of both incoming and outgoing spherical waves $\frac{1}{2i}h_l^{(-)}(kr)$ and $\frac{1}{2i}h_l^{(+)}(kr)$ in region N

$R_{lN}(k,r) = -\frac{1}{2i}h_l^{(-)}(kr)$ is the incoming radial wave.

$(2l+1)i^l$ in

$\varphi(k,r) = \sum_{l=0}^{\infty}(2l+1)i^l(kr)P_l(\cos\vartheta)$ is the weight factor. Not considering the weight factor we can get

$$R_l^{(-)}(kr) = \frac{1}{A_{lN} - iB_{lN}}R_l(k_1 r)$$

where

$$R_l(k,r) = \bullet R_{l1}(k_1,r) \quad , \quad 0 \le r < r_1 \qquad region\ 1$$

$$\bullet R_{l2}(k_2,r) \quad , \quad r_1 \le r < r_2 \qquad region\ L$$

.

.

.

$$\bullet R_{lN-1}(k_{N-1},r) \quad , \quad r_{N-2} \le r < r_{N-1} \qquad region\ N-1$$

$$\bullet R_{lN}(k_N,r) \quad , \quad r_{N-1} \le r \qquad region\ N$$

in region N

$$R_{lN}^{(-)}(k,r) = -\frac{1}{2}h_l^{(-)}(kr) + \frac{1}{2}S_l(k)h_l^{(-)}(kr)$$

where $R_l(k) = \dfrac{A_{lN} + iB_{lN}}{A_{lN} - iB_{lN}}$ is the Scattering Matrix element describing the l^{th} partial wave.

$S_l(k)$ is also the angular momentum projection of the *Scattering-Matrix* (*S-Matrix*)

also we have

$$R_{lN}^{(-)}(k,r) = j_l(kr) + \frac{1}{2i}(S_l(k) - 1)h_l^{(-)}(kr)$$

where $L_l(k) = \dfrac{1}{2i}(S_l(k) - 1)$ is the partial scattering amplitude $L_l(k)$ is used to evaluate the effect of the potential $V(r)$ on $j_l(kr)$ in $\varphi(k,r)$ which is the incoming plane wave.

The 3-D solution has the from

$$\varphi^{(-)}(k_1 r) = \sum_{l=0}^{\infty}\varphi_l(k_1 r)\sum_{l=0}^{\infty}(2l+1)i^l R_l^{(-)}(k_1 r)P_l(\cos\vartheta)$$

This latter form represents the superposition of two waves

Incoming 3-D: $\varphi^{(-)}(k_1 r) = e^{ik.r} + \eta(k_1 r)$

Scattered 3-D: $\eta(k_1 r) = \sum_{l=0}^{\infty} \eta_l(k_1 r)$

At level l the l^{th} scattered partial wave is

$\eta_l(k_1 r) = (2l+1)i^l \left[R_l^{(+)}(k_1 r) - j_l(kr)\right] P_l(\cos\vartheta)$

$\eta_{lN}(k_1 r) = (2l+1)i^l f(k) h_l^{(-)}(kr) P_l(\cos\vartheta)$ is the scattered partial wave in region N.

$L_l^{(k)}$ is again the <u>partial scattering amplitude</u> for distance kr larger than 1 within

region N $h_l^{(-)}(kr)$ is approaches $\frac{e^{ikr}}{kr}$ (asymptotically) we have also

$\eta_{lN}(k_1 r) = (2l+1)\frac{i^l}{k} f_l(k) \frac{e^{ikr}}{r} P_l(\cos\vartheta)$

using $\eta(k_1 r) = \sum_{l=0}^{\infty} \eta_l(k_1 r)$ and the representation of $\eta_{lN}(k_1 r)$ in region N. The total

scattered wave is

$$\begin{aligned} \eta(k_1 r) &= \sum_{l=0}^{\infty} \eta_l(k_1 r) \\ &= \sum_{l=0}^{\infty} (2l+1)i^l f_l(k) h_l^{(-)}(kr) P_l(\cos\vartheta) \end{aligned}$$

$r_{N-1} \le r$ here also. wen $kr >> 1$ in N $\eta(k_1 r)$ is about equal to $f(k_1\vartheta)\frac{e^{ikr}}{r}$

$L(k_1\vartheta)$ is the scattering amplitude we would get from $\eta(k_1 r)$ and it has the following

form:

$$j(k_1\vartheta) = \frac{1}{k}\sum_{l=0}^{\infty}(2l+1)i^l L_l(k) P_l(\cos\vartheta)$$

this latter helps find the modulation of the scattered spherical wave $\frac{e^{ikr}}{r}$ in the polar angle ϑ.

The potential would drive the particles out of their beam and their density can be expressed by

$$|\eta(k_1 r)|^2 = \frac{|f(k_1\vartheta)|^2}{r^2}$$

$$\begin{aligned}\Delta I &= |\eta(k_1 r)|^2 v \Delta a \\ &= |f(k_1\vartheta)|^2 v \frac{\Delta a}{r^2} \\ &= |f(k_1\vartheta)|^2 v \Delta\Omega\end{aligned}$$

ΔI is the current of particles going through a space Δa vertical to the beam at angle ϑ and distance r.

In $|\eta(k_1 r)|^2$ $\Delta\Omega$ represent the solid angle through which Δa is visible. Also

$\Delta I = |f(k_1\vartheta)|^2 \Delta\Omega j$ where $j = |e^{ik\cdot r}|^2 v$ (incident current density)

$\Delta I = \frac{d\delta}{d\Omega}\Delta\Omega j$ representing the differential scattering cross section

$\Rightarrow \frac{d\delta}{d\Omega} = \frac{\Delta I}{\Delta\Omega j} = |f(k_1\vartheta)|^2$ is the differential cross section for all particles with a momentum $p = \hbar k$ at the level of a scatter with a potential $V(r)$ again here

$$f(k_1\vartheta) = \frac{1}{k}\sum_{l=0}^{\infty}(2l+1)i^l f_l(k) P_l(\cos\vartheta)$$

ΔI $l(k_1\vartheta)$ integrated over the angle 4π shows

$$\delta_{total} = \int \frac{d\delta}{d\Omega} d\Omega = 2\pi \int_{-1}^{1} |f(k_1\vartheta)|^2 d\cos\vartheta$$

Also using the following originality of *Legendre* polynomials:

$$(1-(\cos\vartheta)^2)\frac{d^2P_l(\cos\vartheta)}{d(\cos\vartheta)^2}-2\cos\vartheta\frac{dP_l(\cos\vartheta)}{d(\cos\vartheta)}$$

$$+l(l+1)P_l(\cos\vartheta)=0 \qquad l=0,1,2,3...$$

where

$$P_0(\cos\vartheta)=1,P_1(\cos\vartheta)=\cos\vartheta,(l+1)P_{l+1}(\cos\vartheta)=(2l+1)\cos\vartheta P_l(\cos\vartheta)-lP_{l-1}(\cos\vartheta)$$

which is the recurrence rotation satisfying

$$\int_{-1}^{1}P_{l'}(\cos\vartheta)P_n(\cos\vartheta)d\cos\vartheta=0. \qquad l\neq n$$

$$=\frac{2}{2l+1}\delta_{l'l}$$

again using the expression for $f(k_1\vartheta)$

$$\delta_{total}=\frac{4\pi}{k^2}\sum_{L=0}^{\infty}(2L+1)|f_L(k)|^2=\sum_{L=0}^{\infty}\delta_L$$

so $\qquad \delta_L=\frac{4\pi}{k^2}(2L+1)|f_L(k)|^2$

So far we did not approach all those results with our ideas of the QCNN. But we need to describe the conditions under which those ideas can be implemented. For the scattering in 3-D we still need to lay down conditions related to scattering amplitude and phase, the unitarity of the *S-Matrix* and the use of the *Argond Diagram*.

We should refer you to the section about 1-D scattering.

1-3-2 Conservation of Probability, the *S-Matrix* and the *Argand Diagram*

The conservation of current was shown to influence the conservation of probability. In a spherically symmetric potential the elastic scattering for a particle would conserve

1- particle number

2- energy

3- angular momentum

Point 1 and 3 implies that the current density of the spherical waves of angular momentum l is conserved.

The current of the incoming spherical wave is reflected at the scattering point but its magnitude is conserved. In this case the complex scattering element of the *S-Matrix* (element noted S_l) determines the factor or a value common to both the incoming and outgoing spherical waves. In this case and especially in region X, S_l has the absolute value one. This is a very important result and it defines the unitarity relation for the scattering matrix element S_l noted:

$$S_l^* S_l = 1$$

because S_l can only be a complex phase factor we use the following rotation:

$$S_l = e^{2i\delta}$$

S_l is the scattering phase for the l^{th} partial wave.

$$S_l = e^{2i\delta} = \frac{A_{lX} + iB_{lX}}{A_{lX} - iB_{lX}}$$

and this expression of course has modules 1.

The phase is:

$$\cos\delta_l = \frac{A_{lX}}{\sqrt{A_{lX}^2 + B_{lX}^2}}$$

using the following form for the partial scattering amplitude

$f_l(k) / f_l(k) = \frac{1}{2i}(S_l(k) - 1)$ and using the complex phase factor

$$S_l = e^{2i\delta}$$

$L_l(k)$ can be written as

$$L_l(k) = e^{i\delta} \sin \delta_l$$

$$L_l(k) = \frac{1}{2i}(S_l(k) - 1)$$

and $S_l^* S_l = 1$

gives the following expression labeled the unitarity relation for the scattering amplitude

$$\operatorname{Im} f_l(k) = |f_l(k)|^2$$

also the following holds:

$$(\operatorname{Re} f_l(k))^2 + \left(\operatorname{Im} f_l(k) - \frac{1}{2} \right)^2 = \frac{1}{4}$$

This expression represents a circle where radius is $\frac{1}{2}$ way from the center with coordinate $(0, \frac{1}{2})$ in the same plane. The circle is called the Argand diagram.

1-4 Cellular Automata (CA)

1-4-1 Brief Introduction

Cellular Automata were first used by *J. Von Neumann*. His idea was to construct a reliable system from unreliable parts. The subject developed to more sophisticated processes like the game of Life by *John Conway*. CA are discrete space-time models. A set of units (cells) arranged in the Euclidian space with one or two dimensions spaces. The cells are also called sites.

According to some local rules (local to a given site), a cell will take a certain value (i.e. binary value 0 or 1) at a given time t, a value which depends on the value of neighboring cells. We talk about left/ right or north/south effects or even diagonal effects. The state of a given cell changes from time t to time $t - 1$ according to what happens to its adjacent cells.

In general, CA are dynamical systems within discrete space and time. The set of cells is a regular lattice and the states of the cells are synchronously updated according to a deterministic set of local rules. Cellular Automata techniques are applied to many fields such as biology to study clustering of cells and clustering of cell activity; CA were use to study crystal growth and in physics, they are used to study the interaction of different particles. These uses are of particular importance to the study described herein.

1-4-2 CA's relation to the model

In this work we should remember that the study of wave scattering with QED techniques and the determination of future and past behavior of wave scattering processes enforce the idea of the need of a system capable of not only studying the process of wave scattering but also reconstructing all traces of the neural system's behavior by considering the simple manifestation state of an individual neuron at a given time. The neural network is an ever evolving dynamic system (a dynamical one of course). From the field of CA we choose Invertible Cellular Automata (ICA) because they offer a way of not only investigating the consequences of wave scattering in the network but also its origin. We assume that our neural network model's global configuration at time *t* can be reconstructed using the rules by which a neuron detects a wave and scatters it out. It's a process which depends on the existence of a local potential variable which is itself affected dynamically by the state of the neuron (the neuron being an on/off switch, a dimmer and a detector/propagator). This shows the need for our model to be of an ICA type.

We are now faced with a question of decidability. How do we know that our model can be mapped to an ICA? It was shown in a paper by Toffoli and Margolus [37] titled '*Invertible Cellular Automata: A Review*'
"There is no effective procedure for deciding whether or not any arbitrary 2-dimensional cellular Automata given in terms of a local map is invertible". The paper credits J. Kori for the proof. This theorem can be generalized to any Cellular Automata.

We remind ourselves that CA were used to model wave scattering so we will not have any difficulty relating the two in this work. The difficulty lies in that while simulating wave scattering processes described before- reconstructing the global configuration at any time from various global maps of the CA. Again, the CA will be of great help in localizing cluster of activities among subNNs (set of neurons which is a subset of the whole model).

Assume the following:

A pattern consisting of two subpatterns was input to the network. The pattern was encoded as a packet of waves. These waves arrive at the input layer and propagate across a set of neurons. Scattering occurs, and if the two sub-patterns do not belong to the same category a divergence occurs represented by activity in two remote areas of the network, representing two subNNs. It is at the level of these areas that the scattered waves stop propagating. The two areas of activity should be detected by the CA at a given time, and the corresponding map represents the relations between these two sub-patterns (the relation here is that they were input together). It is known that ICA are information conserving. So we may need as many global maps as configurations per unit time. That may be a huge number. Memorizing all maps and retrieving the right one is very difficult (but not impossible).

If at each time interval a map is recorded then, because the system is a continuous dynamical active one, a way to map each configuration into the right CA map must be determined. Even if no external input exists the neural network is active (but less activity than in the "Awake state" when the network is subject to no external input).

Before we conclude let's remind ourselves of the objectives of this proposal.

2- Putting All Together

2-1 Quantum Cellular Neural Network (QCNN): An Introduction

The new theoretical model of an Artificial Neural Network (NN) is introduced. Again, the model is based on techniques from quantum mechanics (QM) especially from the field of Quantum Electro-Dynamic (QED) and the field of Cellular Automata (CA). The model, unlike any traditional neural network (at least the most successful ones inspired by methods from statistical mechanics) has many characteristics which would enable it to take advantage of new advances and results in constructing quantum devices. The ideas underlying the model are supported by well founded results in QED and CA. The physical feasibility of the model is very tied to successful and practical advances in quantum computation (hardware). QED provided us with methodologies from the scattering process, the process by which a free particle propagating through a set of potential in time would cause wave scattering along determined paths. Those paths are set of a discrete organization of points (nodes/units/cells) positioned within a 3-D lattice cubic.

At the level of each node a discrete value potential (step potential) would cause the scattering to behave in a certain way. The uncertainty in the scattering process is explored in time by the other characteristics of the model: the model being a 3-D CA. The nodes are at equal distance form each other and the distance is chosen according to those limits imposed by the successful quantum device built or promised to be built. In our case, the model is simulated by implementing each node on a different processor.

The communication between those nodes is controlled in time and space so to represent the 3-D cubic architecture and all associated quantum processes. Each node behaves as a detector (particle detection), a wave scatterer (output of wave accross one or more outgoing paths), an on/off switch (a potential equal or greater than zero) and a dimmer (when the potential is varied between zero and a higher fixed value). Another and very important aspect in the model is the presence of oscillators within each node. Oscillations within each node are triggered by an incoming scattering wave and the oscillation is maintained by a local energy. We consider zero dumping for those local oscillations because they represent a partial information necessary for future retrieval of the pattern stored through some past scattering. A local oscillation is coupled with other remote oscillations caused by the same scattering process. In our model we are using a single particle per node. We also consider photon creation at the level of one particle and its correlation with other photons created through the model in other remote nodes. In this model, the scattering process is carried out by a set of rules valid for every node in this model (the model being a 3-D CA). Those rules obey all the laws in QED.

In our model the scattering is space and time can be thought of as a sub-model for "short term memory" and the coupled oscillations process is a sub-model of "long term memory". Along a defined path, the scattering process causes a wave to move along the path and be affected by each potential at the level of the nodes present along those paths.

It's well established that in QED we can reconstruct back the wave value back along the same path. The simplest situation is to have a single scattering when the wave is propagating along a given path without split scattering (at a given node. in the path. the wave moves to one and only one adjacent through one outgoing arc).

A given pattern is assumed to be encoded as waves. Incoming particles (described by waves) would arrive at one "side" of the model, would interact with the nodes at that "side" and given the value of the potential (>=0) at that node. would cause the waves to scatter along various paths starting from the nodes at that "side". Pattern storage is done in a distributive way among a certain numbers of nodes. Short time pattern retrieval will be done backward through the same path. The process will start at the level where the scattering waves vanish back to the input "side". The establishment of the scattering matrix (S-matrix) for each path is critical and it's needed for the retrieval process. Unlike in a traditional neural network we don't look for attractors or fixed points along which the pattern is stored. The pattern can be retrieved entirely assuming we follow the scattering path backward. And unlike the summation techniques done along weighted paths to a given cell or unit in a traditional NN in this model the process of retrieving the pattern is done by recollecting all "partial information" along those paths. But let's make it clear that it's not a summation process. It does not mean that, given n nodes through which the scattering process happens, we will be able to reconstruct the pattern by accessing each node, and getting the local "partial information". The scattering process is more complex than that. It may be possible to reconstruct the entire pattern taking back fewer paths than all paths taken by propagating scattering waves.

The physical realization will of course depend on the actual construction of quantum devices which would replace our units. It should be noted that we just described what we labeled as "short memory" process: Encode the pattern among an ensemble of scattering waves. retrieve the pattern by reconstruction of the original wave using QED techniques. The question now is when we talk storage of a given pattern we also are speaking of "long term memory". In this case. the retrieval process follows different procedures. Let's remind ourselves again that nodes behave as harmonic oscillators. A local particle within the node (we assume we have only one) would react to an incoming scattered wave and. given the potential and the wave value at that moment in time. the particle reaction will be a set of forced oscillations which will be controlled and maintained by a local energy. Those local oscillations represent again a "partial information" belonging to the stored pattern. We are speaking of a dynamic storage here for if a local energy is modified the oscillation would be affected or dumped and the information "forgotten". But the information is never lost for if those oscillations are revived even by "chance" (or by a random local perturbation) and if this happens during the retrieval process then this revival process would enable those oscillations to contribute to the global process of retrieval. By global process we means "reaching out" all coupled oscillations.

Locally (at a local node) the reaction of a particle would cause the creation of stationary wave packets. We consider that the oscillation are "read out" by "reading" the corresponding maintained local stationary wave. A wave can be. at any moment. described by a set of partial waves. This may be one of the situations where a summation process can be helpful but not critically necessary as we noticed in our study. The "read out" of these coupled oscillations. or the process of retrieval can be triggered by a new scattering process caused by a new pattern. If the pattern does match a stored one a collective resonance would occur. The resonance would occur when a scattering wave arriving at a given node will match the local stationary wave.

2-2 The Model: Supporting Ideas and Theory

The following will be considered in the simulation process:

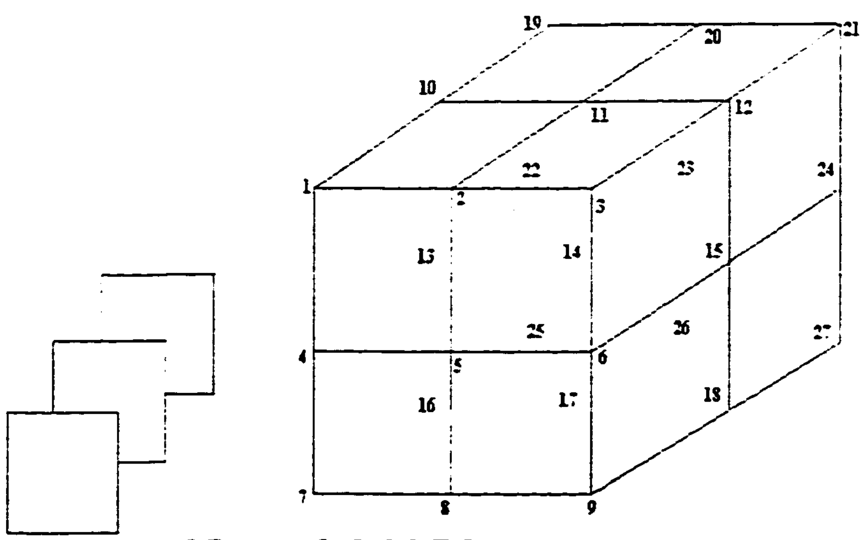
- Scattering waves
- Stationary waves
- Random path scattering
- Potential values at different lattice points (nodes)
- Local rules at the level of each node (rules of scattering and oscillations)
- Oscillations, interference and de-coherence

A description of a single node is necessary. As far as the architecture of a node is concerned, all nodes have the same structure.

A node is just a point in a 3-D lattice of nodes. But for this simulation a node is a networked processor connected to a certain number of adjacent (neighboring) nodes.

Illustration of a model implemented as a 3-D lattice of 27 nodes:

Figure 2-2-1 : 3X3X3 3-D Lattice



(3 Layers,3x3x3 3-D Lattice)

The node represents the following characteristics - depending on its location, a node can have:

* 3 adjacent nodes when the node is at one of the corner of the lattice.

* 4 adjacent nodes when the node is on one of the face but not at the corner

* 6 adjacent nodes when the node is inside (not located at any face)

Figure 2-2-2 : Computing Adjacent Nodes Ids for a 64-QCNN

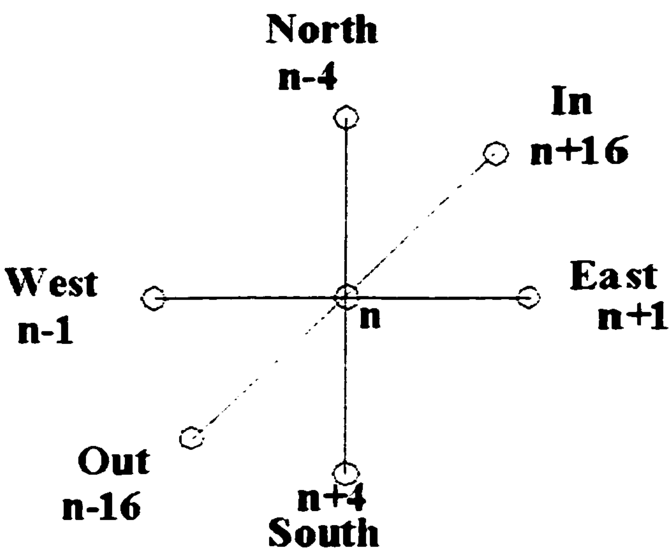
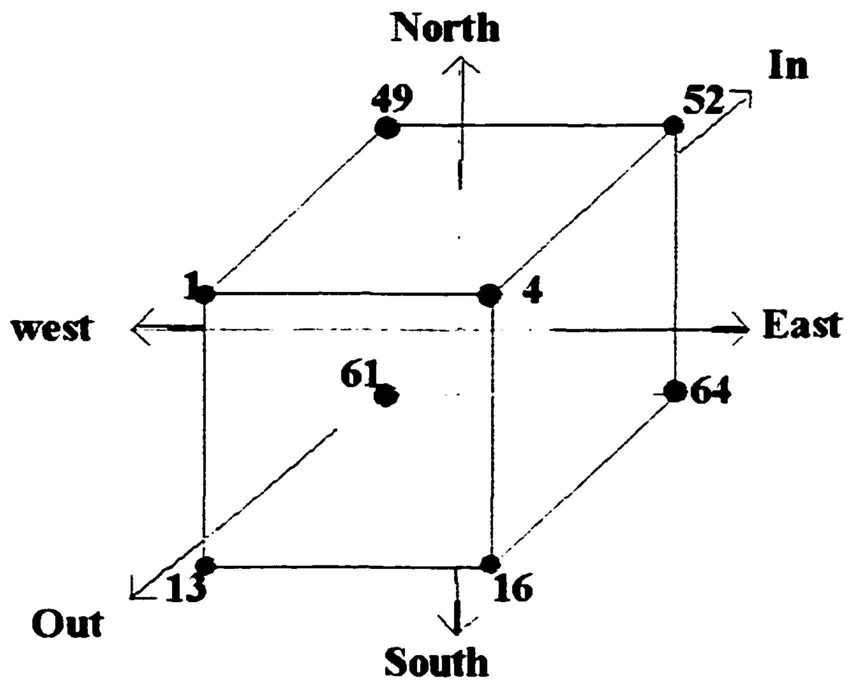


Figure 2-2-3: Directions' Relation to a 64-QCNN Model.

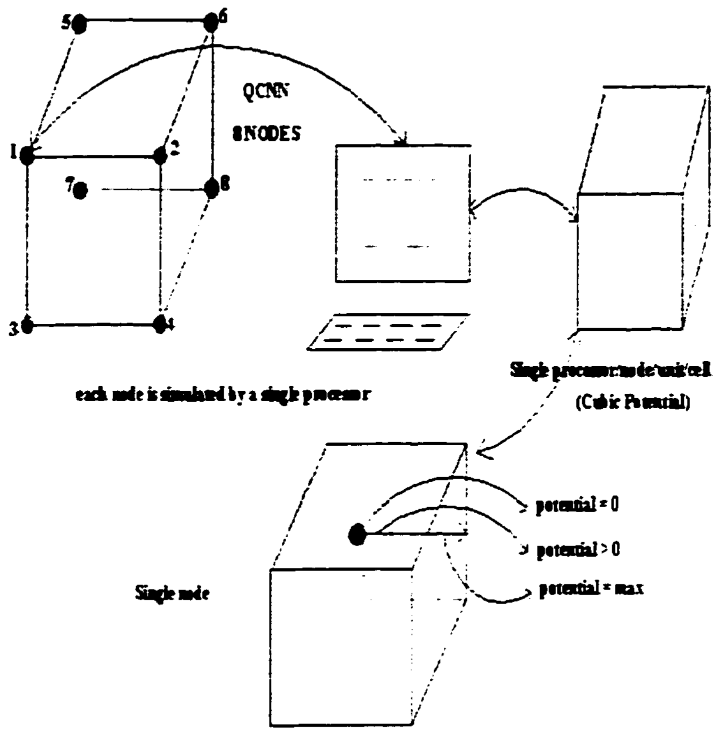


Each node again has full processing capability in this model. We are referring here to implementing each node on a separate CPU. In this simulation we don't need a big computation power. All we need is to be able to communicate data that result from a minor computation within each node. This communication process will be discussed in the section about the lattice being a set of clients/servers.

Let's first discuss the characteristics of a node.

- presence of a step potential within a cubic volume. The value of this potential would affect the scattering process(see figure 2-2-4):

Figure 2-2-4 : Node/Scatterer/Potential/Processor Mapping



- each node behaves as a harmonic oscillator. The number of particles oscillating within a node is important. But the main study considers only one particle within the node.

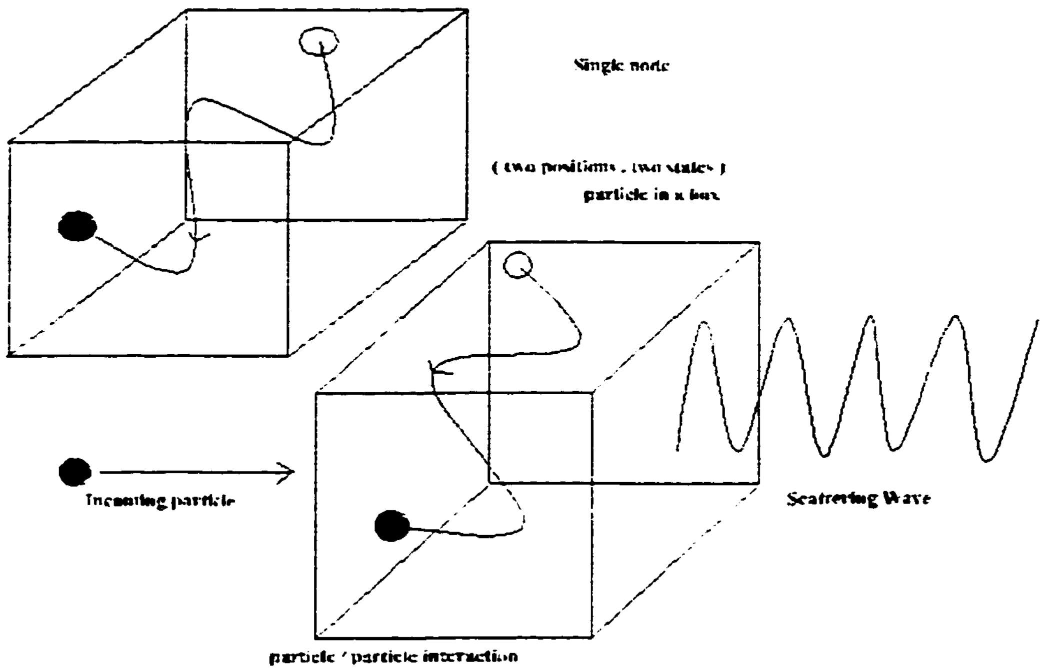
Modeling the scattering in this simulation locally within a node and globally across the whole network will be discussed.

The potential varies between 0 and a maximum value E. See bottom of figure 2-2-4

The next section will present two subsections, one about the functioning of one node. The next about the functioning of the node in relation with its direct neighbors (nodes adjacent to it)

- functioning of one node:

FIGURE 2-2-5 : Particle in a Box and Moving Particle

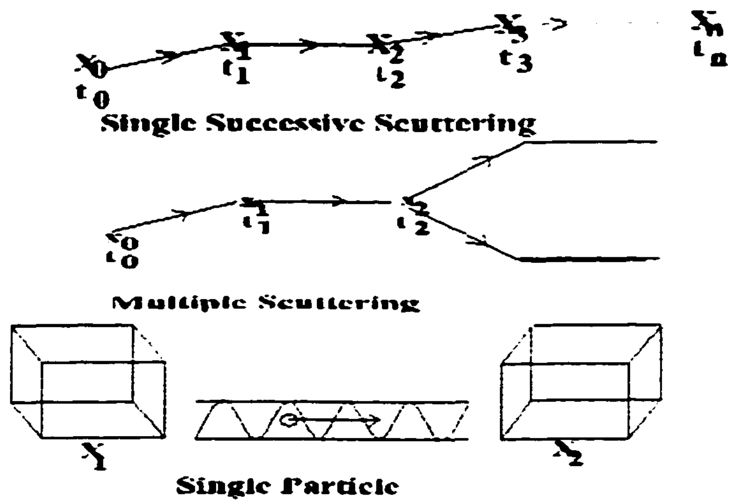


We discussed in the chapter about QED all the techniques necessary for this process. The capabilities of reconstructing the value of wave scattered at time t_1 at node x_1, and then at t_2 at x_2 by analysis of the S-matrix associated with a specific scattering path.

Note that because each node is also a harmonic oscillator we will be considering stationary wave.

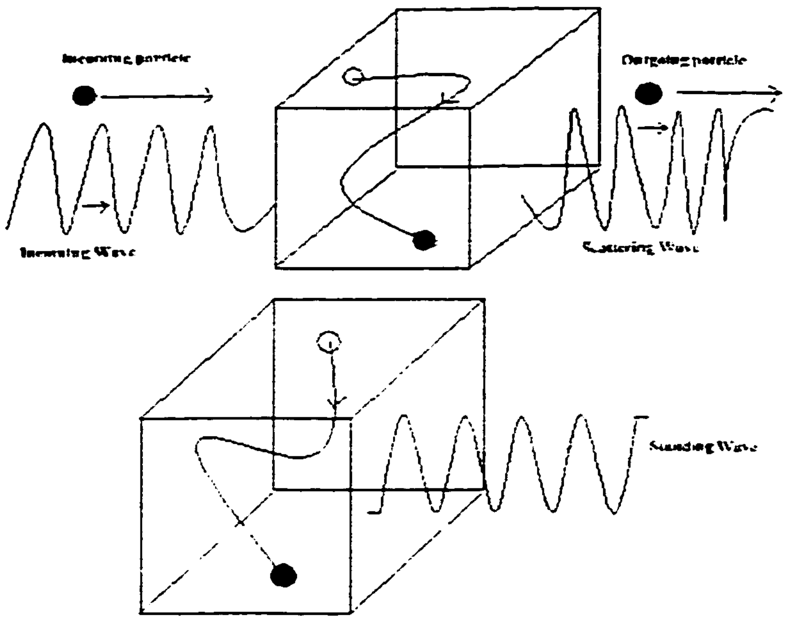
The scattering can be single or split. A scattering wave is associated with a moving particle between two adjacent nodes:

Figure 2-2-6 Single and Split Scattering



The following figure shows a moving particle with an associated descriptive wave and its interaction with a local particle:

Figure 2-2-7 : Particle/Particle Interaction



The scattering process will be developed from the behavior of a wave packet in 3-D. (from *'The picture Book of Quantum Mechanics'*) [9].

Let r be the position vector of a particle in the (x, y, z) coordinate.

$$r = (x, y, z) \qquad\qquad (P_x, P_y, P_z)$$

and let $P = (P_x, P_y, P_z)$

The vector operator for the momentum is

$$\hat{P} = (\hat{P}_x, \hat{P}_y, \hat{P}_z) = \frac{\hbar}{i}\left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z}\right)$$

where $\left(\frac{\partial}{\partial x},\frac{\partial}{\partial y},\frac{\partial}{\partial z}\right)$ is noted ∇ and is called the differential operator (nabla).

The 3-D stationary plane wave at a given node is

$$\varphi_p(r)=\frac{1}{(2\pi\hbar)^{1/2}}\exp(\frac{i}{\hbar}p_x x)x\frac{1}{(2\pi\hbar)^{1/2}}\exp(\frac{i}{\hbar}p_y y)$$
$$x\frac{1}{(2\pi\hbar)}\exp(\frac{i}{\hbar}p_z z)$$
$$=\frac{1}{(2\pi\hbar)^{3/2}}\exp(\frac{i}{\hbar}p\bullet r)$$
$$p\bullet r=P_x x+P_y y+P_z z$$
$$\frac{i}{\hbar}p\bullet r=\partial\Rightarrow p\bullet r=\frac{\hbar}{i}\partial$$
$$\frac{P}{\hbar}=\frac{\partial}{ir}=k$$

$$\frac{i}{\hbar}p\bullet r=\partial\Rightarrow p\bullet r=\frac{\hbar}{i}\partial$$

$$\frac{P}{\hbar}=\frac{\partial}{ir}=k \text{ wave vector}$$

∂ is the phase wave length $\lambda=\frac{2\pi}{|K|}$

The 3-D stationary wave is a simultaneous eigenfunction (solution) of the following 3 equations:

$$\hat{p}_x\varphi_p(r)=p_x\varphi_p(r).\hat{p}_y\varphi(r)=p_y\varphi_p(r).\hat{p}_z\varphi(r)=p_z\varphi_p(r)$$

$p_x.P_y.P_z$ are momentum eigenvalues of the wave $\varphi_p(r)$.

The 3-D time-dependent wave function is

$$\Psi(r.t)=\frac{1}{(2\pi\hbar)^{3/2}}\exp(-\frac{i}{\hbar}Et)\exp(\frac{i}{\hbar}p\bullet r)$$

where $$E = \frac{p^2}{2M} = \frac{1}{2M}(p_x^2 + p_y^2 + p_z^2)$$

M : mass of the particle.

$$\Psi_p(r.t) = \varphi_p(r) x \exp(-\frac{i}{\hbar} Et)$$

$$\Psi_p = (r,t) = \Psi_{p_x}(x.t)\Psi_{p_y}(y.t)\Psi_{p_z}(z.t)$$

So the time-dependent wave in a 3-D is actually equal to the 1-D wave along each axis.

Also. the wave can be described by a superposition of plane waves with a spectral function

$f(p) = f_x(p_x) f_y(p_y) f_z(p_z)$ product of 3 Gaussian spectral functions in here:

$$f_{x,y or z}(p_{x,y,z}) = -\frac{1}{(2\pi)^{1/4}\sqrt{\delta_{p_{x,y,z}}}} \exp\left[-\frac{(P_{x,y,z} - P_{x_0,y_0,z_0})^2}{4\delta^2_{p_{x,y,z}}}\right]$$

P_{x_0}. P_{y_0}. *and* P_{z_0} = P_0 are the expectation values δ_{p_x}. δ_{p_y} *and* δ_{p_z}

$$\Psi(t.r) = \int f(p)\Psi_p(r - r_0.t) d^3 p$$

$\Psi_p(r - r_0.t)$ is a superposition of functions $\Psi(t.r)$ is the wave packet which starts at $t = 0$ at the point r_0 with an average momentum P_0.

In our work at the level of a given node we consider the probability distribution $|\Psi(x.y.0.t)|^2$ in the $x.y$ - plane

Considering time dependent equations here is not that important for we will be talking about stationary waves whose value will be maintained by a local energy.

Those stationary waves are in synchrony with waves existing at all nodes and corresponding to the same input.

We will be considering here an input which is a superposition of multiple pattern.

In this section we will consider among other characteristics of scattering processes the following :

- Scattering wave as the sum over partial waves.
- Scattering amplitude as the sum over partial amplitudes.

First let's present a partial-wave decomposition of plane wave.

At a given time the wave is stationary and has momentum $p = \hbar k$. In polar coordinate with k along the z axis we have:

$$e^{ip.r/\hbar} = e^{ik.r} = e^{ikr\cos v}$$

$$e^{ik.r} = e^{ikr\cos v} = \sum_{l=0}^{\infty} \varphi l$$

$$= \sum_{l=0}^{\infty} i^l (2l+1) j_l(kr) P_l(\cos v)$$

$P_l(\cos v)$: *Legender* polynomial of order l
r : Radius vector
k : Wave vector of plane wave
$\cos v = k \cdot r / kr$: Cosine of polar angle.
$xxx(kr)$: Spherical Bessel function of first kind.

That represents the decomposition of wave into partial waves.

For the scattering process. a given particle would have a momentum p and can be described by a 3-D plane harmonic wave where the wave vector $k = p/\hbar$

$p = \hbar k = \hbar k e_z$ k

The plane has the following form:

$\varphi(k,r)e^{ik.r} = e^{ikz} = e^{ikr\cos v}$

The decomposition process of the wave into partial waves shows that the incoming wave can be described by a group of partial waves with an angular momentum l and a quantum number $m = 0$. Note that in the section about QED we covered all theoretical foundations for the scattering process just to prove the theoretical feasibility of the model. In here we are going to present more equations dealing with other aspects of the wave at a node level.

Let's represent the radial wave function

$j_l(kr)$

$$j_l(kr) = \frac{1}{2i}\left[h_l^{-}(kr) - h_l^{(-)}(kr)\right]$$

incoming wave: $\frac{1}{2i} h_l^{(-)}(kr)$ spherical wave

outgoing wave: $\frac{1}{2i} h_l^{(-)}(kr)$ spherical wave

$$R_{lN}^{m}(k,r) = \frac{1}{2i} h_l^{(-)}(kr)$$

This latter is the incoming radial wave in region N.

Let's assume in our case that the potential at a given node has step values (step potential). What we are saying here is that the potential at a node level. when it's turned on can have discrete values increasing from 0 to a certain maximum value.. Remember that the information is carried via waves encoding and the wave scattering across the node of the network will cause their specific influence at a node level to be recorded.

2-3 A Theoretical Implementation

We covered in the last section the implementation issues of the QCNN as a set of interconnected nodes within a networked set of processors.

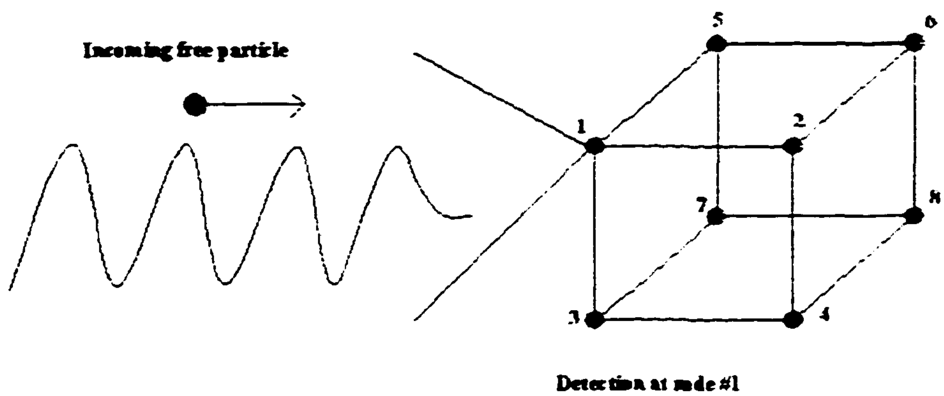
In the following section we will cover the most important aspect of the model. Those aspects consist of the simulation of the scattering process in time. the stationary scattering. problems of coherence resonance. collective oscillation and their control.

The last processes depend on well established result in the QED in particular and Quantum Mechanics in general. The choice of the right rules in the CA is critical because those rules should have as a consequence a behavior similar to actual real processes through the use of real quantum devices.

In this section we must first cover the characteristic of a node being an on/off switch. a wave scatter and a wave detector. The other main characteristic is for the node to be a harmonic oscillator. Because our work was inspired by progress in modeling biological neurons system. a survey of those developments will be given and will be related to our model. That part will show the potential for our model for simulating many of the brain natural processes.

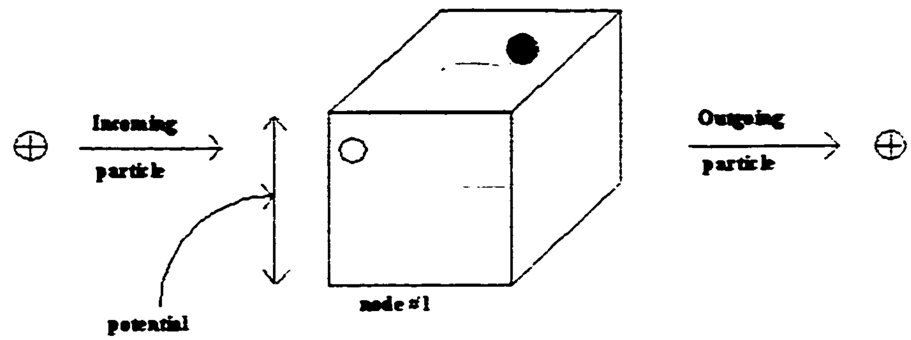
The node as a detector:

Figure 2-3-1 : A Node as a detector



We assume first that the incoming particles together with its wave encode a specific pattern. When the particle attains a given scattering center (here a given node) it would interact with the local particle within that scattering center and also the potential at that moment of time. The wave in this case is being 'detected'. Because we are not dealing with real particles in our system we can call them virtual or quasi particles. The node can be thought of as a quantum device where a single particle is being entangled. Figure 2-3-2 illustrate the presence of a single particle, a potential and a moving particle:

Figure 2-3-2 : Potential at the node's level and interaction with a moving particle

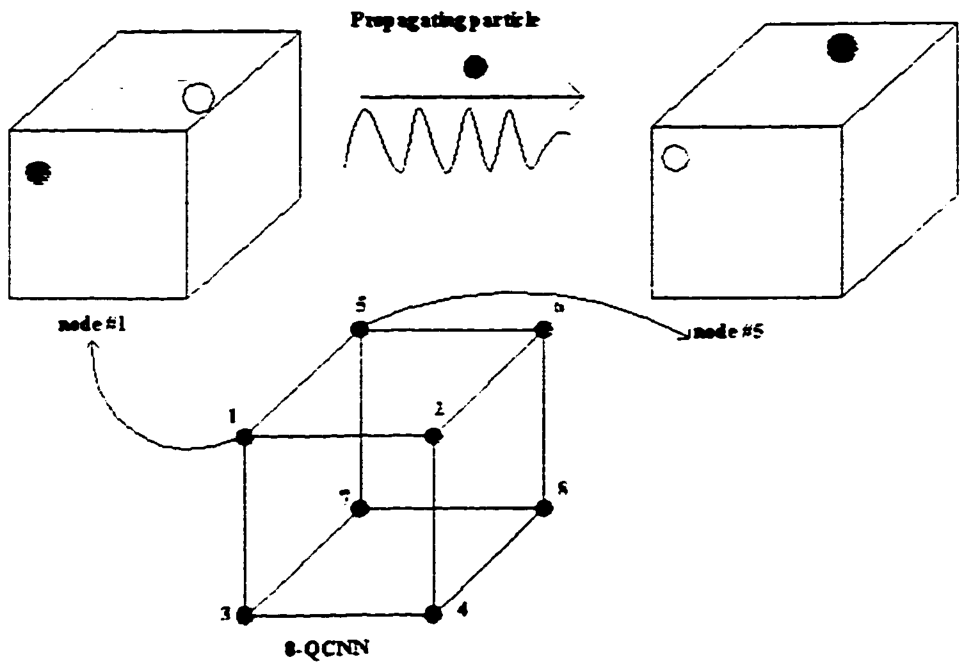


In this figure the local particle's state will be affected by the incoming particle and would behave accordingly (according to the detected wave and the local potential). Unlike in Quantum many body problems the scattering centers are not randomly located but are at equal distance from each other. Also only 1 particle to 1 particle interaction is considered which makes the model much simpler.

Now the question is how this is being implemented?

In our simulation a propagating particle being described by a wave is simulated by a data packet send by the controller to the nodes. The data contain numeric values corresponding to a particular wave packet. When the packet is being received (detection) by a given node (processor), those values will be used with the local ones (representing the values of stationary waves) and based on the local potential a new data packet representing the scattering wave will be sent to other adjacent nodes in a 'controlled random fashion'. For the first time the controller is playing the role of source of pattern input to the model. So, in this case the controller is called the 'initial propagator'. Actually all nodes are quantum propagators. Let's illustrate the problem with two adjacent nodes and explain the implementation:

Figure 2-3-3 : Exploded View of 2 Adjacent Nodes



Within each node we assume the process is time-independent but detection is time dependent.

In our network assume a packet (equivalent to the wave values corresponding to a given particle) arrives at a given node (x_1) at time t_1 then propagate to another node (x_2) and arrives at time t_2.

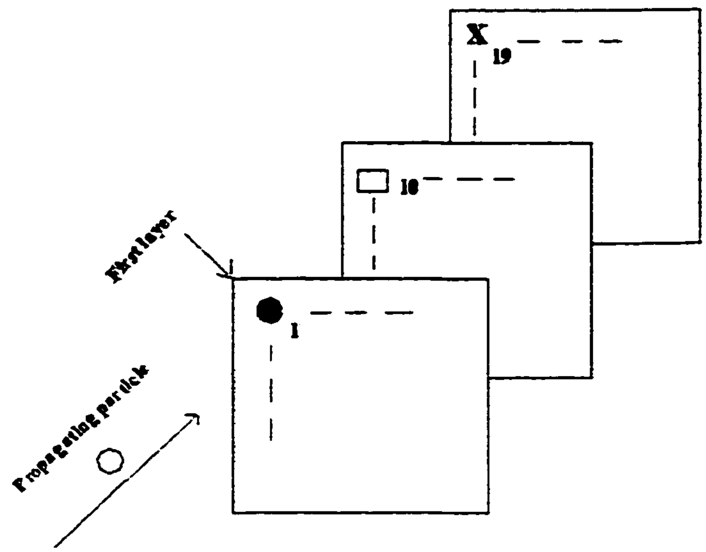
Note that the only interaction allowed is between the propagating particle and the local particle and that there is no direct interaction between particles of adjacent nodes. In this case, all propagations are very controlled within the CA by the CA rules.

Going back to the Green function we can state that

$$iG(x_2, x_1, t_2 - t_1) = iG^{-}(x_2, x_1, t_2 - t_1)$$

is the probability amplitude for the presence of an extra particle within the scattering space x_1 at time t_1 and its presence at x_2 at time t_2. The system being composed of the particles at x_1 and at x_2 and the propagating particle. We have to choose the first layer of nodes through which an incoming particle would propagate in the system. We can illustrate that for our 8-QCNN model:

FIGURE 2-3-4 : The 27-QCNN as a 3 Layers Model



Remember that we introduced all necessary theory in the chapter about QED. Note that a node in itself is a cubic volume and that the local potential exhibit step values within that volume. That feature will help in the normalization process. Let's start with scattering in one-dimension which will be generalized to the scattering in 3-dimension.

The data structure for a given node would contain value for the following parameter energy. momentum. position. time.

The incoming particles incident to a given node x is described by a Ganssian wave packet of spatial width δx.

So at time $t = 0$ and origin x_0 and momentum P_0 we have in general

$$\Psi(x,t) = \int w(p) e^{-iEt/\hbar} \varphi(E(p), x) dp$$

where $\varphi(E(p), x)$ is the continuum eigen-function. $w(p)$ is the Gaussian spectral function

$$w(p) = \frac{1}{(2\pi)^{1/4}\sqrt{\delta_p}} \exp\left(-\frac{(p-p_0)^2}{4\delta_p^2} - \frac{ipR_0}{\hbar} \right)$$

where $E = p^2/2m$ is the energy eigen-value

p = momentum

x = position

t = time

P_0 = expected value of the momentum at x_0

x_0 = initial location

$\delta_p = \hbar / 2\delta_{x_0}$ = momentum width of incident wave packet

$\varphi(E, x)$ = stationary scattering wave

Note that $\varphi(E, x)$ represent those waves created within the node due to the interaction of the incoming particle and the local particle.

Actually in quantum mechanics we speak of a right moving and left moving wave packet:

$$\Psi_{j}(x,t) = \int w(p) e^{-iEt/\hbar} \varphi_{j}(E(p), x) dp$$

$$\Psi_{-}(x,t) = \int w(p) e^{-iEt/\hbar} \varphi_{j-}(E(p), x) dp$$

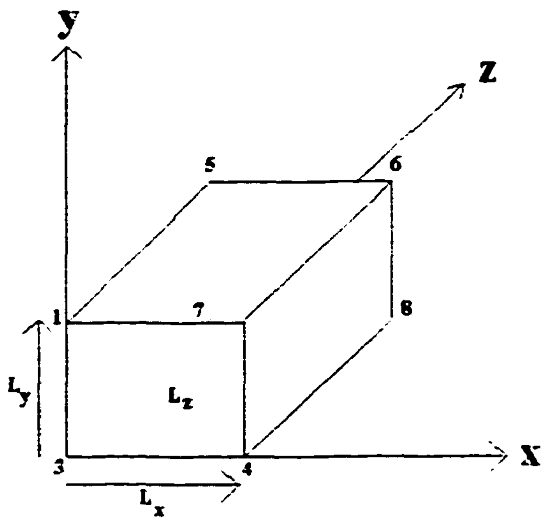
The above computations are developed in [8, 9].

3- Model Architecture

3-1 Building the Model

The following figure shows the process of building the model by tiling unit 3-D lattices starting from the origin of a 3-D Cartesian system.

Figure 3-1-1 : Tiling Unit Cubes



Unit cube
1,2,3,4,5,6,7,8, are the nodes/units/cells/processors

The following list shows the directions of tiling starting from a specific node:

$1:(0,l_y,0)$

$2:(l_x,l_y,0)$

$3:(0,0,0)$

$4:(l_x,0,0)$

$5:(0,l_y,l_z)$ 1,2,3,4,5,6,7,8 are the nodes/units/cells/processors

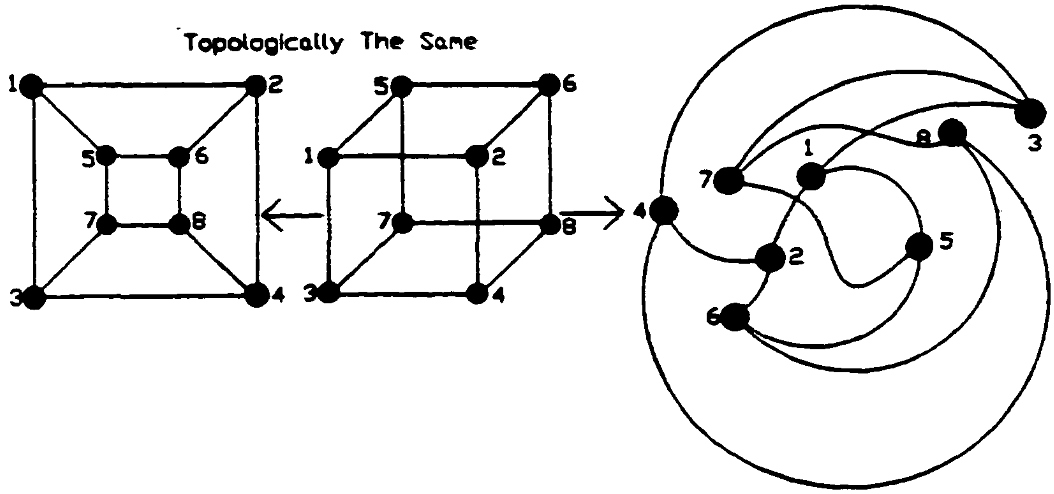
$6:(l_x,l_y,l_z)$

$7:(0,0,l_z)$

$8:(l_x,0,l_z)$

Figure 3-1-2 shows that adjacency is more important than distances between nodes. In the simulation, the situation is supported by the fact that the processor geographical location is not important as far as the architecture of the model is concerned. The 3 forms in the figure are topologically equivalent.

Figure 3-1-2 : 3 Topologically Equivalent Views of the Model



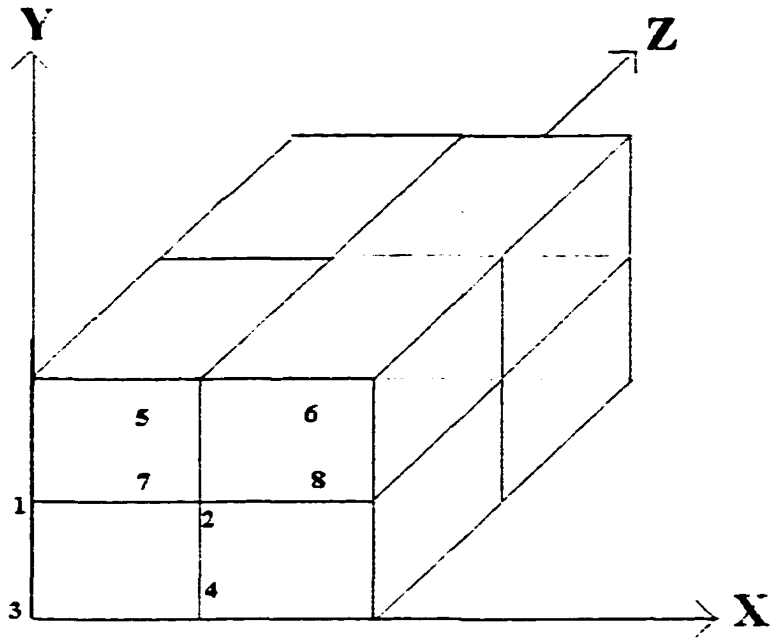
The following list presents what we will call from now on "tuplets". They are local adjacencies. Each tuplet present a node and the nodes adjacent to it.

(1:2.3.5)
(2:1.4.6)
(3:1.4.7)
(4:2.3.8)
(5:1.6.7)
(6:2.5.8)
(7:3.5.8)
(8:4.6.7)

Connections tuplets for the unit cubic lattice

Figure 3-1-3 shows the process of tiling a higher cubic volume with a set of unit cubes.

Figure 3-1-3 : Tiling to a Higher Model



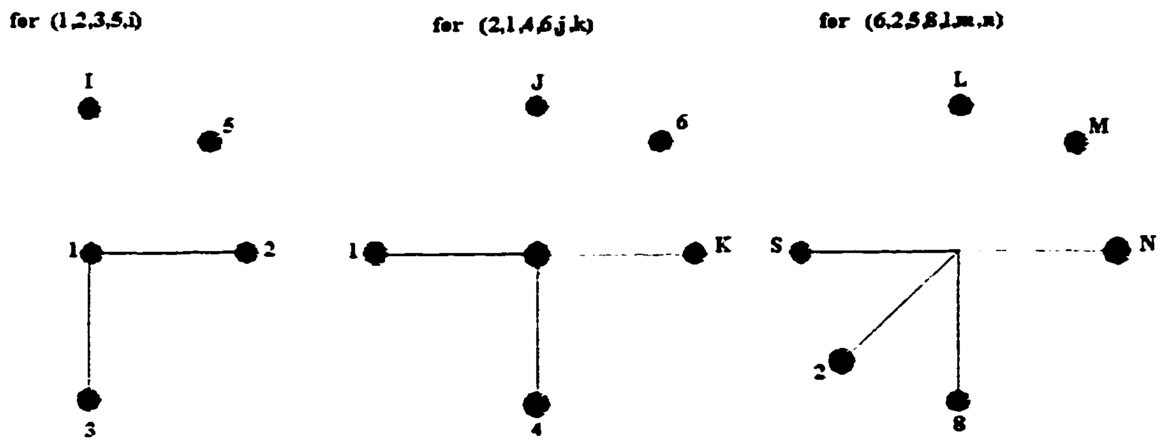
From 2x2x2 to a 3x3x3 and by tiling starting from the unit cube at the (x, y, z) origin, we would like to generate the connections tuplets in a way not to disturb the original relation between the ids for the nodes and conserve the same algorithm. We assume here the unit cube is our starting state and that the connections are set accordingly. The tuplet (1:2,3,5) is now (1;2,3,5, i) i is unknown until we assign an id to the corresponding processor.

The tuplet (2;1,4,6) is now $(2;1,4,6,j,k)$. J and k are additional nodes whose ids are unknown.

For the tuplet (6;2,5,8) we have 3 more additional nodes $(6;2,5,8,l,m,n)$

Figure 3-1-4 shows the process of computing the ids for all adjacent nodes given a specific nodes. As we can notice, the number of adjacent nodes depends on the location of the given node.

Figure 3-1-4 : Node's Adjacency Diagram



3-2 The Model's Special Characteristics

In general if the id assignment is done after tiling, generating those ids is easy. But if we need to conserve the original ids and generate the remaining one dynamically as connections and nodes are added, the problem become more complicated. The final table containing all new connections tuplets is now fixed and should be generated by simple computation each time we need it. Assume we start with an *mxmxm* 3-D cubic lattice, assume we go to a higher *nxnxn* lattice and assume we do so by using the $2x2x2$ unit lattices as the tiling unit. Then we would like, after getting a complete table of all new connections tuplet, to be able to duplicate it dynamically.

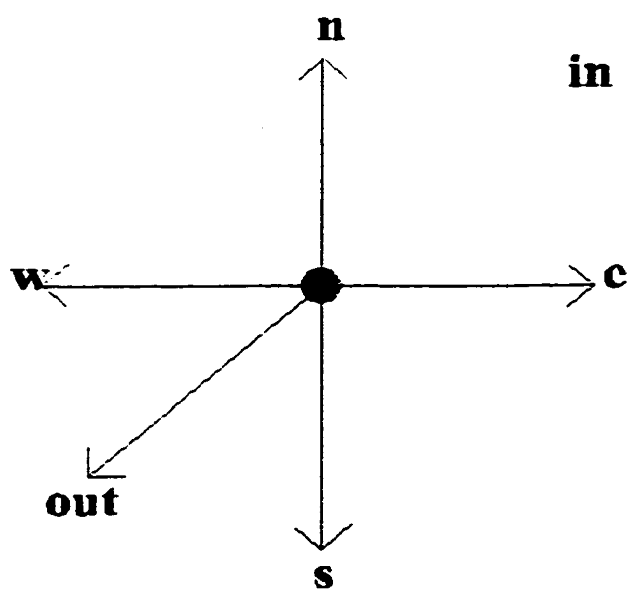
The main problem here is to tile and "de-tile" dynamically (a lattice which grows and shrinks dynamically to a higher or lower 3-D lattice).

When the size of the lattice is fixed at start the problem is much easier. We can generate ids by following the following rules.

First let's describe the identification process in a $3x3x3$ 3-D lattice:

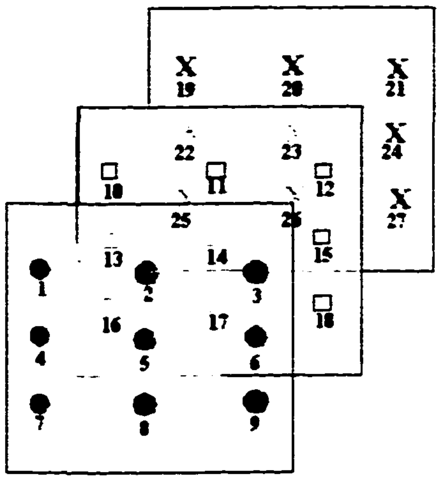
Figure 3-2-1 shows a all existing directions starting from a given node.

Figure 3-2-1 : Directions to Possible Adjacent Nodes



The following figure shows the 3 layers for a 27-QCNN. Each layer is represented by a 2-D lattice with 9 nodes.

Figure 3-2-2 : Nodes' Representation On 3 Layers' View



A 3-D Lattice

This 3-D lattice contains 27 nodes and so it contains 27 connections tuplets. The tuplets have the following format $(X{:}N,S,E,W,I,O)$ where :

X is the node id and *N, S, E, W, I, O* correspond to *North, South, East, West, In* and *Out* directions respectively.

A zero (0) means no connection on that direction.

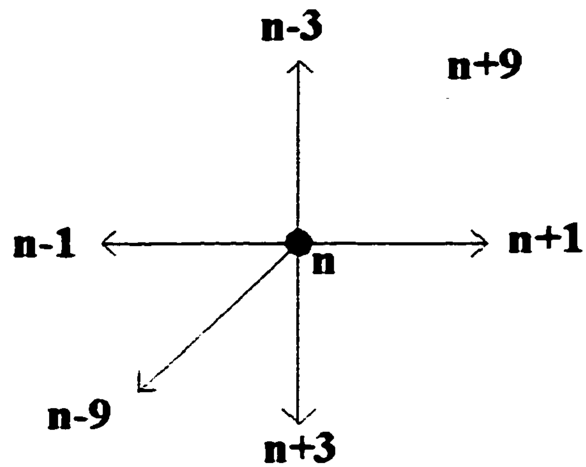
X	*N*	*S*	*E*	*W*	*I*	*O*
(1;	0,	4,	2,	0,	10,	0)
(2;	0,	5,	3,	1,	14,	0)
.						
.						
.						
(14:	11,	17,	15,	13,	23,	5)

27 such tuplets

When the 3-D lattice is known and all nodes are assigned an id starting from the top left node on the first layer and doing the same for each layer using the next unused integer from the set of positive integers $(1,....n)$ we can generalize the computation the following way:

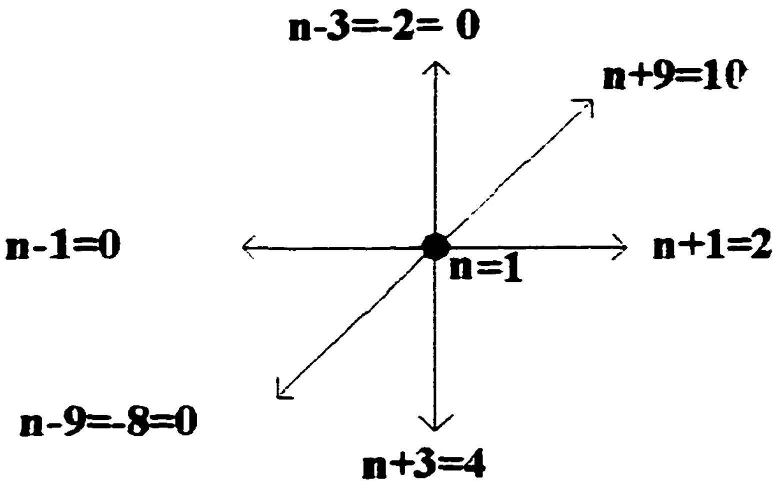
Using the 3-D lattice orientation diagram and replacing all nodes with their expected id. If n is the id of node x then figure 3-2-3 gives the formulas used to compute the ids of all adjacent nodes in all directions *(N, S, E, W, I, O)*:

Figure 3-2-3 : Computing Adjacent Nodes' Ids in a 27-QCNN



If $n = 1$ (corresponding to the first node on the first layer) then the orientation diagram can be represented by the following ids(see figure 3-2-4):

Figure 3-2-4 : Node 1 with Some Non-existing Adjacent Nodes

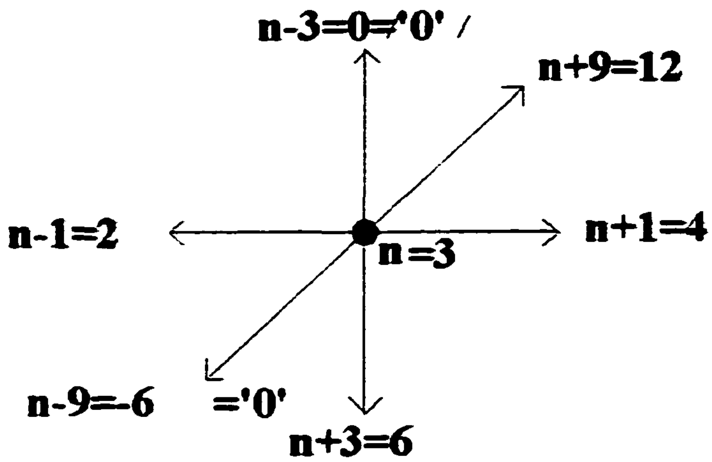


0 = no connection on that direction

Where '0' means no connection on that direction (if the computed value is 0 or negative then no connections on that direction.

What about a node with id $n = 3$?

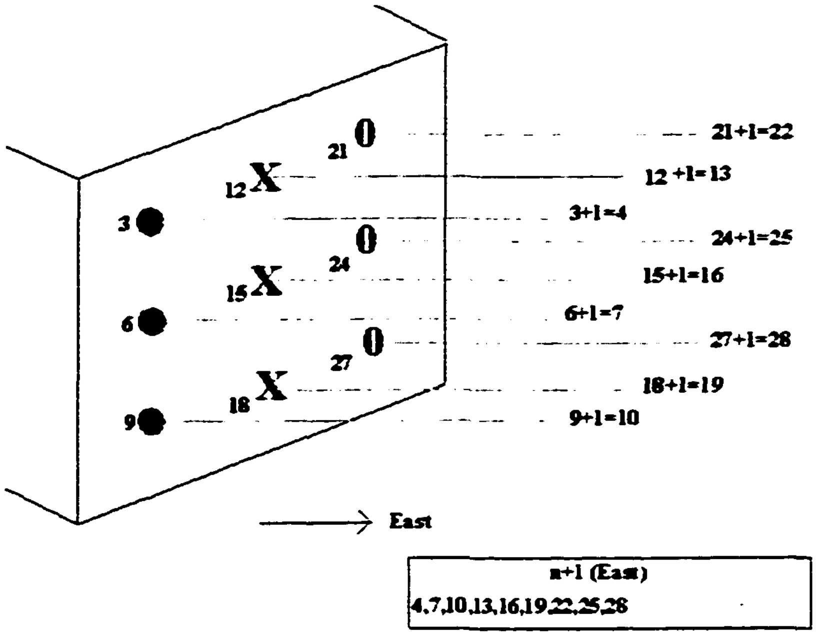
Figure 3-2-5 : Node 3 with Some Non-existing Adjacent Nodes



no connection : East (n+1=4), Out (n-9=-6='0')
North (n-3=0='0')

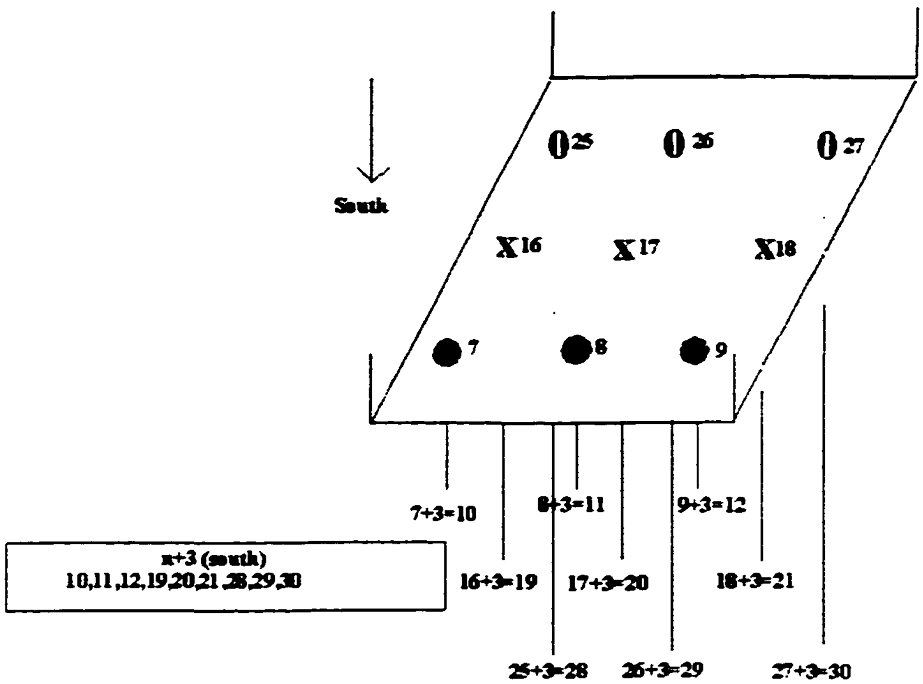
The computation for the *East* of $n = 3$ leads to 4 which is > 0 and 4 is the id of an existing node south of $n = 1$. All *East* computation of all node located on the eastern layer lead to eastern ids > 0.

Figure 3-2-6 : East Face with Possible Adjacent Nodes' Ids



All computed ids exist in the lattice except 8 which actually equal to the largest id (last node) plus 1. We face the same problem with the southern layer

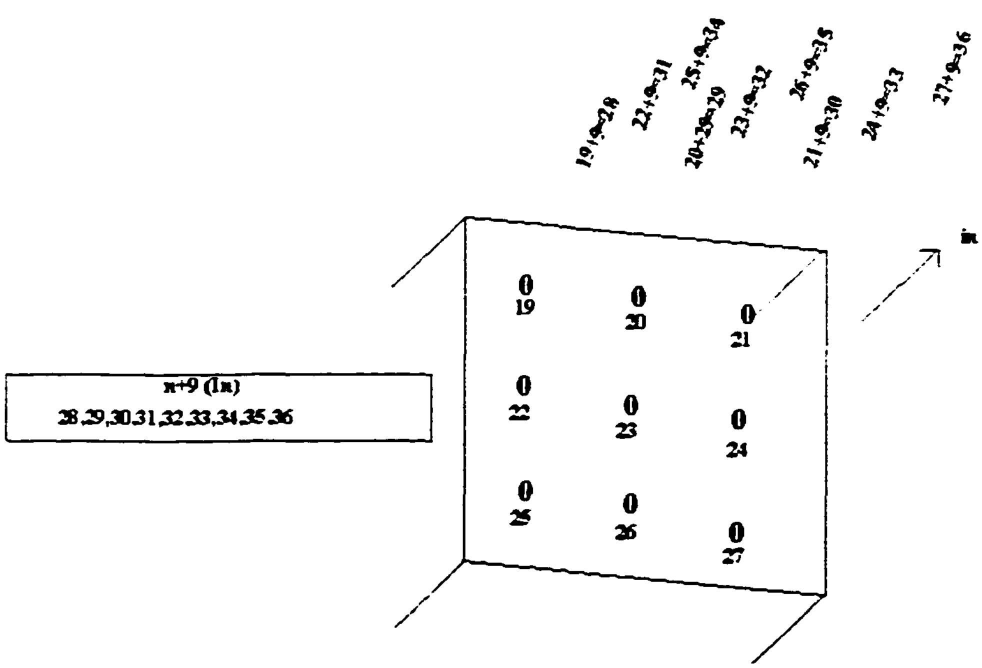
Figure 3-2-7 : South Face with Possible Adjacent Nodes' Ids



28. 29. 30 are ids of non-existing nodes. All other computed nodes exist already.

Let's do the same for the most inside layer.

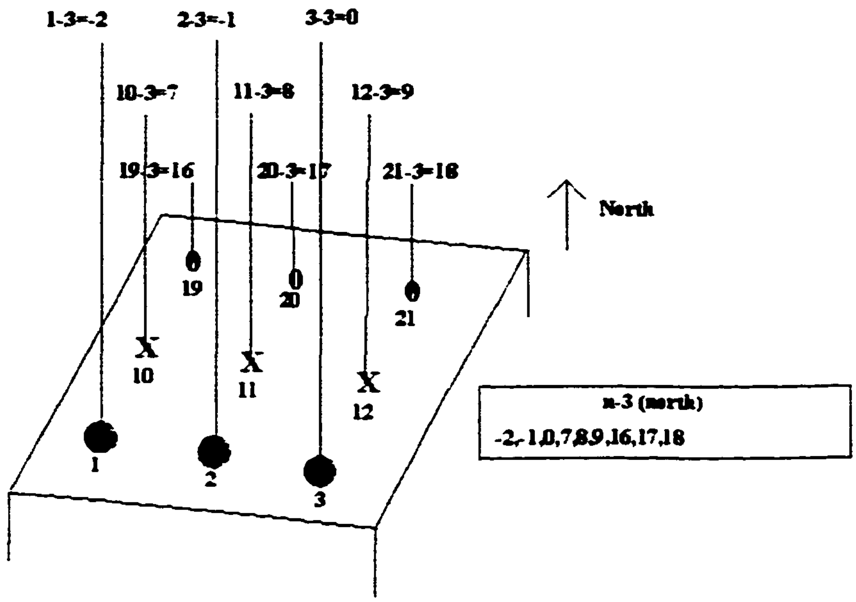
Figure 3-2-8 : In Face with Possible Adjacent Nodes' Ids



In this case all computed ids are > 27 and they are equivalent to '0' or no connection on that direction.

For the northern layer or face:

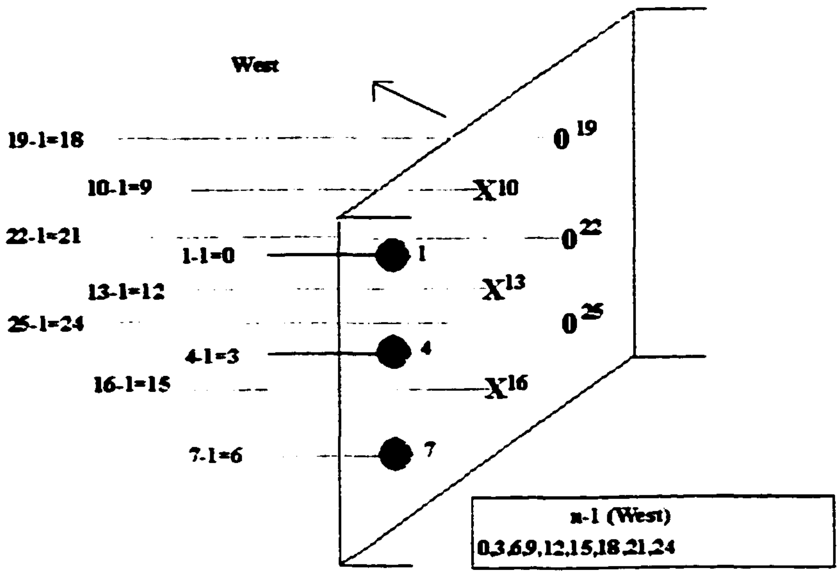
Figure 3-2-9 : North Face with Possible Adjacent Nodes' Ids



3 ids here are candidate for elimination based on the value computed (< = 0) . The others are within the right interval ([1.27]) but there may be no connection between nodes on the northern face and 7, 8, 9, 16, 17 and 18.

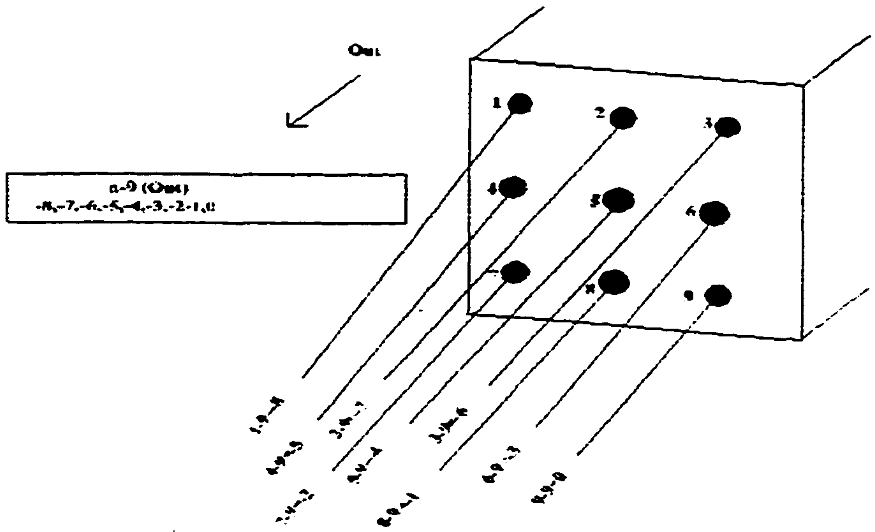
Now let's do the western layer.

Figure 3-2-10 : West Face with Possible Adjacent Nodes' Ids



And last the outside layer:

Figure 3-2-11 : Out Face with Possible Adjacent Nodes' Ids



All computed ids are $<=0$. This case is easy because all computed ids represent non-existing connections.

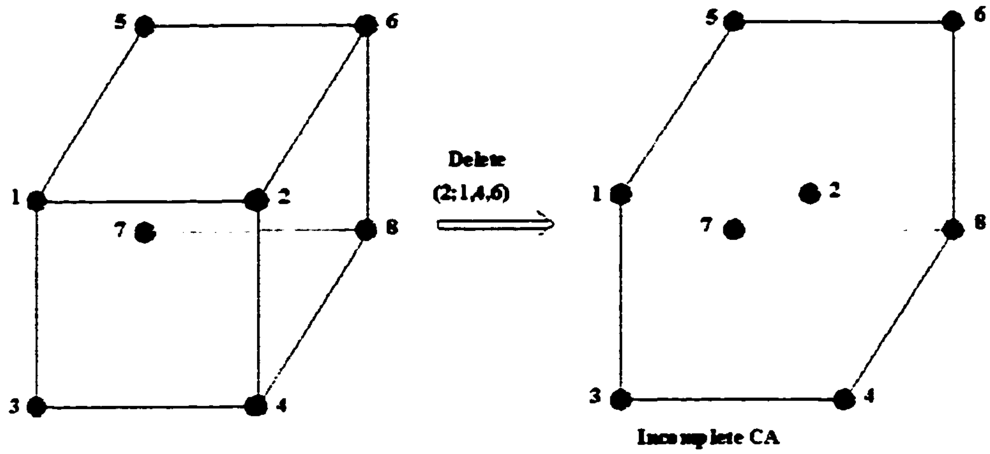
If that architecture is fixed at start then we can assign ids to each node based on the location of the node. Note that nodes not located on any face are much easier to find the adjacent nodes' ids for.

The problem is ids' assignments when the 3-D lattice grows to a higher 3-D lattice. Adding unit 3-D lattice to the existing one must lead to another 3-D lattice. And in the same way "deleting" units (actually we delete or we force into idle status some connection which would "delete" a given unit 3-D lattice) must also lead to a lower 3-D lattice.

Example of deleting a unit 3-D lattice:

Figure 3-2-12 shows the case when node 2 is "deleted". By "deleted" we mean the node is no longer functional. Node 2 being idle would cause all adjacent nodes to stop communicating with it.

Figure 3-2-12 : Example of 'Deleting' a Node in the Model



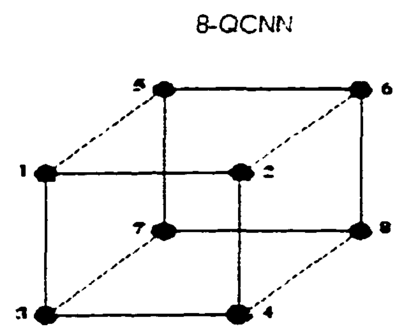
4- Model Simulation

4-1 Hardware Consideration

4-1-1 A Simple 8 Nodes Model(8-QCNN)

Figure 4-1-1 shows a geometric representation of an 8-QCNN. The figure presents the relation between all nodes.

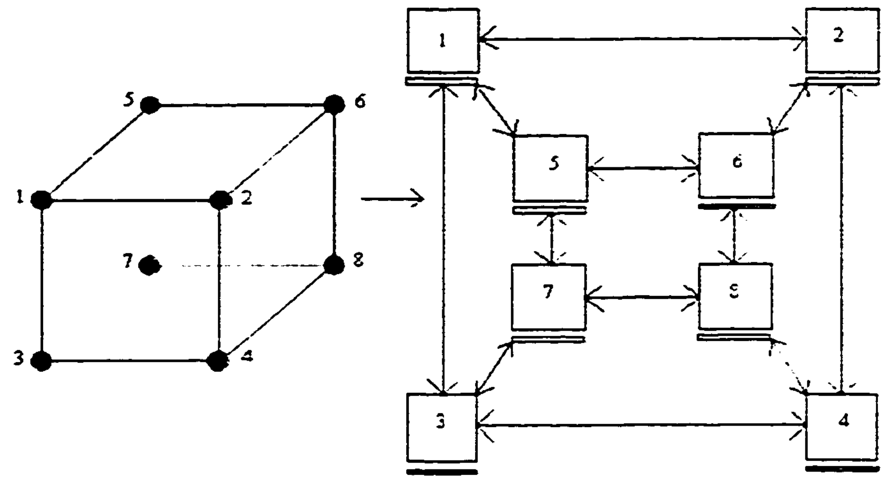
Figure 4-1-1 : A Geometric Representation of an 8-QCNN



In this chapter and throughout the whole thesis a 'Tuplet' is a set of nodes where the first node is the main unit and all nodes after ';' are its adjacent nodes.

Each node will be mapped to a single processor (see figure 4-1-2). The set of 8 processors are networked via a simple TCP/IP and the communication within a communication Tuplet is bi-directional. Later we will discuss a critical process where the graph being a 3-D CA would necessitate a process or remote control after it's implemented on a set of processors. In this latter case an extra processor will be needed. It will be called 'controller'.

Figure 4-1-2 : Mapping 8-QCNN to 8 Processors



8-QCNN mapped onto 8 processors

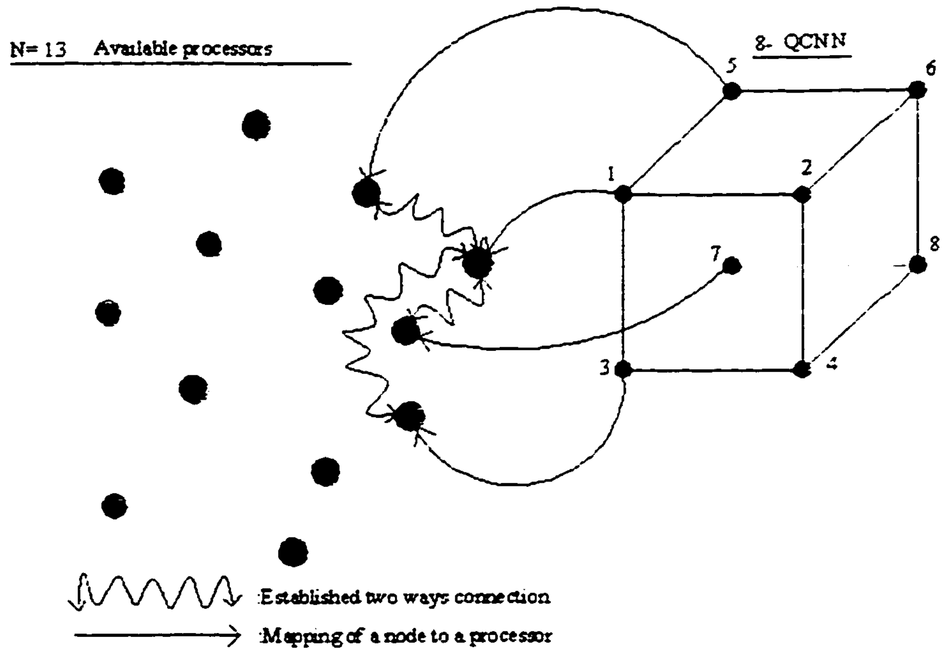
As was described before. the ids' assignment is done dynamically and in this case each processor will be given the same id as its corresponding virtual node. In the implementation an id will be dynamically mapped to an available IP address of an available processor in the network.

In general. given N available processor such that $N > M$ (M -nodes model) we would like to pick k processors out of N in a way to have $M^3 = k$.

With 8 nodes, any 8 processors out of N will do as long as the ids assignment or the choice of IP addresses and connections can be mapped to our 8-node QCNN. We will cover the case where a processor is not available and so an incomplete mapping would occur. We will study the model reliability and functioning in that case. Let's sketch the IP address assignment process. Let's remind ourselves that this will be done according to the computed connection tuplets. We have 8 such tuplets in this case.

Assume we have N available processors with N available IP addresses. We have to map 8 nodes of the 8-QCNN to 8 processors:

Figure 4-1-3 : Nodes' Mapping to a Pool of Available Processors



The established connections is done according to a pre-computed connection tuplet in our illustration.

Given a node N and the 6 direction North (N), South (S), East (E), West (W), In (I) and Out (O) the connection tuplet for x is $(x;N,S,E,W,I,O)$ where N, S, E W, I, O are the ids for all nodes directly connected to the node x.

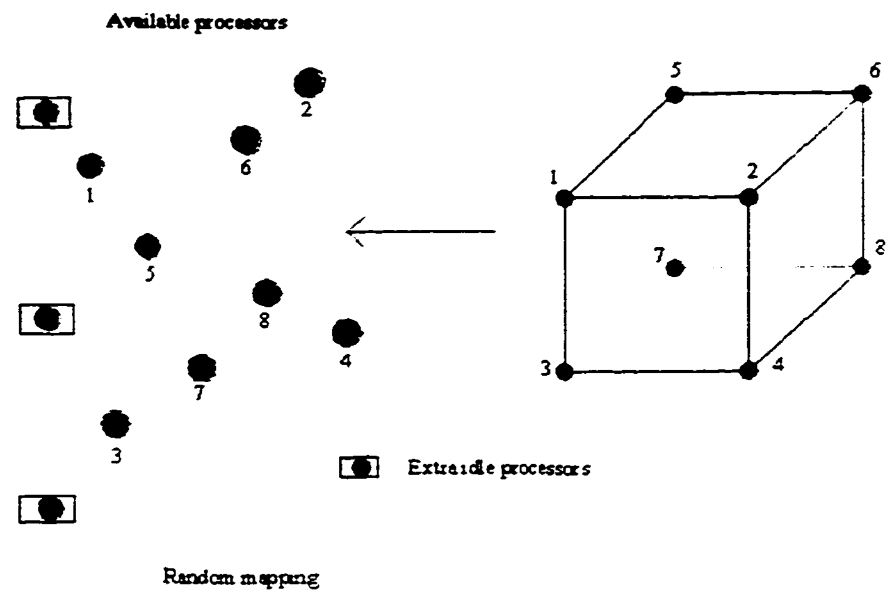
So for $x = 1$ (node with id 1) the connection tuplet is (1;0,3,2,0,5,0) where 0 means no connection on that direction.

4-1-2 Conclusion

Two situations arises here:

- we can first map the whole set of nodes from the QCNN into the available set of processors then proceed by establishing the TCP/IP connections according to all pre-computed connection tuplets.
- We can also map node id = 1 or any node to any available processor. Connections tuplet establish a partial set of TCP/IP connections. Both cases are similar because both would lead to a complete mapping. A more interesting case is when, during the life of the working model (real time communication between nodes), one or more processors become idle for whatever reason or there is a need to add up more nodes to dynamically increase the size of the QCNN to a higher one without disturbing the original one. This situation was described in the previousely. It will be assumed here that a QCNN configuration (architecture, connection, data, ... etc.) will be freezed until the higher order mapping is completed. We will talk about this again when we introduce the software side of the implementation.

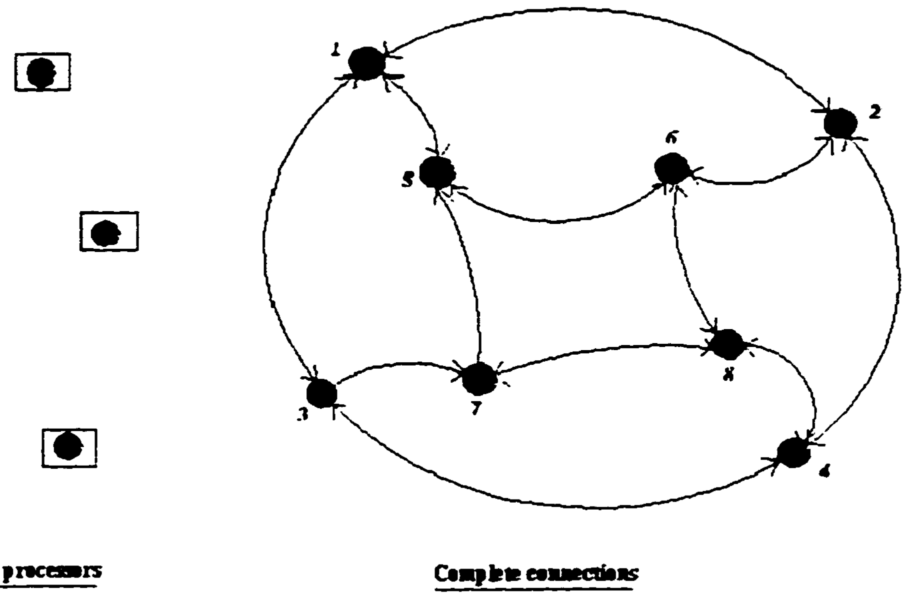
Figure 4-1-4 : Mapping Nodes with Extra Available Processors



Note that the geographic location of the nodes after id mapping was chosen in order to have a clean graph. The mapping is done at random from a set of available processors (see figure 4-1-4). (The efficiency of this approach is a very interesting problem in networking but not very critical in our model).

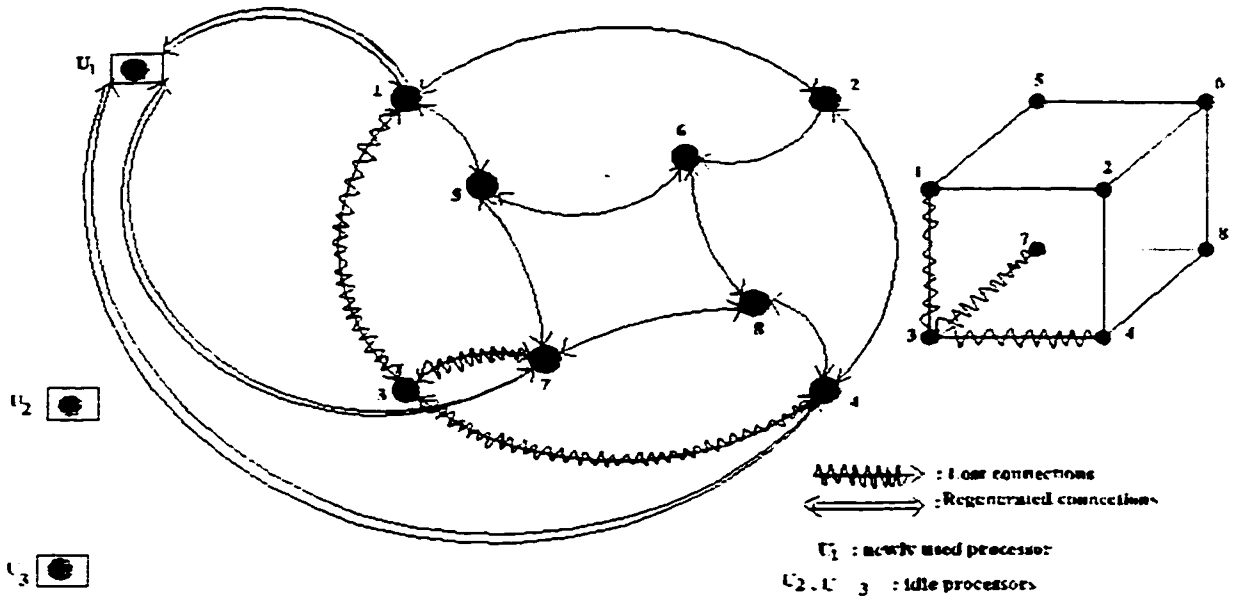
<u>Second compute the connection tuplet accordingly:</u>

Figure 4-1-5 : Complete Mapping Plus Extra Idle Processors



Anytime a connection between two processors is lost (connection between two nodes of the QCNN lost) a search for a new connection is done dynamically (illustrated by figure 4-1-6).

Figure 4-1-6 : Regenerating Lost Connections



U_1 : first unused processor

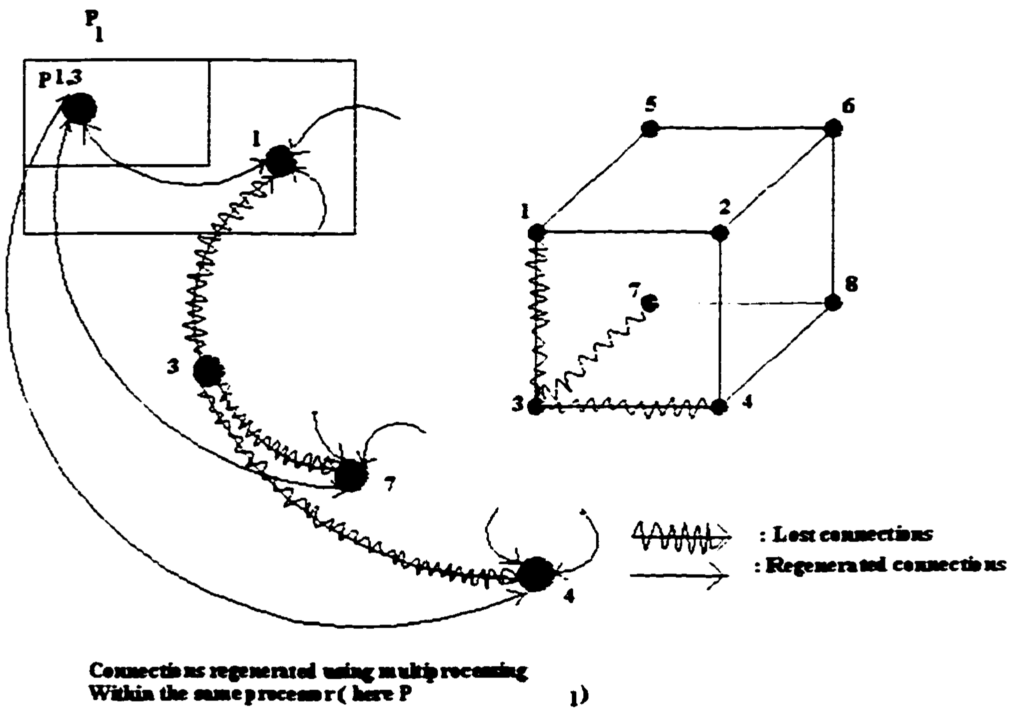
processor 3 ($p3$) becomes idle $\Rightarrow$ the following connection are lost (3,7), (3,4), (3,1) and the following connection tuplet need to be re-generated again using let's say U_1 the 1st unused processor. U_1 would behave as *p3*.

Generate $(U_1 = 3;1,2,4,0,7,0)$

U_1 is reassigned the id for the "lost" processor which is the corresponding processor for node 3 of the QCNN in this case node 3 have other available processors to use but in other cases those extra processors may not be available and the QCNN reliability and complete functioning need to be touched on. As far as implementation is concerned this model was first implemented within the same processor as a set of processes. Deadlock and memory starvation caused that implementation to be dropped. But now we would like to take advantage of that when one of the processor in our current implementation fails and no other processor is available.

Let's illustrate that (let's focus on $p3$ alone):

Figure 4-1-7 : A Node with a Built-in Controller



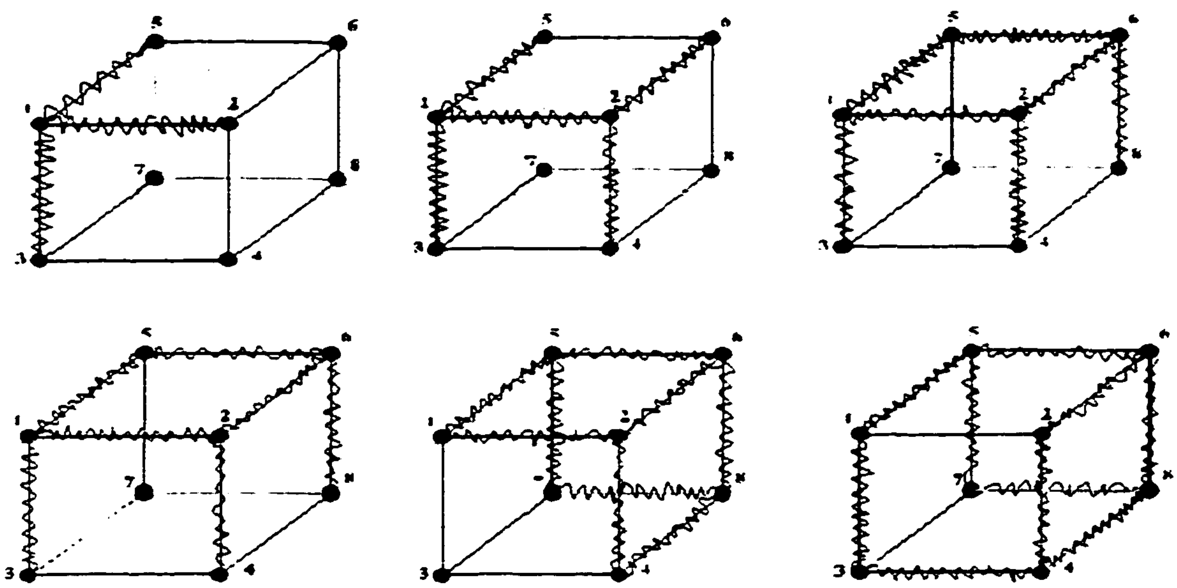
When $P3$ fails all connection in (3;1,0,4,0,7,0) would be regenerated within another existing process in this case $p1$ is the candidate. $P_{1,3}$ represents a process created in P_1.

This can be done for any failed processor. The whole multiprocessors implementation can collapse to a multiprocessing environment only or we can have a mix of both. (another very interesting situation).

When all the nodes of the QCNN are mapped to all processors the actual connections tuplet generation is done by a separate processor which we will call the Controller. The Controller will be in charge of nodes to processors mapping and also connection monitoring. It's the job of the controller to re-generate any lost connection by searching into a pool of available processors. The controller may fail and this case is also very interesting. A solution would be to duplicate the controller as a process within each existing processor or to duplicate it across the other unused processors when they are available. The reliability of the model as far as implementations will be affected but the functionality of the model is another question. We can advance already that the QCNN will be very reliable when one or more processors fail. The reliability issue must be evaluated. As of now no numbers will be considered for this purpose but we can illustrate that again. A question will be: how many idle or failed processors within the model would cause the model to 'die' (stop functioning).

Here is an example for the 8-QCNN:

Figure 4-1-8 : From a Partial to a Complete Failure of an 8-QCNN



Waving Lines are Lost Connections Between the Adjacent Nodes

So. again the problem will be to analyze the effectiveness of the QCNN when a set of processor fails.

Remember that we choose to regenerate those lost connections by creating corresponding processes within existing functioning processors. Until we implement the full features of our QCNN we should suppose that failed processors can be replaced with processes across existing processors and that the early problem we face is how efficient the remedy is. In the next session we will present a description of the software side of the implementation. The section will be mainly about the networking aspect of what we can call now a simulation of the mode. The next section will put us closer to describing in detail the simulation of quantum processes and their monitoring with a cellular automata.

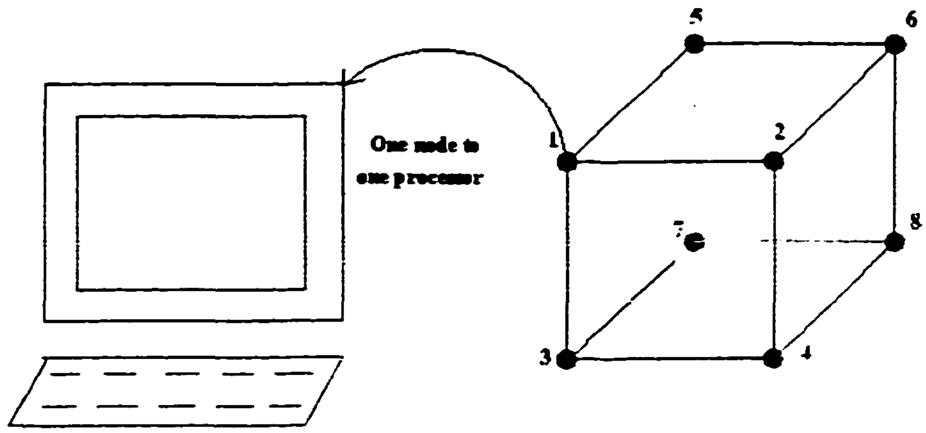
4-2 Software Considerations

In this section we will focus on a single processor as a mapping for a single node from the QCNN. We will also discuss the Controller implemented on a separate processor. Another role for the Controller will be specific to the neural network characteristic or the model and will be introduced later.

A node as a processor:

Every characteristic of a node described in early chapters has to be implemented. Figure 4-2-1 illustrates the mapping of one node to one processor.

Figure 4-2-1 : One Node to One Processor Mapping



As was mentioned in the previous section the communication between nodes is a single set of TCP/IP connections.

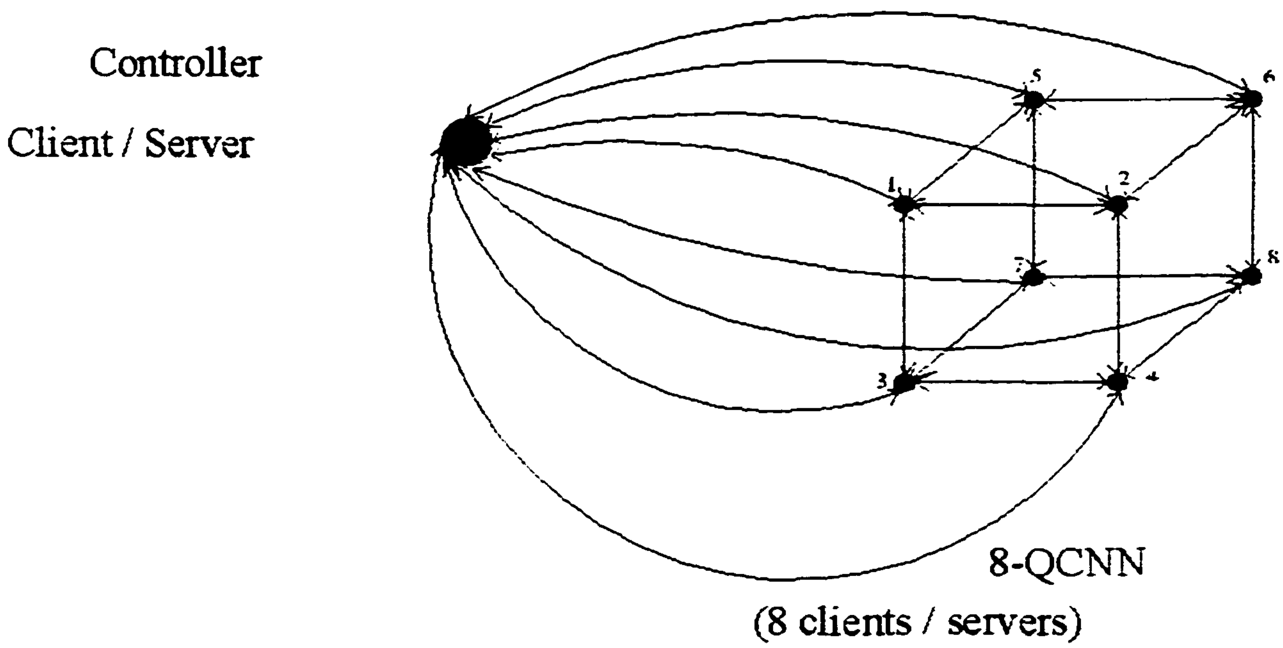
Each processor (node) behaves as a client and as a server.

As a client a node/processor would receive a message from its adjacent nodes (node belonging to the same connection tuplet) and also from the controller (which will be discussed later) and as a server the node/processor would send messages to its adjacent nodes/processor. The node communicates (receive/send data) with its neighbors according to a very specific set of rules (assumed to be the rules handled by the model which behaves as a CA). Those rules are based on 'real time' behavior of adjacent nodes. Those behavior are the actual simulation's values for scattering wave, propagating particle, stationary waves and oscillation. These events were discussed in detail in early chapters and will be revisited when we discuss the programming process.

The controller:

The controller is a separate processor whose function is to monitor the status of the whole model as to report connections failure, and regenerate lost connections. The controller itself behaves as a client/server and does communicate with all other nodes. Let's illustrate the controller relation with the model:

Figure 4-2-2 : Controller to all Nodes' Relation



Given a QCNN it's the controller job to

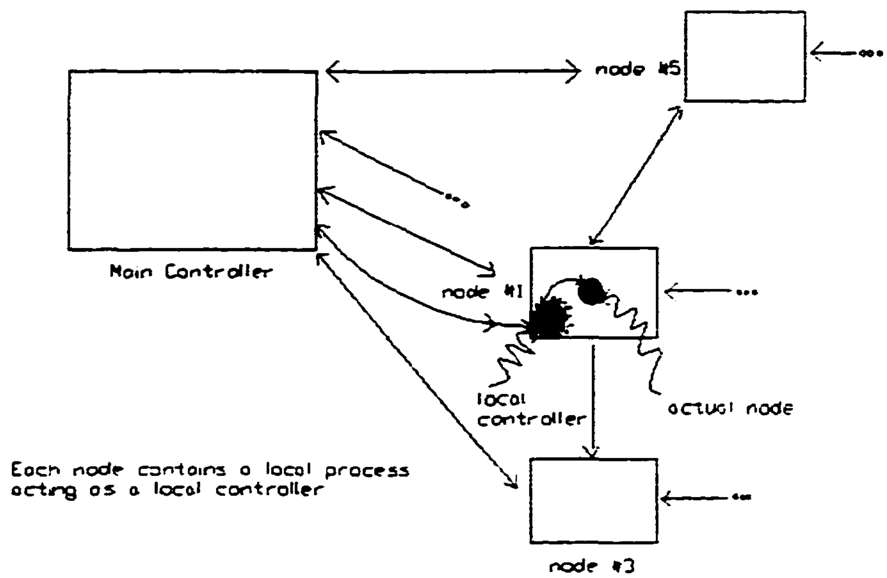
- generate all connection tuplets and store them in a database (connections database).
- Store all available IP addresses of all available processors. (IP database)
- Connect to all available processors (Awake all processors by establishing a two way communication channel)
- Assign to each node id a given IP address and establish the connections between the given node and its adjacent nodes based on the connection tuplet from the connection.

- Regenerate new connections when a processor 'fail' (connection lost) by searching for the next available node or nodes.
- Monitor the perseverance of the CA architecture through the life of the QCNN.

The controller does not interfere directly with any process personal to a specific node as each node interacts with its neighboring nodes independently from the controller. But each node must send to the Controller a signal acknowledging being 'alive'.

We can already feel the system deficiency when the model is a large one. The controller must monitor all nodes functioning and if it's done sequentially in real time that will definitely slow its performances. Again if we introduce the fact that there is a certain nodes' failure threshold under which the model will still perform. We can introduce at the level of the controller some type of randomness when it comes to how often the controller should check if a given node is functioning or not. Another idea will be to create a local controller to each node. The local controller will be a local process constantly requesting an acknowledgement from the local node. As a connection from that node to its adjacent node is lost, the local controller would send a signal to the main controller informing it of the event. It's true again that the local controller also behaves as a client/server. Let's summarize this by the following illustration.

Figure 4-2-3 : Controller Failure and Take-over by a Local Controller



The actual node and the corresponding local controller are within the same system (processor) we can have as many local controllers as number of nodes.

4-3 Programming the model

The model will be implemented within the same processor using multi-processes.

The system is a 300 Mhz Pentium II computer with 64 Mb of memory running windows NT 4.0 OS. The client/server applications for the network were written in C++.

The controller, a process in itself, causes a set of 8 processes to start locally. The controller as was maintained before computes all connection tuplets then assign to each Id a given port number.

The controller was named CANN (controller for the Artificial Neural Network). The application containing the CANN has two other port the ANNC (ANN client) and the ANNS (ANN server) all other nodes are set of ANNC/ANNS. The controller would compute all connection tuplets the following way for an 8 nodes model:

The function in the code is called GenerConnect ( )the function would receive a specific node id and would return all ids for all nodes adjacent to it. In this node n, s, e, w, i, o stands for ids in all directions. A value of zero means that there is no adjacent node on that direction.

If the node id we are trying to find the adjacent nodes for is x then (for an 8 nodes model):

The following is the C++ code used to compute the connections for a given node:

```
n = x - 2; // North
s = x + 2; // South
```

$e = x + 1;$ // East
$w = x - 1;$ // West
$i = x + 4;$ // In
$o = x - 4;$ // Out

$if\left((n < \ = 0) \parallel (n > 2 \&\& n < s)\right)$
$n = 0:$
$if\left((n\%2 == 0) \parallel (e > 8)\right)$
$e = 0;$

$if\left((s > 8) \parallel (s > 4 \ \ and \ \ s < 7)\right)$
$s = 0:$
$if(n\%2 == 1) \ \ w = 0:$
$if(i > 8):$
$i = 0$
$if(0 < 1)$
$o = 0:$

This is the kind of algorithm we would like to automate. Without manual update. and when the model shrinks to a lower CA or grows to higher CA. the connection tuplets should be computed dynamically.

The problem is interesting and can be stated this way : Can we create an algorithm which can be used to assign labels in a dynamic way to a growing/shrinking CA. which is a 3D-lattice result of tiling *nxnxn* units cubes of dimension $2x2x2$. In our case we will avoid that by manually changing the algorithm to suit each n-QCNN where n is the number of nodes in the model.

The multiprocessing environments was inadequate for memory starvation and deadlocks arise. The model will not be doing any complex computation. The exchange of messages across the processes is very slow. For this latter reason the implementation of each node on a single processor will be used. All nodes, including the controller are one separate processor and behave as a client/server.

First, the whole model was a mix of NT and Linux machines with different processors, brand and powers (AMDs and Pentium with speed ranging from 120 Mhz to 400 Mhz). While the configuration (hardware and software) is different, the communication protocol is the same. The communications use simple TCP/IP protocol. But the client/server features have to be rewritten and re-updated within each system. The controller will implemented in a Linux server. The communication between processors in time affected the behavior of the model.

Based on the CA rules, simulating a wave arrival at a given node before another one is difficult due to real problems caused by bad communication between two adjacent nodes. The CA uses an algorithm which introduces latency so to simulate the 'real time' it would take a scattering wave to move from a node to another node. All 8 nodes for this model were implemented on Linux machines. Writing client/server applications is easy. And the good thing about it is that the same code is duplicated across all nodes. Even the controller uses the same code. The main difference between the controller and the other nodes lays in the code used for the other functions. The controller monitors the communication and starts the model by assigning IP addresses to nodes based on their id. A node computes a set of values equivalent to the local stationary waves and the scattering waves. An ideal environment would be a system with hundreds of processors embedded in the same system. In order to mimic that kind of system, we should implement what is labeled as a metha-computation environment. In this case the processors would be available across the internet. Those resources can be used to implement a good distributed processing environment. For now we will discuss the networking in our set of Linux systems.

Our networking model is based on the Berkeley socket API. The model as was stated before. is a mix of inter-process communications and TCP/IP. Basic socket operation are handled a *socket ( )* system call would create a file through which packet of data are being written (received and other packet of data being read/sent). The socket accepts 3 integers parameters the domain. the type and the protocol. After the socket is created connections are established. In our case a node being a client/server means it needs to create a socket and make available to the system all IP addresses of the adjacent nodes. If a node has to connect to 3 other adjacent nodes then all 3 nodes must accept the connection before a two channels communication start.

A problem arises: The only way for a node to communicate with all 3 adjacent nodes at the same time is through the existence of 3 sockets. In our case we created only one and we binded the addresses for the 3 remote nodes in succession.

Let's explain things by using node 1 as an example.

- First the controller (CANN) is executed first on the main server.
- The controller will compute all connection tuplet for the given model (here 8 tuplet).

- The controller should remotely execute the client/server application on all other processors. (Note that all nodes are networked to the controller in advance. But no connection between nodes is established yet).
- The controller will bind through a system call (using *bind ( )*) an address to the local created socket.
- The controller as a server starts listening to all other nodes. Let's choose node #1 for this illustration.
- Node #1 as a client with an already created socket would send a request to the controller. (Note that id = 1 was randomly assigned to that node by the controller).
- The controller would acknowledge node #1 (and of course all other nodes).
- The controller as a server will accept the connection to node #1.
- The controller will then send to node #1 a packet of information containing the id and addresses of all adjacent nodes to node #1 (in this case node 2. 3 and 5).
- The controller would do the same for node 2. 3 and 5.
- Finally node #1 would connect to node #2. #3 and #5 randomly (in any sequence) and that will also depend on the local rule which depend on the local wave and the scattering wave being simulated at that given moment.
- Node #2. #3 and #5 would then proceed the same way.
- Finally when the whole network is fully connected (we mean here that each node is connected to its adjacent nodes) and that the controller enters its monitoring services. each node will be sending and receiving data from its neighbors. All nodes are in listening mode all the time.

We must remind ourselves that previously we discussed the problem of monitoring the connections and we delegate that to the controller. In this case the controller has to check each node and its corresponding connections for potential failure. This will put a lot of strains on the controller when the model is ver large. The solution we choose is to randomly check failed processors and connections. The options will be to implement a local controller on each node. Using Inter Process Communication (IPC) through Unix domain sockets. we can generate a local connection and monitor the node connections with its adjacent nodes. A question is : what if the local monitor/node connection fails?

A good idea will be a mix of both remote monitoring by the main controller (CANN) and the local controller. In this work only the CANN is implemented.

The following is an illustration of the node as a client/server:

Remember the problems associated with computing the connection by the controller. Based on the adjacency matrix for the model. some computed values of the 8-QCNN may be $<= 0$ or > 8 in which case we assign the value 0 . A value of zero means no connection on that direction or no adjacent node exists on that direction. Example: for node #3 the connection tuplet is $(3:1.5.4.2.7,-1)$ where 5. 2 and -1 would be associated the value 0 because for node #3 no connections *south. west or out*.

The new connection tuplet is then $(3;1.0,4,0,7,0)$. The connection tuplet using IP addresses can be written this way:

(128.80.70.3; 128.80.70.70.1, 127.0.0.1, 128.80.70.4, 127.0.0.1, 128.80.70.4, 127.0.0.1) where 127.0.0.1 can be the loop back address and can be used to monitor the node (local host).

In general the following will be considered :

- Values associated with parameters specific to wave scattering between two given nodes.
- Values associated with parameter specific to stationary waves within the node.
- Values resulting from the interactions of the local and the incoming particle.
- Values associated with the node being an oscillator.

The ideal system would include all computations across the whole set of nodes. By this we mean that a local computation would make available to the node all values for the above mentioned values. For our model this is very difficult especially when the processors are within a heterogeneous set of operating system and computing power. Also it's time consuming to write programs for every platform (node level) in order to calculate all values associated with the process of wave scattering and harmonic oscillations.

So, in our case we are going to use a set of tools written by others and which would facilitate the task of values computations so we can concentrate on analyzing the results and relate them to traditional results found by other researchers. Luckily for us, two software packages were chosen for the numeric computation level: *Mathematica* and *Interquanta* . We will use a *Mathematica* Package which is a set of prewritten functions which handle all needed quantum mechanics computation. The package is written by Jim Feagon Professor at the California State University. The package is a companion to his book titled *'Quantum Methods with Mathematica'* [28]. The package in our case will be mainly used for numerical solutions to Potential Scattering.

When it comes to the computation of values associated with the S-matrix and computations closely related to QED. Our choice shifted to *Interquanta (IQ)*. *Interquanta* is a software which accompany the textbook by S. Brandt and H. D. Dahmen titled *'Quantum Mechanics or the Personal Computer'* [8]. *IQ* simulates actual experiments which are done in the lab. It unables the user to change values of all parameters and study their corresponding plots. We used *IQ* to present already established results in topic such as scattering in 1-D and 3-D and 1-D and 3-D harmonic oscillators.

IQ was easily integrated in our model because of the following:

IQ presents a set of files called descriptor files. Each file contains a set of descriptors. The descriptors are sets of commands together with the values on which those commands act on. Some commands target the plot format (size. direction. title. orientations. ... etc.) and the most important commands are those acting on wave values and parameters such as the potential. the momentum ... etc.

In our work virtually all graphics associated with scattering process is done using *IQ*. When we describe an already established result by plotting the corresponding function we would retrieve the right descriptor make changes to the descriptor to get the right plot format. When we establish our own result we, again, use *IQ* to get a graphical representation of those results. We find it very critical to not only show in our algebraic and symbolic way our result but also to graph those result so we can easily compare them to each other and be able to analyze similarities and differences. In our case we will feed the software with different values through our own created descriptors. In this work we will focus only on those values directly tied to changing behavior of a wave packet and we will not present anything related to formatting.

What happened when we went to compute wave's values in time in our model? The problem is very difficult and involves writing the functions for each node and so for each platform the node is implemented on. In our case, instead of doing that we computed those values ahead and we duplicate them into a local database on each node.

The controller at start time would send to each node a set of initial values making up the local states of the local oscillators. Also when a scattering is involved, each node would receive new values from the controller (for nodes on the first layer) or values from an adjacent node (for other nodes). Those values are combined accordingly based on well defined Quantum Mechanics equation so we can trace numerically the time development of a wave packet. Also the behavior of stationary waves local to a given node with *Mathematica* Quantum Mechanics software package. We are better off studying the Scattering process numerically.

5- Analysis of the Model: A Theoretical Approach

5-1 Storage and Retrieval in the Model

As was shown in the previous sections a Quantum model of Neural Network presents more potential than any existing model. Unfortunately what we can simulate using 8 and 27 processors can be tackled by more powerful models with thousands of processors. Our resources were very limited but we did theoretically made our point. In all cases no Quantum Neural Network was built so far and even most traditional model based on techniques from statistical mechanics are only simulated on classical or Von Neuman machines. Our model with no doubt depends on the success of Quantum Computing. What we witnessed in our theoretical model is that the system can be capable of:

- Distributed storage with no need of addressing.
- A stored pattern can belong to more than one state. can be associated with itself or with other already stored pattern.
- The fact that a pattern can be retrieved and related to other stored pattern is one type of correlation process as the pattern can exist in more than one state associated with a collection of distributed states among a certain number of nodes.

In the following section we will take on each characteristics of our model and compare it to existing models.

* Distributed Storage:

We can label the storage process in neural network as the 'learning process'. For the comparison process we will use a book by Hemmen and Schulton titled "*Models of Neural Networks III, Association, Generalization and Representation*" it's the 3rd book in the serie titled 'Physics of Neural Networks' the article in the book is written by Palm and Semmen. They focus on two rules in this process, the Hebb rule and the Hopfield rule. A given pair of patterns (x^l, y^l) is presented to the NN at the input and output side which causes a given synapse (junction between neuron i and j) Mij to have a pre and postsynaptic values.

For each pair of value (x, y) there exist what is called the local synaptic rule $R(x, y)$. Each rule would determine the amount of synaptic connections.

When the stored patterns are binary there are 4 post-synaptic and pre-synaptic activities: pairs (a, a), $(1, a)$, $(a, 1)$ and $(1, 1)$. The synaptic rule is determined by 4 number: $R = (r, r_2, r_3, r_4)$.

For Hebb rule [see Hebb, D.O. (1949) '*The organization of Behavior*' (Wiley, New York)], a symmetrical coincidence rule $H := (0,0,0,1)$ for Hopfield rule [see Hopfield, J.J. (1982) '*Neural networks and physical systems with emergent collective computational abilities*' Proc. Natl. Sci. 79:2554-2558]:

The agreement rule or symmetrical coincidence rule $A = (1, -1, -1, 1)$

The Hebb rule increases the synaptic matrix element for coinciding pre and post-synaptic firings and Hopfield rule increases the synaptic matrix element for agreeing pre-synaptic and post-synaptic states and decreases the synaptic weight for disagreeing states.

It's given that both rules are product rules when $R(x, y) = xy$. Note again that with the 4 pairs (a, a), $(1, a)$, $(a, 1)$ and $(1, 1)$ when $a = 0$ we get the Hebb rule and for $a = -1$ we get the Hopfield rule.

The reason we introduced the above information is to be able to connect to other ideas of pattern storage and especially the distributed storage process.

In our case local rules depends on the physical state of the incoming wave and the local interaction. The node as a neuron has no defined post or pre-synaptic activities as in other models. Actually our neuron is not similar to a biological neuron.

Let's go back to the storage procedure using the Hebb or the Hopfield rule. Storage can be local or non-local and supervision is used. There are two procedures. The incremental storage procedure where the matrix for a synapse is

$$Mij = \sum_{k=1}^{M} R(x_i^K + y_j^K)$$

And the binary storage procedure denoted $Mij = Sgn(M.j)$ when Sgn is the *Sigmoid* function of the form $\frac{1}{1+\exp(x)}$

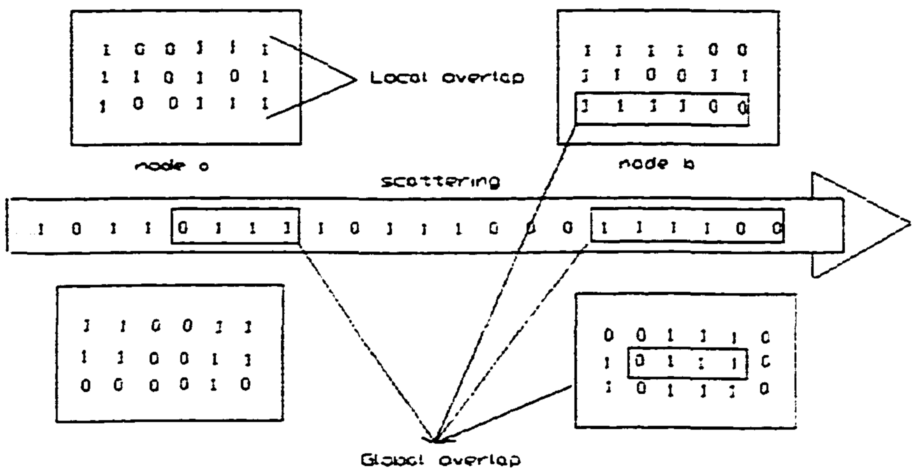
For distributed processing local rules exhibit a hetero association between an address and the content pattern. For the Hebb rule a pair $x_i^K . y_i^K$ of active neurons acts on one synapse Mij.

The storage is distributed if the pattern being represented by a pair of neuron affects other remote set of patterns. Obviously the connections in the network span more than one neuron and so the effect of a single neuron activity would span more than one neuron. A stored pattern is contributed to by more than one neuron and more than one stored pattern would cause an overlap between different set of neurons contributing to different stored patterns. The main aspect of this type of distributed processing is the requirement that the patterns must be non-singular and overlapping.

In our case the model exhibit an all-for-all type of storage process. All nodes contribute in a way or another to the pattern storage. The best characteristic in our model is the existence of overlap within a single neuron and across all other neurons.

The local overlap is caused by the effect of two interacting particles, the propagating one and the local one. Local superposed states can belong to different patterns and they do exist in time but their existence is time independent. Their retrieval, which we will discuss later, is time dependent of course. The local overlap, is due to local oscillations caused by different incoming waves and are maintained by absence of dumping which is due to a permanent local energy. Overlapping across the whole network (non-local) is represented by the scattering process (non-stationary scattering) and does depend on time. In our case the pattern is not related to specific values corresponding to the position of the two interacting particles within the box but by the resulting waves representing uncertain precise positions of the particle (estimated within a given interval using probability density). The classical position of the two particles within the node at a given time can be mapped to the phenomenon exhibited by traditional Hebb and Hopfield nodes. In our model the main advantage is that a given pattern 'piece' can exist in more than one state within the same node and multiple states across the whole network. We can illustrate this by figure 5-1-1 showing 4 nodes using binary pattern.

Figure 5-1-1 : Local and Global Overlap of Patterns



You may have noticed already that we are comparing our model to models using associative memory. In the retrieval phase, an input pattern will behave as the address of the content pattern or stored pattern. In this section we will focus on the retrieval of a stored pattern.

S^C is the set of stored patterns or content patterns.

S^A is the set of input patterns or address patterns.

In the retrieval process a mapping $(S^A \rightarrow S^c)$ would map an input pattern to its corresponding stored pattern.

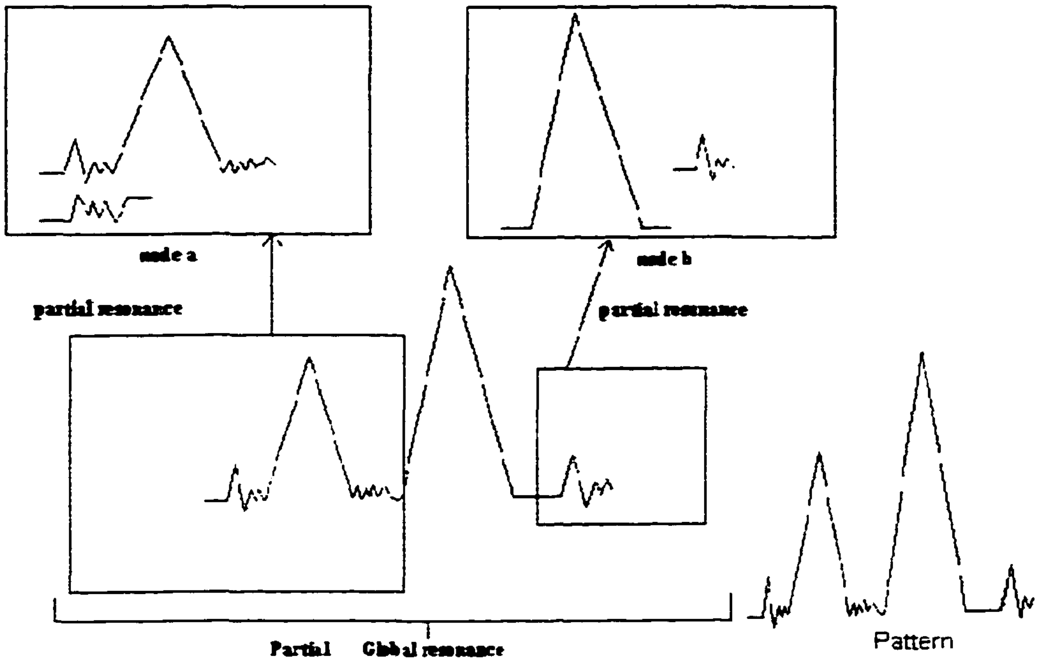
$$\{(x^1, y^1), \ldots, (x^M, y^M) / x^K E S^A, y^K E S^c\}$$

In this type of addressing a fault tolerance process exist by which the Hamming distance is used to map a set of address patterns to one content pattern. A given interval for the Hamming distance would represent a closeness or similarity between patterns. So, if x and $\hat{x}$ are two patterns and $h(x,\hat{x})$ is the Hamming distance, and if x^k, x are two address pattern $(x^h \neq x)$ then if $h(\dot{x}, n) < (\hat{x}, n^k)$ $\dot{x}$ is close to x than to any other x^k.

In our case addressing depend on a global collective resonance and on local collective resonance. A pattern may be completely retrieved when it's completely 'recognized' through retrieving pieces of it (local node resonance) or can be partially matched to a closest one through a summation of partial waves scattered through the network.

Let's illustrate that in figure 5-1-2:

Figure 5-1-2 : Local and Global Resonance During Retrieval of Patterns



Fault tolerance in traditional Associative memory models must by well defined so to limit retrieval errors where an address pattern is mapped to the wrong content pattern. Fault tolerance is handled by a probabilistic process which is not always reliable.

In our case the existence of a pattern in more than one state (across more than one neuron) help minimize retrieval errors. We may talk of highly redundant storage process.

For traditional models the repeated process of back propagation of errors would cause the errors to be minimized but do not guarantee a retrieval.

From the same text referenced at the beginning of the section using the summary from the article by Palm and Sommer we can also summarize the features of our own model within the same context.

A one-step retrieval process in a feed-forward model is also an example of distributed and associative storage. Obviously this type is not powerful or reliable and is prone to errors. In our model, a feed-forward process will cause global resonance result of retrieval and the difference, as far as distributed storage is concerned, is handled in our model by a multiplicity of local and global states (states which will contribute to the retrieval of a pattern).

Our model does not use the fixed-point paradigm used for auto association. We can assume that our model presents a multiplicity of fixed points all or some of which can contribute to the retrieval process of a stored pattern.

As far as the capacity of the model is concerned, the contents in our model can be mapped to a lower size network and we can state that it's not proportional to the number of neurons. But it's true that there is a lower bound at the level of which, even with the principle of super-positions of states, storage would be limited. A more interesting suggestion would be the case where the model shrink (number of neurons decreases). In that case we can add up more particles within a given nodes. This can be equivalent to creating sub-lattices within a given node. Only the Hebb model presents a capacity larger and non-proportional to the number of neurons in the model but not close to our model.

In conclusion, and in our theoretical quantum model, the distributed storage can occur within a single neuron and can be increased by adding more particles to the neurons.

Our model does not mimic exactly a biological neuron, as far as the activities of a synapse is concerned, but the model is more powerful given the fact that a pattern can exist among multiple states locally within a neuron and globally within multiple set of neurons. All neurons contribute to the storage and retrieval of a stored pattern. It's true that the contribution of a given neuron can be weak but can be compensated by highly exited and correlated states within another neuron.

5-2 Model Analysis: Relation to Other Models

Now that we have established -using Quantum Mechanics techniques- that the model proposed can be very powerful in theory when dealing with problems of pattern storage and pattern correlation, we need to show if the model can be implemented as a real physical system. By the latter we mean if we have the technology to turn our simulation into a feasible system. It must be noted that from QED to ANN with the help of CA we managed to create a system where all the parameters are chosen and controlled to achieve what we were after. In Quantum Mechanics. we can witness many models created for the purpose of simulating and analyzing processes of quantum nature of very difficult situation(i.e. Fluid Dynamics. Many-body systems ...etc). In our cases the model has to be simplified and also controllable.

Let's remind ourselves of the architecture and the functioning of a unit (or a node or a cell) in our QCNN. A node behaves as a particle/wave detector. The existence of a potential at the node level and the way the potential value is varied (node being an off switch and also a dimmer) turn the node into a wave scatterer. The scattering process was described in detail before. The other most important characteristic of a node is being a harmonic oscillator. In this present work we limited ourselves to a single particle oscillator. Choosing to put all those nodes on a 3-D lattice (a cubic lattice) and the use of CA was very helpful in controlling the whole network by setting specifics CA rules. A given node has a predetermined number of neighboring cells at a fixed distance. The way the model is simulated does not mean the architecture should be mapped exactly when constructing a real physical

QCNN. The main purpose of using CA Techniques and a 3-D lattice is to be able to simulate the system on a set of classical computers. With an *nxnxn* lattice we used n^3 microprocessors connected together using a simple TCP/IP protocol. A simple set of messages are distributed across the systems. We showed what happens at a microprocessor level which, acts as a given node. A quantum process at the node level is still a complex process. But we did limit ourselves to a single particle at the node level. Scattering is carried by the process of sending set of values corresponding to the scattered waves packet from a node to another node. But the creation of the scattered wave is the result of local computation which take into consideration the local value of the potential and the value of the energy which maintain the local oscillations constant and so the local information memorized. The other aspect is the direction of scattering: Single to one neighboring node or multiple to a sequence of adjacent nodes. In real physical system, most scattering processes depend on real time that is why in our case we introduced latency in a controllable way to simulate 'real time' process. The latency was introduced to delay and to advance the value of time taken by propagating wave between two nodes.

Now that we have briefly reviewed some aspects of the model we must discuss the feasibility of implementing the model using real quantum devices and explore their suitability in our model.

We are going to first survey a list of newly developed devices and analyze those which present any hope for direct implementation:

A paper by Singh and Hong titled *'Quantum Well Based Excitonic Devices'* presents a connection to quantum effects and an implementation in neural networks. In our discussion here we need to see if their device is of any use to our QCNN. Their focus is mainly in implementing traditional ANN such as the Hopfield model and Associative Memory. They are also trying to mimic the functioning of biological neurons. Threshold and integration properties are considered the first using a device they call controller-modulator and the second using a device they call spatial light modulator which work as a synaptic mask. In this case we will focus mainly on the characteristics of their devices. Both of them are based on excitionic transition in III-V quantum well structures. These structures are suitable for the implementation of optical Hopfield networks. These devices can be used to construct optical switching devices. Note that their device final state is to get a photonic switches and that the real quantum aspect is not taking advantage of in the ANN implementation. The photo-current produced by these devices can be helpful in creating thresholding processes.

The characteristics of this device make it possible for the implementation of a stable neural network. Their paper concludes by claiming that the photonic characteristics of those devices support the idea of using an optical architecture to implement neural networks. This device may be helpful when we consider designing a circuit which would implement the on/off switching present in at the node level of our mode.

In Eindhoven Netherlands E.C. Mos of Philips Research Lab and H. de Waardt of Eindhoven University of Technology in a work published in the journal Applied Optics constructed what they called Laser Neural Network. Their main aim is to create a computer where the basic processing units is an actual event equivalent to splitting the beam of a diode laser into its longitudinal nodes. Each node represents one neuron. But the rules are very similar to traditional neural networks once again. The work in this paper is not of very important use in our model but their exploration of optoelectronics devices is a good sign that there is progress in taking advantage of quantum processes: Their device generates multiple quantum well Laser. Once again their work can be considered as an encouragement but not of a direct use in our model.

5-3 Advances in Quantum Computing: An Overview

The following work presents some aspect of our model especially the CA characteristics. Actually the name the author gave to their model is *'Quantum Dot Cellular Automata (ACA)'* also abbreviated QCA. Their work is not on neural network but it's mainly on creating new computing devices based on quantum techniques. They claim the possibility of constructing a new type of transistor-less computing. Their work was published on The American Institute of Physics Bulletin of Physics News. Their quantum dot is what they claim to be a zero dimensional artificial atom. The atom can free a pair of electrons within a set of four adjacent and closely spaced dots. The electrons are controlled so to move from a dot to another dot. A binary bit is being created by a specific electron's motion representing the state of a given cell. A cell here is the set of the 4 dots representing the set. They show that the electrons different configurations can represent the binary 1 and 0 and that the whole model of those dots behaves as an artificial neural network. More than that. they claim that the architecture and the rules make up a programmable cellular automata network. The computing is done here by quantum interactions among all dots in the QCA. The good news for us is that at the atomic/molecular/optical physics meeting in SantaFe it was reported the success of manipulating a single electron by another closed by single electron.

Wolfgang Porod of the University of Notre Dame (Applied physics letter) together with other researcher constructed quantum dot cellular automata where a dot-dot coupling was achieved. Their cellular arrays are set of coupled quantum dot molecules. Each molecule (cell in this QCA) is a set of four or five quantum dots close enough to unable electron tunneling between two given dots. Again, their main application is the creation of some type of logical and computing gates and so they target quantum computing. Their work was supported by those of Orlor, Amlani, Bernstein, Lent and Snider [2]. These recent advances are again encouraging given the need in our model to entangle a single particle within a cubic volume which is represented by a 3-D cubic potential of varying size. The tunneling effect demonstrated by these researchers is a proof that in our case we can rely on those results to construct some characteristic of our cellular node. And here again, it's shown that the need for a CA for the purpose of control is critical. The coupling in their work is considered when constructing Boolean logical gates and other gates used in computations schemes. In our case we deal with more than just the coupling between two nodes, but with multiple coupling in parallel or in other word a synchrony among a set of a certain number of nodes representing a specific stored pattern or patterns.

The logic devices implemented using QCA are claimed to perform very complex computational processes. Logical gates, which can be programmed, can be built using these devices. The success is mainly due to the feasibility of coupling two electrons entangled at two closely positions. Their work faces some practical challenges the same we face theoretically (given that our work was a simulation). The process of input is carried by changing a conductance and by that process repelling electrons from a dot to another. A given state is 'read' by a process of measurement. That process, as we stated before, would cause a perturbation of a state and by the same way causes a given information to be lost. Those researchers constructed what they call an electrometer made up of other quantum dot. It seems this is similar to our worry when we choose to act on a node with a weak perturbation strong enough to generate a feedback from a given node (local feedback) without destruction of that local information.

The work of Platzman and Dkyman[27] presents more encouraging results when it comes to taking advantage of quantum technique in designing new quantum computing devices. Their paper titled 'Quantum Computing with Electrons Floating on Liquid Helium' presents a quasi-two-dimensional set of electrons in vacuum trapped in a hydrogenic levels on top of a film of liquid helium. The motion of these electrons is controlled in a way to construct the process by which a quantum bit is created . Here, individual electrons are trapped within metal pads under the helium. A wave function within the system is manipulated so to correspond to reads of excited electrons from the surface. The author claims that their devices can be used to implement analog computing. They call their system AQC or Analog Quantum

Computers. Their AQC is a set of N interacting quantum bits actually called qubits by Dr. Bennet from the Watson Research Center of IBM. A qubit as we know has 3 states instead of 2. The authors are taking advantage of the superposition states of electrons. Their approach is similar in a way to ours because of the use of wave functions. In their case, the initial input wave function Ψ_0 representing the qubits is a set of states. After different interactions between those states, interactions depending on time and under a given Hamiltonian, a final wave Ψ_f would represent the answer to the computation input via the initial wave. Their challenging problem is finding a suitable algorithms based on which those qubits interactions would lead to a final sought result. Another problem, and for a given state, the interactions among those qubits must be isolated so to avoid interaction with the outside environment. The authors are facing other challenges similar to those we face with our model. Time dependent processes should be precisely controlled. The authors fortunately, and good news for us, proposed among other solution such as those imported from quantum dots devices and nuclear spin of atoms, they proposed techniques from QED especially cavity quantum electrodynamic.

Like in our theoretical model the negative effect of de-coherence effect was considered and they claim that in their device de-coherence is small or by quoting directly 'acceptability small'. Let's present some of the number and equations they explored.

They suggest using a set of N electrons such that $1 < N < 10^9$ trapped within a vacuum at a low-temperature liquid-helium. A single electron in a film of thickness $d \geq 0.5\mu m$ will be weakly attracted by a potential $V = \wedge e^2 / z$

Here: e : charge of the electron
Here: z : coordinate of the electron

$$\wedge \equiv \frac{(E-1)}{(E+1)} \cong 0.01$$

and $E \cong 1.057$ (is the dielectric constant of liquid helium)

The helium barrier requires $1eV$ for an electron to penetrate it. The electron motion along it is described by 1-D hydrogenic spectrum and the m state has an energy

$E_m = -R/m^2$ R is the Rydberg energy. $R = \wedge^2 e^4 m_e / 2\hbar^2 \cong 8k$

and a Bohr radius $r_B = \hbar^2 /(m_e e^2 \wedge) \cong 76 \mathring{A}$

The only existing coupling to the outside environment is what they label as the thermally excited height variations $\partial(r.t)$. r is the vector representing the electron at time t. Also the following equation was established

$$\frac{\hbar}{T_1} \cong R\left(\frac{\partial_I}{r_B}\right)^2$$

T : Time

r_B : Bohr radius

R : Rydberg energy

$\partial_i / r_B \cong lo^{-3}$ $\frac{1}{T_1} = lo^{-6}$.

The authors claims suggest that one can use the lowest two Hydrogenic levels of a single electron as a qubit. This research concentrates mainly on trapping one single electron within a space subject to small external perturbation. The main challenge again is to construct algorithms efficient enough to take advantage of those quantum states.

Like in our case, the effect of resonance is also explored. We should conclude that here again we are encountering very encouraging ideas which target especially quantum devices that can be used to construct potential quantum computers. What we can notice here is that in our case we claim we have the algorithm and a model which is proven to be theoretically sound. And now we are looking to physically implement the model using feasible quantum devices. So far the only result we get is the feeling that these devices are very realistic. But we should not forget that what we are after is not a computing machinery similar to a classical computer but just more powerful one which can enable us to implement more 'concrete' artificial neural networks. The applications we are targeting are assumed to be un-tractable with classical computer and potentially solvable by quantum computers.

So far the works presented are not directly related to our work but we can assume they are hits for progress in the design of quantum devices. In the following work we will be revisiting QCA or Quantum Cellular Automata. These results are very recent and show a lot of promises. The goal in this work shows the necessity to find new ways of computing which would replace the Field-effect transistors or FET. The problem of circuit integration or the process of constructing smaller and smaller circuit while increasing computing power is now facing an insurmountable limit. These classical devices are becoming closer and closer to quantum limit. Switching will not be carried anymore by FET. In this case, and at the quantum level, other type of switching paradigm should be found. At quantum level we will be dealing with electrons and so with their different states. In order to take advantage of the state of an electron we should be able first to isolate the electron and second to monitor its location within a specified space. In their paper titled 'Digital Logic Gate Using Quantum-Dot Cellular Automata' Amlani and company[2] presented a new and functioning logic gate using QCA. Again QCA is a nano-structure made up of quantum-dot cells with 4 dots each. Each cell has actually two electrons which can be found in two state each diagonally opposed. Their structure of 5 cells was shown to demonstrate the functioning of an OR and an AND gate. Cell polarization which unable positioning of the electron in one or the other diagonal position is done via capacitance which causes the quantization. The fabrication process is partially shown and the authors describe the material and the technology used to construct those quantum gates. They demonstrated that application of potentials to those dots in a very defined manner, (set of rules in the Automata) shows the process of electron switching. The challenges facing these implementations are mainly the critical temperature under which the material used to construct the devices would react. Remember that we are speaking of nano-structure at the quantum level. To conclude the work done by these authors we can state that in classical computer the information is being manipulated by switching voltage while in the QCA information is a set of cell states where each cell is a set of 4 dots representing coupled diagonally electron states. Information in a QCA is transferred by successive polarization of adjacent cells. The main thing here is that the polarization is manipulated by external voltage. The output on one side of the cell would depend on the polarization states presented as inputs in the other adjacent cell. Again the temperature under which these devices would work is - $272.9^{o}C$ or $0.1K$. Those numbers mean a feasible complete quantum computer is still a theoretical model. But again those ideas show that basic computing quantum device, can be constructed.

In a paper published in Science titled 'Bunches of Photons-Antibunches of electrons' Markus Buttiker [12] at the department of physics university of Geneva. shows that the intensity correlation of the light measured with two detectors depend on what they call bunching of photons. The particle in these experiments are actually fermions which can occupy the same energy state (unlike electrons). In the case of electrons or fermions he talk about anti-bunching. For the case of electrons mesoscopic conductors can be fabricated so when we are at low temperature the wave function is apparent and the transport is phase coherent. Bunching in the case of photons and anti-bunching in the case of electrons would be useful in scattering where both situations arises. A connection can be between existing quantum devices and the implementation of time-dependent scattering processes so critical in our theoretical model. In this latter paper correlation phenomena will be very important in our understanding of states occupation with photons versus electron.

Again there is a sense of completeness when we think about the feasibility of constructing a physical model. A link should be established between the architecture and functioning of a node in our model and those of a quantum dot. The goal in our case is not the building of quantum gate capable of some logical and numeric computation but just to prove that when at the quantum level the construction of a device which can exhibit quantum states helpful in designing logical gates can also support the idea of building basic 'quantum nodes' in our model. Again, let's remind ourselves that we are after the realization of a Quantum Cellular Neural Network where the main process is pattern storage, manipulation, correlation and retrieval. In the next section we will survey work related directly to the main characteristic of our model. Our model being first a Neural Network. The next paper by Ashoori and colleagues [4] presents very important results using QCA. This time the purpose is to add more electrons to a quantum dot and explore what the author labeled as 'localization-delocalization transition in Quantum dots'. They found out that single-electron capacitance spectroscopy can measure the energy or energies required to add individual electrons to a quantum dot. The dot being within a given potential and the wave functions corresponding to the states of these electrons can help analyze the energies relation to the electrons states and potentials. For a low density of electrons these latter occupy spread out local states within the dot. In this case they speak of localization and when the density is high the electrons become de-localized. In our case this study is very important as we suggested before a node in our case can contain more than one electron. And these electrons do not exist at the initial states. So adding and subtracting electrons is very important and this paper describe physical feasibility. The problem or the challenge is limit to how many electrons can exist within the quantum dot and what is the effect of local potentials and their effect on the behavior of the electrons. An exiting and not explained situation is the pairing phenomenon when correlated repulsive process occurs in two adjacent regions. So, given a quantum dot, adding or subtracting an electron is very critical. The addition is dependent on the voltage at the gate. Let's explore this result for our model. In our case if a node contains more than one electron, pairing of two electrons is very important for it will unable us to address their states as belonging to the same incoming waves (input) and also unable us to know more about scattering corresponding waves. But local pairing is not as important as remote correlation between electrons of two adjacent nodes. We should explore this phenomenon for a node belonging to the same scattering path a one to one pairing will, without a doubt. solve the synchrony problem when trying to read out or retrieve states corresponding to the same pattern. We already advanced the idea of collective coupled oscillations as one aspect of our model by which a set of correlated pattern can exist and be related to each other. A node in our model would be very similar to a cell in QCA. In our case the node is not a set of 4 electrons but can contain one or more depending on the complexity of the neural network. In our model we don't have any unit (gate) capable of computing. The main aspect of the model is storing pattern and retrieving them according to processes of time-dependent scattering and local oscillations. In our model we would like to take advantage of the phenomenon of super-positional state of an electron and not just a precise known localization. Once again we are not after constructing any logical or computing gates. We can state know that we are closer and closer to review work which will exhibit important results which touch on the characteristics of our model.

In the journal Science volume 285 of 1999 a surprising results in the domain of photonic transportation, especially when one characteristic of the model is to handle photons,was achieved through a very recent work done by French researchers at the Ecole Normale Superieure in Paris. The group of researchers led by Serge Haroche were capable of designing an experiment where a single photon was manipulated. Those researchers were able to make repeated measurement of the photon without destroying it. The case is similar to trapping a particle in a box and making measurement on it. The phenomenon is called quantum non-demolition. The measurement of quantum state was done before but this time it was unique as it was done for a single photon. It is known that trying to perform a measurement on a quantum state would disturb it and the measuring process would lead to wrong results. Researchers use a technique called interferometry. Interferometry is a technique by which two beams of light are mixed together so that a change in one beam or the other would affect the beam resulting from the mix. A signal passing through one of the beam would leave a recording or imprint in the altered interference pattern. Those researchers used the swollen Rydberg atom as the detector to check the existence of a photon within) the box and to show that the photon is still in its original state (Rydberg Technique. Science of July 19th 1996, p.307. So within that box (cavity) the atom would be absorbed and freed before the atom exit the box. In conclusion the main result is to act on a single photon. Now going back to our model these results open new doors to us. Making measurement on a single photon is very important. Think of the scattering process again. If you have one particle within each node. an incoming scattered wave when it hit the particle would cause it (assuming it's an atom) to free on photon. The process would be correlated with other in other remote nodes and in this case measuring the effect across the whole lattice can be thought of as a collective measurement needed to retrieve a stored pattern or superposed patterns.

6- Summary and Conclusion

In this final chapter, we would like to summarize what inspire us to explore new ideas in the field of Neural Networks and conclude with a list of future investigations which we think would lead to better models of Neurocomputing.

6-1 Limits of 'Classical Computation'

We know that with classical computer or the Von Neuman Machines we are faced with the following constraints:

- limited speed of computation:

  *Information cannot be transmitted faster than light (a given computation may involve spatially distant elements within a processor). Maybe we can make the processor smaller and smaller in size(increase circuit integration)?

- Limit of circuits' integration:

  * As the components of a processor become smaller and smaller they approach the atomic size which has the following consequences:

  ⇒ Disturbed switching activity due to *Heisenberg*'s uncertainty principle.

  ⇒ Unreliable computation.

Those limitations would have an impact on the implementation of ANN.

- Because most ANN are being simulated using many processors, we will face the same constraint at the processor level.
- Also, as the number of processors used in an ANN grows to thousands, other problems such as communication speed and network reliability arise.

All those limits have a negative effect on the implementation of true parallel and distributed environment which is a critical characteristic of ANN.

6-2 Quantum Computing

In this work we surveyed new advances in computing which offer new hopes in dealing with the limits of classical computation in general and ANN in particular. Those advances are in the domain of Quantum Computing or what is labeled as 'transistor-less computing' [56,60,79].

One of the ideas who encouraged researchers to explore those new trends originated from the work of the Physicist Richard Feynman on what he called the 'entangled state'.

The following is a summary of those ideas:

- Distant photons originating from the same particle carry imprint of each other. It's a sort of Information transmission and information processing at a distance.
- A classical state in a classical machine can be in an On or Off state only while a quantum state can be a superposition of many on/off states.
- There is a potential if we can carry basic computation steps at the subatomic level and be able to implement such computation in a real physical system.

Those ideas face problems of a different kind such as the fragility of Entangled States.

External perturbation (noise) or external perturbation (attempt to 'read' or measure the state) can cause 'De-coherence'. In this latter case it means that whatever information is coded into those states that information will be lost.

- What about the potential of implementation of ANN?

So far we can see two main advantages:

→ Quantitative: If those quantum computing ideas are to succeed , an increase of those computing states and number of corresponding particles at the atomic level can help build ANN with a huge number of units.

→ Qualitative: distances between computing units is subatomic and so will increase the transmission speed of Information and by the same process the speed of computation.

Advances in building Quantum Devices was touched on previously. The following is a summary of the foundations those advances are based on:

- A Quantum coherent state is a superposition of *n* distinct states.
- Under a given condition (a computation process) a Quantum State would de-coherence an collapse to a sought state (result of the computation) the state is not *1* or *0* but a superposition of both. The state is called *Qbit* (term first used by Dr. Charles Bennett from IBM).
- One can perform computation on each state which implies some kind of parallel computation.
- The necessity to construct Quantum Gates leaded to different ideas. The most promising one is Quantum-Dot Cellular Automata (QCA). Those techniques were discussed in detail before.

6-3 Quantum Aspects and Features of our QCNN

We achieved two main goals in general. First. we showed that it is possible to manipulate particles and act on them in a very controlled manner. Second. that constructing Quantum computing devices is feasible.

Finally, let's summarize ideas specific to our model or our Quantum Cellular Neural Network (QCNN):

= Quantum Aspects: (description of a single node)

- A node represent a virtual 'particle in a box' structure
- A wave packet associated with a propagating particle would scatter at a node level. (depending on a local potential)
- Presence of N regions of potentials within each node.
- The interaction between the incoming particle and the local particle would cause the propagating associated wave to scatter to other adjacent nodes.
- The scattering process is time dependent

In the whole system of neurons (our ANN) wave scattering will be well defined because we have a complete knowledge of the "Read" waves.

Preliminaries conditions:

-potential V at neuronal level are weak (small perturbation, a boundary condition exist)

-multiple waves, multiple scattering processes series of scattering that converge

-possibility to construct any wave at any level (neuron) in the network

-possibility of local construction of scattering (at a subNN level)

-be able to control the dimming capability of any neuron in accordance with different incoming waves (process of learning).

-We described before a linear position of neuron $x_1, x_2, \ldots, x_n$ and that will not be the case for our network. The position of the neuron will be well defined.

-The scattering of waves or scattering waves process has to come to a halt at a specific subNN level that should constitute the process of memory for we know we can trace back the scattering process to construct back the original wave.

All computations at a node level are developed using the Scattering Matrix(*S-Matrix*)

- The conservation of the particle Quantum number and the angular momentum contribute to the reconstruction of the scattering wave in time.
- The construction of the Scattering Matrix (*S-Matrix* is critical).
- Characteristics of the *S-Matrix* considered in the Scattering process.
    * Time reversal invariance
    * Spatial reflection invariance
    * Unitary of the matrix
    * Transmission resonance for the element of the Matrix
- The node being a quantum harmonic oscillator.

  The oscillations are associated with the local particle.
- Oscillations are generated by the effect of the propagating particle. (indirect manipulation of the local particle by the propagating particle)

== Cellular Aspect.

- Nodes are at equal distance from each other(characteristic which is transparent in the simulation as networked processor can be at any distance from each other).
- Well defined adjacency 3-D matrix (nodes located within a 3-D lattice and each node is connected to specific adjacent nodes).
- All nodes follow the same rules as far as scattering and oscillations are concerned.
- Each node in our case contains one particle only.
- The rules for all nodes are based on physical law of quantum Mechanics. An event on a given node depend on:
  - local step potential(a potential varying between 0 and a max value in a discrete manner).
  - local stationary waves (result of local oscillations resulting from the interaction of the incoming particle with the local one). This event is time independent.
  - incoming scattering waves originate as inputs and propagate through the network. The event in this case is time dependent.
- Communication between nodes is indirect: in the case where two nodes are not adjacent but in the path of a given scattering.
- Collective oscillations across the whole CA : All nodes are affected by a given scattering directly or indirectly and the interactions at the node level of a scattering wave and the local particle would cause the creation of local oscillations.

The following is a set of questions and answers related to the main characteristics of our model:

Why wave scattering?

→ given that we can reconstruct a scattering wave in time and so reconstruct the encoded pattern, we assume it's a sort of 'short term memory'.

→ the scattering will create a set of oscillations at each node level and we assume it's the learning process.

Why oscillations?:

→ 'long term memory' consists of those collective oscillations originated from the same scattering event.

Why a CA architecture?

→ all computations related to the scattering process will be possible with the chosen architecture. The node are at defined locations within the 3-D CA

→ This architecture makes it possible to control the model and easy to implement(especially when only a simulation is sought).

In the following section we will list ideas related to the storage and retrieval process in the model:

- distributed storage

because patterns are associated with oscillations across the whole set of nodes, we can state that the model presents a true distributed storage. All nodes contribute to all stored patterns. The model is highly redundant.

- Retrieval

  * in the case of 'short memory' retrieval is done in time using back propagator or reverse scattering.

  * in the case of 'long term memory' retrieval of a given pattern. encoded as a set of propagating waves. is done via local and collective resonance. At each node a local set of oscillations may resonate with those created by incoming waves.

6-4 QCNN Relation to Other ANN

The following section will cover a set of features related to other model which can be similar or totally different from other traditional and non-traditional models.

- Comparison to traditional ANN.

  → true distributed storage: All-For-All. A pattern is associated with and stored across all nodes.

  → no thres-holding: A node is always involved (weakly or strongly involved) in the storage or retrieval process. A node is more than just an On/Off switch.

  → no summation function: the input to a node is not the sum of input from all adjacent nodes. Only through scattering of waves we can talk of an input to a given node and only through collective resonance only we can speak of retrieval.

  → no energy functions to minimize: In this model. there is no local minima or fixed point to compute.

  → existence of a pattern in more that one state across a set of nodes (true redundancy).

- Comparison to other ANN with Quantum features.

  We will do the comparison through mentioning some already published work. Those works are mentioned mainly to show that there is a potential in approaching

  ANN with other methods than the traditional ones.

  * *'Optical Neural Network Using Quantum Well Devices' Jennings. Andrew 1994 (Applied Optics)*

  * *'Quantum-Effect Device: Tomorrow's Transistor?' (Scientific American 1988) by R. T. Bates.*

The ANN here uses what is called the Tunnel Effect. The device is used to implement a simple switching device and is used to mimic thres-holding in this ANN. The model mimic traditional ANN only.

* *'A Feed-forward Artificial Neural Network Based on Quantum Effect Vector-Matrix Multipliers' IEEE transaction on Neural Networks V4N3 by Levy and McGill.*

This model uses techniques from Quantum Mechanics but again just to mimic a traditional Feed-forward ANN.

* *'Learning in Non-Superpositional Quantum Neurocomputers' by R.L. Chrisley. School of Cognitive and Computing Sciences University of Sussex, UK.*

In this model the author try to avoid de-coherence when superposed states cannot be maintained and collapsing occurs when repetitive measurement are done.

6-5 Future Work

*A large simulation will be necessary. We are thinking of implementing the model on a very large set of processors connected via the internet. The process is called Methacomputation. As was mentioned in our work, delays are used to mimic a real time scattering.

*The model physical implementation is still non-feasible in our case as we luck the resources to use actual built Quantum devices. We will proceed by simulating those devices and propose other architecture we think will lead to a good model of Artificial Neural Networks.

Bibliography

[1] Aleksander,I. ,D. , Neural Computing Architecture, the MIT Press, Cambridge, MA, (1989).

[2] Amlani, I., Bernstein, G. H., Lent, C. S., Orlov, A. O., Snider, G. L., Toth, G., Digital Logic Gate Using Quantum-Dot Cellular Automata, Science Vol. 284 No. 5414, pp.1507-1512, (1999).

[3] Arbib, M. A. , Amari, S., Dynamic Interactions in Neural Networks: models and data, Springer-Verlag, New York, (1989).

[4] Ashoori, R. C., Brodsky, M., Pfeiffer, L. N., West, K. W., Zhitenev, N. B., Localization-Delocalization Transition in Quantum Dots, Science Vol. 285 No. 5427, pp. 715-718, (1999).

[5] Barnes, N., Healey, P., Mckee, P., Parallel Interconnection and Neural Network Systems Employing Semiconductor Array Technology and Computer-Generated Holography, Optical and Quantum Electronics v24 n4 spec. pp. 1-3, (1992).

[6] Blow, K. J., Cotter, D., Ellis, A. D., Kelly, A. E., Manning, R.J., Nesset, D., Phillips, I. D., Poustie, A. J., Rogers, D. C., Nonlinear Optics for High-Speed Digital Information Processing, Science Vol. 286 No. 5444, pp. 1523-1528, (1999).

[7] Bose, S., Knight, P. L., Vedral, V., Purification via entanglement swapping and conserved entanglement, Physical Review A, Vol. 60, No. 1, pp. 194-197,(1999).

[8] Brandt, S., Dahmen, H. D., Quantum Mechanics on the Personal Computer Third Edition, SpringerVerlag, New York, (1994).

[9] Brandt, S., Dahmen, H. D., The Picture Book of Quantum Mechanics, John Wiley & Sons, New York, (1985).

[10] Buhrman, H., Van Dam, W., Bounded Quantum Query Complexity, Proceedings of the 14th Annual IEEE Conference on Computational Complexity. pp. 149-156.(1999).

[11] Buhrman, H.. Van Dam, W., Holyer, L., Tapp, A.. Multiparty Quantum Communication Complexity Physical Review A, Volume 60(4). pp. 2717-2741.(1999).

[12] Buttiker. M.. Electron Statistical Effects: Bunches of Photons-Antibunches of Electrons. Science Vol. 284 No. 5414. pp.1507-1512. (1999).

[13] Cercignani. C., the Boltzmann Equation and its Applications. Springer-Verlag. New York. (1989).

[14] Christofferson. R. E.. Basic Principles and Techniques of Molecular Quantum Mechanics. Springer-Verlag. New York. (1989).

[15] Christoph. J.. Eiswirth. M.. Ertl. G.. Strasser. P.. Remote Triggering of Waves in an Electrochemical System. Science Vol. 284 No. 5414. pp.1507-1512. (1999).

[16] Cirac. J. I.. Ekert. A. K.. Macchiavello. C.. Optimal purification of a single qubit. Phys. Rev. Lett. Vol.82. No 4344. pp. 1-7. (1999).

[17] Cirac. J. I., Ekert. A. K.. Huelga. S. F., Macchiavello, C.. Distributed quantum computation over noisy channels. Phys. Rev. A. Vol.59, No 4249. pp.45-57. (1999).

[18] Cleve. R.. Van Dam. W., Nielsen, M.. Tapp. A., Quantum Entanglement and the Communication Complexity of the Inner Product Function, Proceedings of the First NASA International Conference on Quantum Computing and Quantum Communications. C.P. Williams (editor), Lecture Notes in Computer Science 1509, pp. 61-74. (1999).

[19] Cleve, r., Ekert, A, Henderson, L, Macchiavello, C. Mosca, M.. On quantum algorithms. Complexity Vol. 4. No. 1, pp. 33-42, (1998).

[20] Cowan, G. A., Pines, D. Meltzer, D.. Complexity: Metaphors, Models and Reality, Addison Wesley, Reading, MA, (1994).

[21] Davis, P. , The Mind of God, Simon and Schuster, New York, (1992).

[22] Dayhoff, J., Omidvar, O.. Neural Networks and Pattern Recognition. Academic Press, New York, (1998).

[23] Deutsih, D.. 1999, Quantum Theory of Probability and Decisions
Proc R Soc Lond A455. London, (1999).

[24] Dold, A.. Eckerman, B., Stochastic Aspects of Classical and Quantum Systems. Springer-Verlag. New York. (1985).

[25] Dunin-Barkovskii, V. L., Osovets, N. B., Neural Network with Formed Dynamics of Activity, Radiophysics and Quantum Electronics v37 n9, pp. 1-5. (1994).

[26] Dunn, M. H.. Ebrahimzadeh, M., Parametric Generation of Tunable Light from Continuous-Wave to Femtosecond Pulses. Science Vol. 286 No. 5444. pp. 1507-1512. (1999).

[27] Dykman, M. I., Platzman, P. M.. Quantum Computing with Electrons Floating on Liquid Helium. Science Vol. 284 No. 5422. pp. 1967-1969, (1999).

[28] Feagin, J. M., Quantum Methods with Mathematica. Springer-Verlag. New York, (1994).

[29] Feynman, P. R., Leighton, R. B., Sands, M., the Feynman Lectures on Physics. Vol. 4, Addison Wesley, Reading, MA, (1977).

[30] Fischler, M. A., Firschein, O., Intelligence: the Eye, the Brain and the Computer. Addison Wesley, Reading, MA, (1987).

[31] Freeman, J. A., Simulating Neural Networks with Mathematica. Addison Wesley, Readings, MA, (1994).

[32] Gallmann, L., Keller, U., Matuschek, N., Steinmeyer, G., Sutter, D. H., Frontiers in Ultrashort Pulse Generation: Pushing the Limits in Linear and Nonlinear Optics. Science Vol. 286 No. 5444, pp. 1507-1512, (1999).

[33] Graubard, S. R., the Artificial Intelligence Debate, the MIT Press, Cambridge, MA, (1988).

[34] Greiner, Reinhardt, Quantum Electrodynamics. Springer-Verlag, New York, (1994).

[35] Greiner, W., Quantum Mechanics. Springer, New York, (1989).

[36] Greiner, W., Relativistic Quantum Mechanics: Wave Equations. Springer, New York, (1995).

[37] Gutowitz, H., Cellular Automata. MIT/North-Holland, Cambridge, MA, (1991).

[38] Haake, F., Quantum Signature of Chaos. Springer-Verlag, New York, (1992).

[39] Hansen, R. H., Quantum Hall Computers. Proceedings of Physical Phenomena at High Magnetic Fields III.eds.: Fisk, Gorkov & Schrieffer. World Scientific, pp. 61-64, (1999).

[40] Hardy, L., Van Dam, W., Quantum Communication Using a Nonlocal Zeno Effect, Physical Review A, Volume 59(4), pp. 2635-2640, (1999).

[41] Hayken, S., Neural Networks: A Comprehensive Foundation, Macmillan. New York, (1994).

[42] Hecht-Nielson, R., Neurocomputing, Addison Wesley, Reading . MA, (1989).

[43] Hellemans, A., Holograms Can Store Terabytes, But Where?, Science Vol. 286 No. 5444. pp. 1502-1504, (1999).

[44] Hertz. J., Krogh. A., Palmer. R. G., Introduction to the Theory of Neural Computation, Addison_Wesley. Reading. MA. (1991).

[45] Hockney. R. W., Eastwood. J. W., Computer Simulation Using Particles. Mc Graw-Hill. New York. (1989).

[46] Hussey. M.,Fundamentals of Mechanical Vibrations. Macmillan. New York. (1983).

[47] Jennings. Andrew, Horan. Paul. Hegarty. John. Optical Neural Network Using Quantum Well Devices, Applied Optics v33 n8. pp. 1-9, (1994).

[48] Johnson. G.,In the Palaces of Memory. Knopf. New York. (1991).

[49] Johnson. M. K., Troan, E. W., 1998. Linux Application Development. Addison Wesley. Reading. MA. (1998).

[50] Jones. J. A., Mosca, M., Approximate Quantum Counting on an NMR Ensemble Quick Computer Phys. Rev. Lett., 83, pp. 1050-1055,(1999).

[51] Kifer, Y., Random Perturbations of Dynamical Systems, Birkhauser, Boston. (1988).

[52] Knight, P. L., Murao, M., Plento. M. B., Vedral, V., Ion trap quantum gates. decoherence and Error correction. Chaos Solitons & Fractals. Vol. 10. No. 10. pp. 1621-1625. (1999).

[53] Larsen, J.. Optimizing Neural Network Architecture Using Generalization Error Estimators. Radiophysics and Quantum Electronics. v37 n9. pp. 1-5. (1994).

[54] Levy. H. J., McGill. T. C.. A Feedforward Artificial Neural Network Based on Quantum Effect Vector-Matrix Multipliers. IEEE transactions on neural networks v4 n3. pp. 1507-1512. (1993).

[55] Lewenstein. M.. Olko. M., 'Quantum' Neural Networks. Network: Computation in Neural Systems v2 n2. pp.1-4. (1991).

[56] Lockwood. M.. Mind. Brain and the Quantum. Blackwell. New York. (1989).

[57] Luzyanina. T. B.. Synchronization in a Neural Network of Phase Oscillators with Time Delayed Coupling. Radiophysics and Quantum Electronics v37 n8. pp. 1-6. (1994).

[58] Matthew. N.. Stones. R.. Beginning Linux Programming. Wrox Press Ltd.(1996).

[59] Mattuck. R. D.. A Guide to Feynman Diagrams in the Many-Body Problems. Dover. Cambridge. MA. (1976).

[60] Milburn. G. J.. The Feynman Processor. Perseurs Books. Reading. MA. (1998).

[61] Mosea. M., Ekert. A. K.. The Hidden subgroup problem and eigenvalue estimation on a quantum computer Quantum Computing and quantum communication First NASA International Conference. QCQC'98: Palm Springs. California. USA. February 1999 Lecture Notes in Computer Science Vol. 1509. pp 174-188. Springer, New York, (1999) (Ed. C.P. Williams)

[62] Muller, G.. Quantum Mechanics: Symmetries. Springer. New York. (1994).

[63] Murao, M., Jonathan, D., Plenio, M. B., Vedral, V., Quantum telecioning and multiparticle entanglement, Physical Review A, Vol. 59, No. 1, pp. 156-161, (1999).

[64] Nakamura, K., Quantum Chaos: a new paradigm of nonlineardynamics . Cambridge UP, New York, (1993).

[65] Omnes, R., The Interpretation of Quantum Mechanics, Princeton Series in Physics, Princeton, (1994).

[66] Pagels, H., The Dreams of Reasons, Simon and Schuster, New York, (1988).

[67] Papanicolaou, G.,Random Media, Springer-Verlag, New York, (1986).

[68] Penrose, R., The Emperor's New Mind, Oxford, (1989).

[69] Pines, D., The Many-Body Problem, Addison Wesley, Reading, MA, (1961).

[70] Plakhov, A. Yu, Unlearning Dynamics in Hopfield Neural Network, Radiophysics and Quantum Electronics v37 n9, pp.1-8, (1994).

[71] Poyatos, J. F., Garcia, G., de Polavieja, Controlling Quantum Transport by State Synthesis in Ion Traps, Physical Review Letters, vol. 83, no 166, pp.1-6, (1999)

[72] Rigoll, G., Advances in Automatic Speech Recognition by Combining Information Theory and Neural Network Algorithms, International Journal of Quantum Chemistry v42 n4, pp.1-4 (1992).

[73] Rose, F., Into the Heart of the Mind, Harper and Row, New York, (1984).

[74] Schafer, G., Quantum Chromodynamics, Springer, New York, (1994).

[75] Seligman, T. H., Nishioka, H., Quantum Chaos and Statistical Nuclear Physics, Springer-Verlag, New York, (1986).

[76] Seminov, S. A., Dynamics of Circular Neural Network, Radiophysics and Quantum Electronics v37 n9, pp.1-3, (1994).

[77] Slobodkin, L. B., Simplicity and Complexity in the Games of the Intellect, Harvard, Cambridge, MA, (1992).

[78] Smith, C. G., Nanotechnology: Computation Without Current, Science Vol. 284 No. 5414, pp.1507-1512, (1999).

[79] Stapp, H. P., Mind, Matter, and Quantum Mechanics, Springer Verlag, New York, (1993).

[80] Steane, A. M., Quantum Reed-Muller Codes, IEEE Trans. Inf. Theory Vol. 45, pp. 1701-1700 (1999).

[81] Steane, A. M., Efficient fault-tolerant quantum computing, Nature, Vol. 399, pp. 124-126, (May 1999).

[82] Steane, A. M., Letter: Bit of a Hype , Physics World, Vol. 12, No.9, pp. 17-18, (1999).

[83] Stewart, I., Does God Play Dice?: the mathematics of chaos, Blackwell, New York, (1989).

[84] Teerekhov, S. A., Multilayer Neural Network with Randomized Learning: model and applications, Radiophysics and Quantum Electronics v37 n9, pp.1-2, (1994).

[85] Vedral, V., On bound Entanglement Assisted Distillation, Phys. Lett. A 262 , pp. 121-124,(1999).

[86] Vemuri, V., Neural Networks Artificial Neural Networks, Theoretical Concepts. IEEE Computer Society Press,(1990).

[87] Watson, A., Quantum Mechanics: Physicists Tame a Single Photon. Science Vol. 285 No. 5426, pp. 307-309, (1999).