PANCREATIC CANCER DETECTION USING ARTIFICIAL NEURAL NETWORK

Submitted by

Sanhita Das

College Roll No: 079122

University Roll No. 12608912021

University Reg. No. 121260410090 of 2011-12

Under the supervision of

Professor Mohuya Byabartta Kar

In fulfilment for the award of the degree

Of

Master of Technology

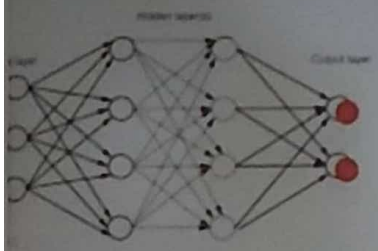
in

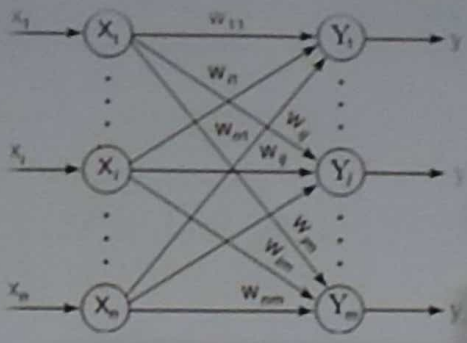
COMPUTER SCIENCE & ENGINEERING

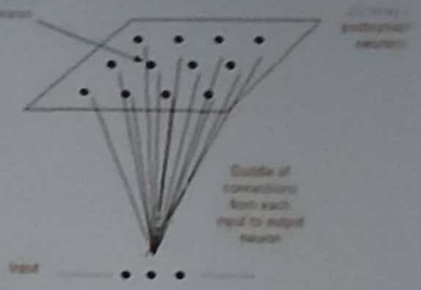
HERITAGE INSTITUTE OF TECHNOLOGY

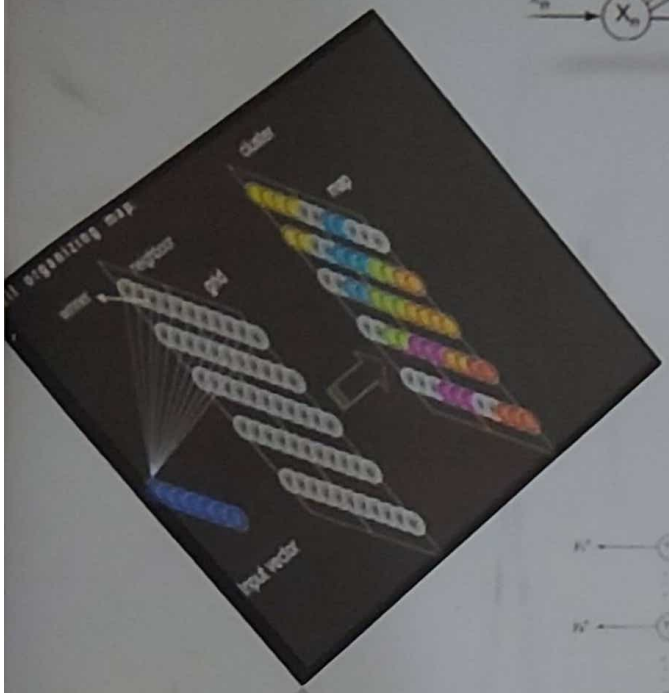
WEST BENGAL UNIVERSITY OF TECHNOLOGY
KOLKATA

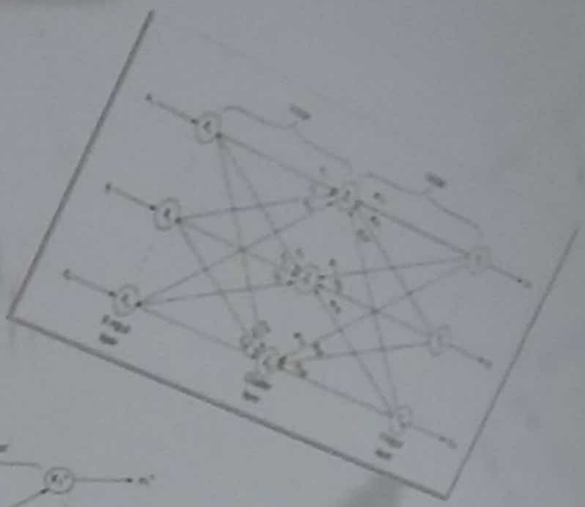
ckpropagation Learning

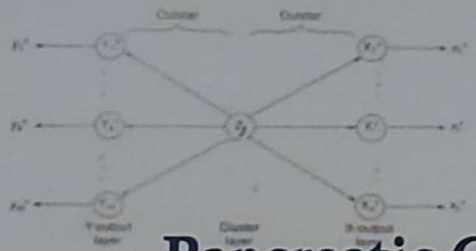






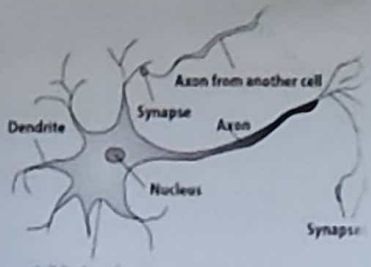


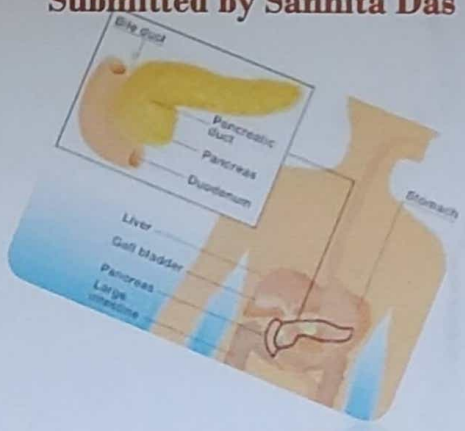




Pancreatic Cancer Detection Using Artificial Neural Network

Submitted by Sanhita Das





PANCREATIC CANCER DETECTION USING ARTIFICIAL NEURAL NETWORK

Submitted by

Sanhita Das

College Roll No: 079122

University Roll No. 12608912021

University Reg. No. 121260410090 of 2011-12

Under the supervision of

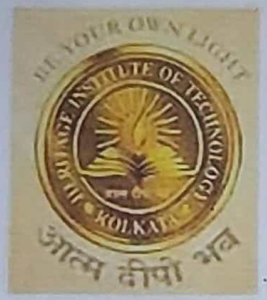
Professor Mohuya Byabartta Kar

In fulfilment for the award of the degree

Of
Master of Technology

in

COMPUTER SCIENCE & ENGINEERING



HERITAGE INSTITUTE OF TECHNOLOGY

WEST BENGAL UNIVERSITY OF TECHNOLOGY
KOLKATA

Acknowledgement

The author remembers with gratitude the constant guidance, suggestions and encouragement forwarded by several respected persons and knows well that is not possible to express her indebtedness for all those valuable assistances by writing some lines. Following conventions, she therefore, acknowledges in this page, the assistance rendered by all the concerned persons, as a token his gratitude.

The author thankfully acknowledges the kind and persistent valuable suggestion, advice, guidance, help and encouragement from Prof. ***Dr. Pranay Chaudhuri,*** Principal, Heritage Institute of Technology for providing me with all the necessary facilities and I must thank my Head of the Department, ***Prof. (Dr.) Subhashis Majumder***, and our Departmental Co-ordinator, ***Prof. Shilpi Saha*** for showing confidence in our innovative ideas and also am grateful to my project mentor ***Professor Mohuya Byabartta Kar*** without which this project would not have been a reality. She is also highly indebted and grateful to her for pulling up her spirits in the most crucial and disheartening moments in a sister manner.

The author is thankful to the head of Computer Science & Engineering for her kind permission to carry out the project under ***Professor Mohuya Byabartta Kar***.

The author is like to convey her gratitude to all professors in Master of Technology course in the Computer Science & Engineering department of Heritage Institute of Technology for excellent teaching that helps to partially complete the project smoothly.

Finally, the author wishes to express her profound gratitude to her parents, brothers, heartily beloved person and friends for providing the necessary atmosphere of understanding and support during the untold amount of hours at studying space and department required for writing this project.

Sanhita Das

Sanhita Das

HERITAGE INSTITUTE OF TECHNOLOGY

WEST BENGAL UNIVERSITY OF TECHNOLOGY

KOLKATA

BONAFIDE CERTIFICATE

Certified that this project report "PANCREATIC CANCER DETECTION USING ARTIFICIAL NEURAL NETWORK" is the bonafide work of "Sanhita Das". Who carried out the project work under my supervision.

SIGNATURE
HEAD OF THE DEPARTMENT
Computer Science & Engg.
Heritage Institute Of Technology

Mahuya Bhattacharya Kar 14/5/14.

SIGNATURE
PROJECT GUIDE
Computer Science &Engg.
Heritage Institute Of Technology

15/5/14

SIGNATURE
EXAMINER(S)

Abstract

Neural network have seen an explosion of interest over the last few years and are being successfully applied across an extraordinary range of problem domains, in areas as diverse as finance, medicine, engineering, geology, physics and biology. Neural network are interesting because of their use in prediction and classification problems. In this project report, Back Propagation NetworkAlgorithm, Kohonen Self Organizing Feature Map Algorithm and Full Counter Propagation Network Algorithm are applied to detect Pancreatic Cancer.

CONTENTS

Abbreviations ----------I

1. Introduction ----------1

1.1 Introduction to neural network----------2

1.2 Introduction to Back Propagation Network ----------3

1.3 Introduction to Kohonen Self-Organizing Maps----------3

1.4 Introduction to Full Counter Propagation Network----------3

2. Artificial Intelligence----------4

3. Back Propagation Network----------6

3.1. Training ----------7

3.2. Algorithm----------8

4. Kohonen Self Organizing Map ----------9

4.1. Architecture----------13

4.2. Training Algorithm----------14

5. Full Counter Propagation Network ----------15

5.1. Architecture----------16

5.2. Full Counter Propagation Operation----------17

5.3. Training Algorithm----------17

6. Problem Definition ----------19

6.1. Pancreatic Cancer Detection----------20

6.2. Methodology----------20

7. Analysis----------22

7.1 Training of parameters using Back Propagation Algorithm------23

7.2 Training of parameters using Kohonen Self Organization Map----24

7.3 Training of parameters using Full counters Propagation Network Algorithm ----------25

8. Scope of Future Work ----------28

9. Limitations----------30

10.Conclusion ----------30

11.Reference----------31

List of Abbreviations

1. **BPN:** Back Propagation Network
2. **KSOM:** Kohonen Self-Organizing Map
3. **SOM:** Self-Organizing Map
4. **1D:** One Dimensional
5. **2D:** Two Dimensional
6. **CPN:** Counter Propagation Network

Chapter 1

Introduction

1. Introduction

1.1 Introduction to Artificial Neural Network

Neural networks are a fascinating concept. The first neural network, "Perceptron", was created in 1956 by Frank Rosenblatt. Thirteen years later in1969, a publication known as *Perceptrons*, by Minsky and Papert, brought a devastating blow to neural network research. It formalized the concept neural networks, but also pointed out some serious limitations in the original architecture. Specifically, it showed that Perceptron could not perform a basic logical computation of a XOR (exclusive-or). This nearly brought neural network research to a halt. Fortunately, research has resumed, and at the time of this writing, neural network popularity has increased dramatically as a tool to provide solutions to difficult problems.

Designed around the brain-paradigm of **Artificial Intelligence**, neural networks attempt to model the biological brain. Neural networks are very different from most standard computer science concepts. In a typical program, data is stored in some structure such as frames, which are then stored within a centralized database, such as with an Expert System or a Natural Language Processor. In neural networks, however, information is *distributed* throughout the network. This mirrors the biological brain, which stores its information (memories) throughout its' synapses.

Neural networks have features that make them appealing to both connectionist researchers and individuals needing ways to solve complex problems. Neural networks can be extremely fast and efficient. This facilitates the handling of large amounts of data. The reason for this is that each node in a neural network is essentially its own autonomous entity. Each performs only a small computation in the grand-scheme of the problem. The aggregate of all these nodes, the entire network, is where the true capability lies. This architecture allows for parallel implementation, much like the biological brain, which performs nearly all of its' tasks in parallel. Neural networks are also fault tolerant. This is a fundamental property. A small portion of bad data or a sector of non-functional nodes will not cripple the network. It will learn to adjust. This does have a limit, though, as too much 'bad' data will cause the network to produce incorrect results. The same can be said of the human brain. A slight head trauma may produce no noticeable effects, but a major head trauma (or a slighthead trauma to the correct spot) may leave the person incapacitated. Also, as pointed out in many studies, we can still understand a sentence when the letters of a word are out of order. This proves that our brain can manage with some corrupt input data.

How do neural networks learn? First off, we must define learning. Biologically, learning is an experience that changes the state of an organism such that the new state leads to an improved performance in subsequent situations. Mechanically, it is similar: Computational methods for acquiring and organizing new knowledge that will lead to new skills. Before the network can become useful, it must learn about the information at hand. After training, it can then be used for pragmatic purposes. In general, there are two flavours of learning:

• ***Supervised Learning***, The correct answers are known and this information is used to train the network for the given problem. This type of learning utilizes both input vectors and output vectors. The input vectors are used to provide the starting data, and the output vectors can be used to compare with the input vectors to determine some error. In a special type of supervised

learning, *reinforcement learning*, the network is only told if its output is right or wrong. Back-propagation algorithms make use of this style.

• ***Unsupervised Learning***, The correct answers are not known (or just not told to the network). The network must try on its own to discover patterns in the input data. The input vectors are used solely. Output vectors generated will not be used to learn from. Also and possibly most importantly: no human interaction is needed for unsupervised learning. This can be an extremely important feature, especially when dealing with a large and/or complex data set that would be time-consuming or difficult to a human to compute. Self-Organizing Maps utilize this style of learning.

1.2 Introduction to Back Propagation Network

One of the most popular NN algorithms is **Back Propagation Algorithm**. Rojas [2005] claimed that BP algorithm could be broken down to four main steps. After choosing the weights of the network randomly, the back propagation algorithm is used to compute the necessary corrections. The algorithm can be decomposed in the following four steps:

i) Feed-forward computation
ii) Back propagation to the output layer
iii) Back propagation to the hidden layer
iv) Weight updates

The algorithm is stopped when the value of the error function has become sufficiently small.

1.3 Introduction to Kohonen Self Organization Map

Kohonen Self-Organizing Maps is a type of neural network. They were developed in 1982 by Tuevo Kohonen, a professor emeritus of the Academy of Finland. Kohonen Self-Organizing Maps are aptly named. "Self-Organizing" is because no supervision is required. KSOMs learn on their own through unsupervised competitive learning. "Maps" is because they attempt to map their weights to conform to the given input data. The nodes in a KSOM network attempt to become like the inputs presented to them. In this sense, this is how they learn. They can also be called "Feature Maps", as in Self-Organizing Feature Maps. Retaining principle 'features' of the input data is a fundamental principle of KSOMs, and one of the things that makes them so valuable. Specifically, the topological relationships between input data are preserved when mapped to a KSOM network. This has a pragmatic value of representing complex data. Another intrinsic property of KSOMs is known as vector quantization. This is a data compression technique. KSOMs provide a way of representing multidimensional data in a much lower dimensional space – typically one or two dimensions.

1.4 Introduction to Full Counter Propagation Network

It is multilayer networks based on a combination of input, clustering and output layers. It is introduced by R. Hecht- Nielsen in 1987. Full Counter Propagation and Forward only Counter propagation is two type of **Counter Propagation**. Full Counter propagation is an efficient method to represent a large number of vector pairs by adaptively constructing a look up table. It produces an approximation into output relationship. Forward only counter propagation is simplified version of the counter propagation. It produces a mapping from input to output.

Chapter 2

Artificial Intelligence

2. Artificial Intelligence

Man is called Homo sapiens, human the wise. We think of ourselves as being wise and intelligent. For millennia, we've tried to understand how to think, perceive, understand, act and anticipate. When do we say someone is intelligent? Is it that they are very good in mathematics or quantum physics? These people are very good at making abstract models of the world. But what's about a musician? He can be said to be intelligent because of his creativity. Intelligence encompasses just about every from human activity.

A precise definition of intelligence is unavailable. It is probably explained best by discussing some of the aspects. In general, intelligence has something to do with the process of knowledge and thinking, also called cognition. These mental processes are needed for, i.e., solving a mathematical problem or playing a game of chess. One needs to possess a certain intelligence to be able to do these tasks. Not only the deliberate thought processes are part of cognition, also the unconscious processes like perceiving and recognizing an object belong to it. Usually, cognition is described as a sort of awareness, enabling us to interact with the environment, think about your consequences of actions and being in control of these processes. It is also associated with our 'inner voice' that expresses our deliberate thoughts and our emotions. The field of Artificial Intelligence tries to gain more insight in intelligence by building models of intelligent entities. Principles from deferent scientific fields, like mathematics, psychology, neuroscience, biology and philosophy are incorporated to design such models.

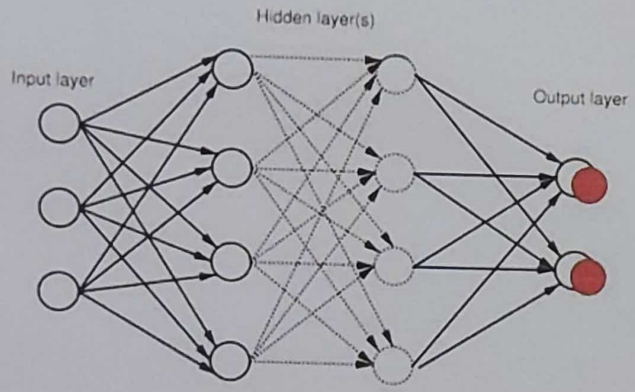
Chapter 3

Back Propagation Network

3. Back Propagation Network

Back propagation is a supervised learning process. It is multilayer, feed forward neural networks consisting of processing elements with continuous differentiable activation functions. For a given set of training input output pair, this algorithm provides a procedure for changing the weight in a BPN to classify the given input pattern correctly. The basic concept for this weight update algorithm is simply the gradient-decent method as used in the case of simple perceptron network with differentiable unit. The training of BPN is done in three stages – the feed-forward of the input training pattern, the calculation and back-propagation of the error, and updating of weights. The inputs are set to the BPN and the output obtained from the net could be (0, 1).It can theoretically perform "any" input-output mapping. It can learn to solve linearly inseparable problems.

Backpropagation Learning



3.1 Training

A back prop network consists of at least three layers of units:

- An Input Layer
- At least one intermediate Hidden Layer
- An Output Layer

Typically units are connected in a feed over fashion with input units fully connected to units in the hidden units fully connected to units the output layer.

When a back prop network is cycled, an input pattern is propagated forward to the output units through the intervening input-to-hidden and hidden –to-output weights.

Output of a back prop network is intercepted as a classification decision.

With BPNs, learning occurs during a training phase.

The steps follows during learning are:

- Each input pattern in a training set is applied to the input units and then propagation forward.
- The pattern of activation arriving at the output layer is compared with the correct (associative) output pattern to calculate an error.
- The error for each such target output pattern is then back-propagated from the outputs to the inputs in order to appropriately adjust the weights in each layer of the network.
- After a back propagation network has learned the correct classification for a set of inputs, it can be tested on a second set of inputs to see how well if classifies untrained patterns.

3.2 Algorithm

- Initialize each wi to some small random value.
- Until the termination condition is met, Do
  For each training example <(x1,…xn),t> Do
  - Input the instance (x1,…,xn) to the network and compute the network outputs ok
  - For each output unit k
    $\delta k = ok(1-ok)(tk-ok)$
  - For each hidden unit h
    $\delta h = oh(1-oh)\ \Sigma k\ wh,k\ \delta k$
  - For each network weight w,j Do
  - $wi,j = wi,j + \Delta wi,j$ where
    $\Delta wi,j = \eta \delta j\ xi,j$

Chapter 4
Kohonen Self Organizing Map

4. Kohonen Self Organizing Map

The SOM encompasses a number of characteristics which bear similarities to the way the human brain works, or at least be thought to work. In fact, the notion of having available a set of neurons which through learning experiences specialize in the identification of certain types of patterns is consistent with current research on the human brain. The idea that certain parts of the brain are responsible for specific skills and tasks is remarkably similar to the SOM principles. The concept of organizing information spatially, where similar concepts are mapped to adjacent areas, constitutes a trademark of the SOM and it is believed to be one of the paradigms of the functioning the human brain.

The basic idea of the SOM is to make available a certain amount of classificatory resources (we will call them neurons, although the term units is also widely used) which are organized based on the patterns available for classification (which will be called here input patterns). A very relevant characteristic of the SOM, which in fact constitutes a trademark, is the development of an ordered network in which nearby neurons will share similarities, this way patterns which are similar will activate similar areas in the SOM. Thus it can be said that different parts of the SOM will be activated by specific types of input patterns, leading to a representation based upon global organization and local specialization. The SOM is trained iteratively through a large number of epochs. An epoch is defined as the processing of all the input patterns once, thus each input pattern will be processed as many times as the number of epochs.

A SOM is single layer neural network, where the neurons are set along an n-dimensional lattice. In most applications this lattice is 2-dimensional and rectangular, though many applications use hexagonal lattice, and 1, 3, or more dimensional spaces. In this lattice we can define neighbourhoods in what we call the output space, as opposed to the input space of the data patterns. The concepts of input space and output space are rather important as they constitute the centre of the SOM's activity. In fact, the can be thought of as a tool which maps (projects) the vectors in the input space onto the output space trying to preserve the topological relations observed in the input space.

Each neuron, being an input layer neuron, has as many weights (which are often also called coefficients) as the input patterns, and can thus be regarded as a vector in the same space as the patterns. When we train or use a SOM with a given input pattern, we calculate the distance between that specific pattern and every neuron in the network. We then select the neuron that is closest as the winning neuron, and say that the pattern is mapped onto that neuron or that the neuron as won the representation of that input pattern. As a consequence the neuron will move towards the input pattern position in order to improve its representation. The extent of this movement is controlled by a parameter usually referred to as learning rate.

In order to preserve the topology in the output space it is essential to correct the position of the wining neuron but also the position of its neighbouring neurons. If the SOM has been trained successfully, then patterns that are close in the input space will be mapped to neurons that are close (or the same) in the output space, and vice-versa. Thus, as already said, the SOM is "topology preserving" in the sense that (as far as possible) neighbourhoods are preserved through the mapping process. Generally, no matter how much we train the network, there will always be some difference between any given input pattern and the neuron it is mapped to. This is a situation identical to vector quantization, where there is some difference between a pattern and its code-book vector representation. Thus, we refer to this difference as the quantization error, and

use it as a measure of how well our neurons represent the input patterns. In other words the quantization error in calculated by summing all the distances between each input pattern and the neuron to which is mapped, giving a notion of the quality of the representation achieved by the SOM.

There are two different models for the self-organizing map:

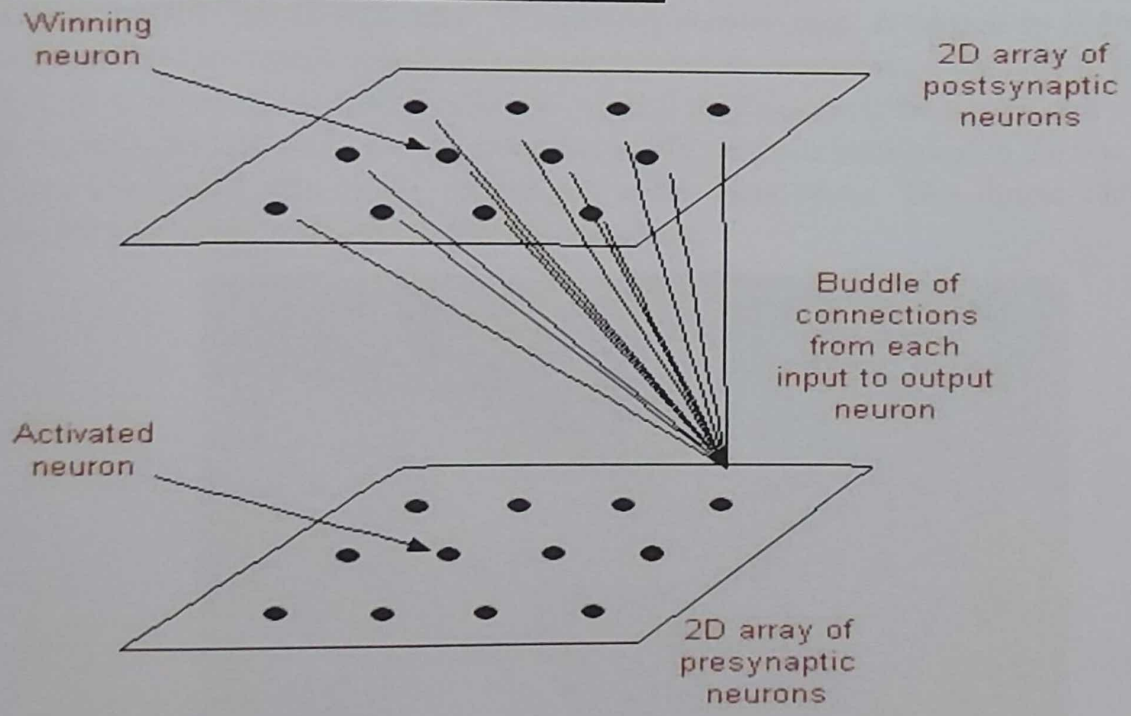
(1) Willshaw-von der Malsburg model.

(2) **Kohonen model**.

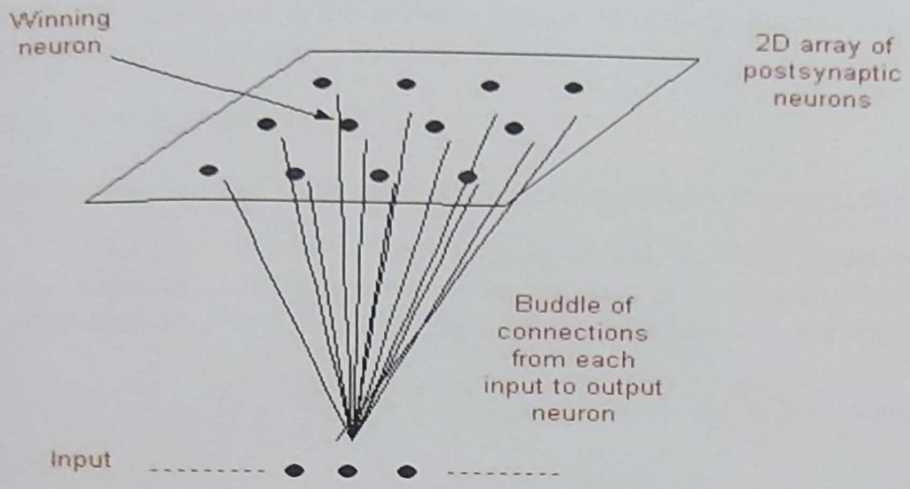
In both models the outputs neurons are placed in a 2D lattice.They differ in the way input is given: In the Willshaw-von der Malsburg model the input is also a 2D lattice of equal number of neurons. In the Kohonen model there isn't any input lattice, but an array of input neurons. In the Willshaw-von der Malsburgmodel the input is also a 2D lattice of equal number of neurons; In the Kohonen model there isn't any input lattice, but a 1D or 2D array of input neurons.

Schematically the models are shown billow:

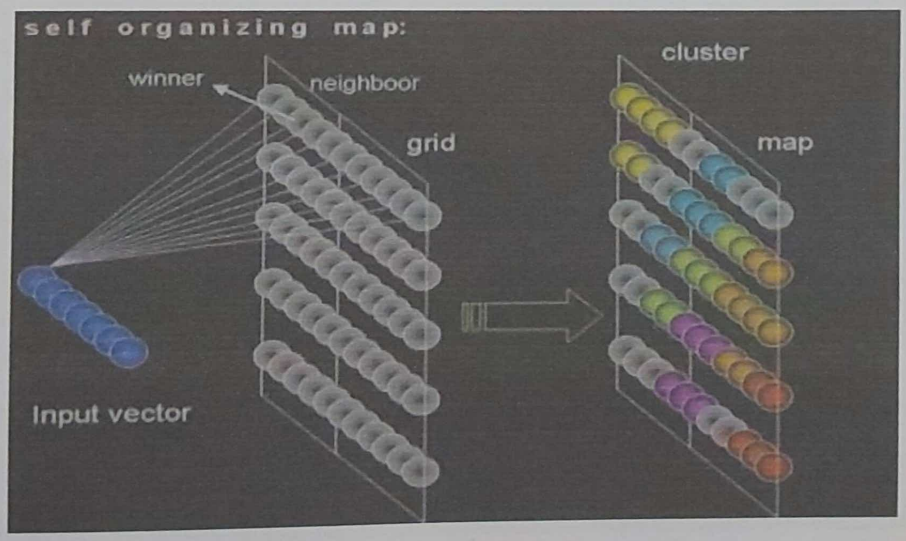
(1) Willshaw – von der Malsburg model:



(2) Kohonen model:



The Kohonen model provides a topological mapping. In a topological preservation map, units located physically next to each other will respond to classes of input vector that are likewise located next to each other. The self-organization map, developed by kohonen, groups the input data into cluster which are commonly used for unsupervised training. In the SOM, all the units in the neighborhood that receive positive feedback from the wining unit participate in the learning process. Even if a neighborhood unit's weight is orthogonal to the input vector, its weight vector will still change in response to the input vector. This simple addition to the competitive process is account for the order mapping.



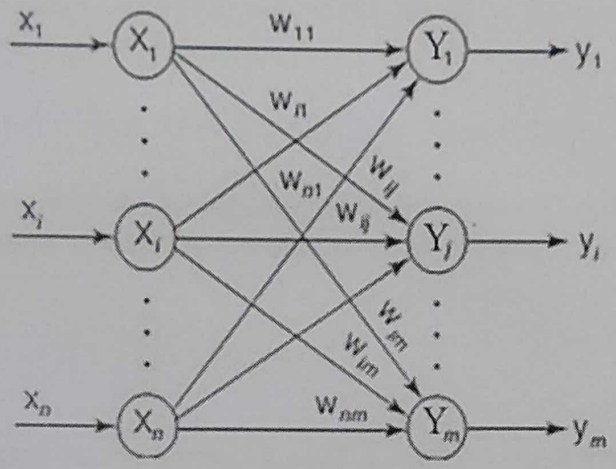
The topology structure property is observed in the brain, but is not found any other artificial neural network except SOM. There are 'm' cluster unites, arrenged in 1or 2-D array and the input signals are n-tuples. The weight vector for a cluster unit is the exampler of the input pattern associated with the cluster. In self organizing process, the cluster unit whose weight vector matches the input pattern closely is selected as the winner. The winning and the

neibourhood units update their weights. The neibourhood units weight vectors are not close to the input pattern but differ to some extent. Hence, the connection weightes do not multiply the signal sent from the input unites to the cluster units.

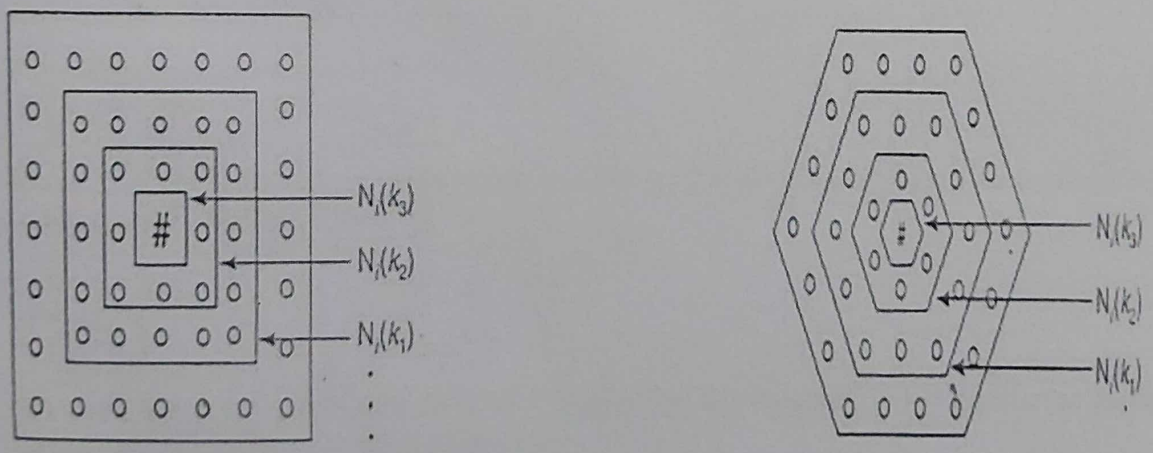
4.1 Architechture

The architechture of Kohonen Self Organizing Map is shown in this Figure.

Here are 'n' units in the input layer and 'm' units in the output layer. Here the wining unite isidentified by using either dot product or Euclidean distance method and the weight updatation using Kohonen learning rules is performed over the wining cluster unit.



The key principal for feature map formation is the training should take place over an extended region of the network centered on maximally active mode. Hence ,the concept of neighbourhood should be defined for the net. In this Figure



Neighbourhood sceheme for SOMs (Rectangular and Hexagonal)

Two neibourhood schemas are shown based on 2-D arrays in the form of rectengular and hexagonal grids. In all cases, neighbourhood are shown delimited with respect to a shaded unit at distences of 1,2 and 3 arrays from this node. The winning unit is indicated by '#' and the other units are indicated by 'o'. In the rectengular grid, each unit has eight nearest neighbourhood but only six will be there in the hexagonal grid and two in the linear grid.

4.2 Training Algorithm

Initially, the weight and learning rate are set. The inpur vectors to be clustered are presented to the network. Once the input vector are given, based on the winner unit is calculated either by Euclidian distance or sum of products method.based on the winer unit selection, the weights are updated for that particular winner unit.An epoch is said to be completed once all the input vectors are presented to the network. By updating the learning rule several epochs of training may be performed.

Step1: Initialization

Set initial synaptic weights to small random values, say in an interval [0,1], and assign a small positive value to the learning rate parameter α .

Step 2: Activation and Similarity Matching

Activation the Kohonen network by applying the input vector X, and find the winner-takes-all (best matching) neuron jX at iteration p, using the minimum-distance Euclidean criterion

$$j_X(p) = \min_j \left\| \mathbf{X} - \mathbf{W}_j(p) \right\| = \left\{ \sum_{i=1}^{n} [x_i - w_{ij}(p)]^2 \right\}^{1/2},$$

Where n is the number of neurons in the input layer, and m is the number of neurons in the Kohonen layer.

Step3: Learning

Update the synaptic weights

$$w_{ij}(p+1) = w_{ij}(p) + \Delta w_{ij}(p)$$

Where Δwij (p) is the weight correction at iteration **p**. The weight correction is determined by the competitive learning rule:

$$\Delta w_{ij}(p) = \begin{cases} \alpha [x_i - w_{ij}(p)], & j \in \Lambda_j(p) \\ 0, & j \notin \Lambda_j(p) \end{cases}$$

where α is the learning rate parameter, and Λj(p) is the neighborhood function centered around the winner-takes-all neuron jX at iteration p.

Step4: Iteration

Increasing iteration p by one, go back to Step2 and continue until the minimum-distance Euclidean creation is satisfied, or no noticeable changes occur in the feature map.

Chapter 5

Full Counter Propagation Network

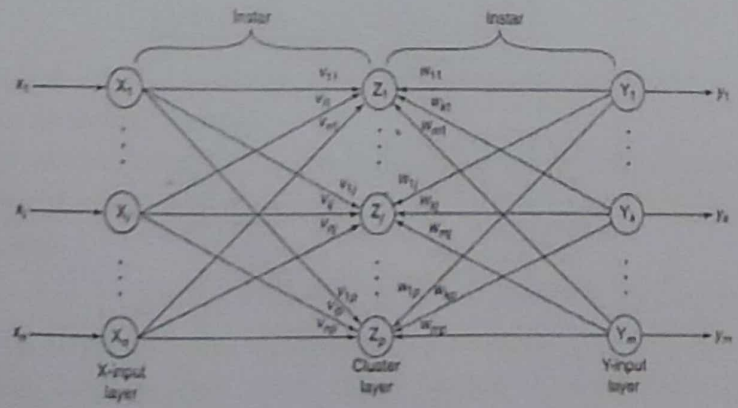
5. Full Counter Propagation Network

Counter propagation neural networks were developed by Robert Hecht– Nielsen as a means to combine an unsupervised Kohonen layer with a teachable output layer known as **Grossberg layer**. The operation of this network type is very similar to that of the Learning Vector Quantization (LVQ) network in that the middle (Kohonen) layer acts as an adaptive look-up table.

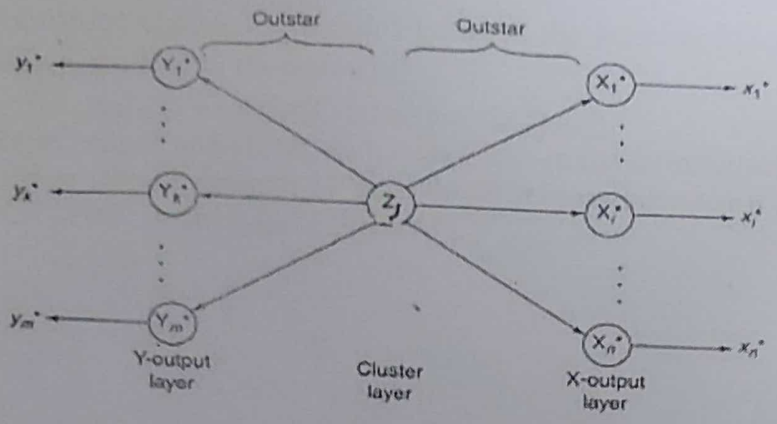
The structure of this network type is characterized by the existence of three layers: an input layer that reads input patterns from the training set and forwards them to the network, a hidden layer that works in a competitive fashion and associates each input pattern with one of the hidden units, and the output layer which is trained via a teaching algorithm that tries to minimize the mean square error (MSE) between the actual network output and the desired output associated with the current input vector. In some cases a fourth layer is used to normalize the input vectors but this normalization can be easily performed by the application (i.e., the specific program implementation), before these vectors are sent to the Kohonen layer. Regarding the training process of the counter propagation network, it can be described as a two-stage procedure: in the first stage the process updates the weights of the synapses between the input and the Kohonen layer, while in the second stage the weights of the node between the Kohonen and the Grossberg layer are updated.After the first phase of the training, each hidden-layer neuron is associated with a subset of input vectors. The training process minimized the average angle difference between the weight vectors and their associated input vectors. In the second phase of the training, adjust the weights in the network's output layer in such a way that, for any winning hidden-layer unit, the network's output is as close as possible to the desired output for the winning unit's associated input vectors. The idea is that when later use the network to compute functions, the output of the winning hidden-layer unit is 1 and the output of all other hidden-layer units is 0.

5.1 Architechture

The general structure of full CPN is shown bellow



First Phase of training of full CPN



Second Phase of training of full CPN

Here,

x = input training vector x = ($x_1,\ldots,x_i,\ldots x_n$)

y = target output corresponding to input x,y = ($y_1,\ldots,y_k,\ldots,y_m$)

z_j= the output of cluster layer unit z_j

v_{ij}= weight from X-input layer unit Xi to cluster layer unit z_j

w_{ij}= weight from Y-input layer unit Yi to cluster layer unit z_j

5.2 Full Counter Propagation Operation

- Present input to network.
- Calculate output of all neurons in Kohonen Layer.
- Determine winner (neuron with maximum output)
- Set output of winner to 1 (others to 0)
- Calculate output vector

5.3 Training Algorithm

1. Randomly select a vector pair (x,y) from the training set.
2. Normalize the input vector x by dividing every component of x by the magnitude $\|x\|$,where

$$\|x\| = \sqrt{\sum_{j=1}^{n} x_j^2}$$

3. Initialize the input neuron with the normalized vector and compute the activation of the linear hidden-layer units.
4. In the hidden layer, determine the unit W with the largest activation (the winner).
5. Adjust the connection weights between W and all N input-layer units according to the formula

$$W_{wn}^{H}(t+1) = W_{wn}^{H}(t) + \alpha(t)(xn - W_{wn}^{H}(t)).$$

6. Repeat step 1 to 5 unit all training pattern have been processed once.
7. Repeat Step 6 until each input pattern is consistency associated with the same competitive unit.
8. Select the first vector pair in the training set.
9. Reset steps 2 to 4.

10. Adjust the connection weights between the winning hidden-layer unit and all M output units according to the equation

$$W^{o}_{mW}(t+1) = W^{o}_{mW}(t) + \beta(y_m - W^{o}_{mW}(t)).$$

11. Repeat step 9 and 10 for each vector pair in the training set.
12. Repeat steps 8 through 11 until the difference between the desired and the actual output falls below an acceptable threshold.

Chapter 6
Problem Definition

6. Problem Definition

The following are description of application utilizing Kohonen Self Organizing Map, Full Counter Propagation Network Algorithm and Back Propagation NetworkAlgorithm. It is an approach to Pancreatic Cancer Detection using KSOM, Full CPN, BPN.

6.1 Pancreatic Cancer Detection

Pancreatic cancer is of malignant type neoplasm originates from transformed cells arising in tissues form the pancreas. Pancreas is a 6-inch long spongy organ located behind the stomach in the back of the abdomen. The pancreas contains exocrine and endocrine glands that create pancreatic juices, hormones, and insulin. Pancreatic juices, or enzymes, made by the exocrine glands are released into the intestines by way of a series of ducts in order to help digest fat, proteins, and carbohydrates. The endocrine cells are arranged in small clusters called islets of Langerhans, which release insulin and glucagon into the bloodstream. These two hormones manage levels of sugar in the blood. When they are not working properly, the result is often diabetes. The abnormal pancreas tissues continue dividing and form lumps or masses of tissue called tumours. Tumours then interfere with the main functions of the pancreas. If a tumour stays in one spot and demonstrates limited growth, it is generally considered to be benign.

The aim of this approach is to show that neural networks can make an accurate individualized prognosis of a patient given his or her particular condition. The need of designing a system that would help to diagnose of pancreatic cancer cannot be over emphasized. Here the practical application of human intelligence in the health sector. Here provides the information of the patients whether they have pancreatic cancer or not. This diagnosis is functioning depending on some symptoms which has been taken from their previous medical records and physician and train these symptoms through neural network to detect which patient is suffering from pancreatic cancer or he/she might be suffering from pancreatic cancer or may not suffer at all.

6.2 Methodology

Collected data set included the patient's previous state of health, living condition and other medical conditions. Back Propagation Network, Kohonen Self Organization Map,Full Counter Propagation Network are proposed to diagnosis PC diseases.

Dataset

Table 1 shows a part of the used input data set. This data set provides 11 symptoms and 3 categories of pancreatic diseases. The dataset contains a measured feature of 10 patients. The set of symptoms are:

S= {Weight Loss, Jaundice, Irritability, Gallbladder enlargement, Deep venous thrombosis (DVT), Swelling lymph, Diabetes mellitus, Loss of appetite, Pain in upper abdomen, Alcoholic Stool & steatorrhea and Fatly tissue abnormalities}. The set of diagnosis are

D = {detected, might be detected and not detected}.

Table 1: MEDICAL KNOWLEDGEBASE TABLE

Weight loss	jaundice	Irritability	GB enlargement	DVT	Swelling Lymph	DM	Loss of appetite	Point in upper abdomen	AS & abdomen	Faulty tissue abnormalities	Result
.60	.45	.30	.10	.55	.35	.15	.26	.80	.72	.18	Detected
.62	.18	.17	.78	.14	.82	.50	.00	.59	.36	.47	Might be detected
.32	.73	.25	.55	.61	.13	.20	.69	.33	.49	.86	Might be detected
.28	.24	.39	.36	.23	.70	.55	.63	.08	.17	.63	Not detected
.43	.59	.73	.29	.70	.37	.13	.68	.75	.55	.43	Detected
.56	.44	.57	.63	.34	.19	.41	.53	.69	.72	.38	Detected
.72	.52	.60	.19	.42	.40	.21	.63	.30	.39	.52	Not detected
.63	.13	.11	.47	.69	.23	.51	.12	.24	.10	.63	Might be detected
.33	.63	.49	.72	.77	.62	.24	.38	.21	.43	.40	Detected
.78	.24	.54	.41	.64	.78	.43	.55	.30	.18	.23	Detected

Chapter 7
Analysis

7. Analysis

7.1 Training of perameters using Back Propagation Algorithm:

Initially, the weight are set and learning rate α =0.25. The training of BPN is done in three stages – the feed-forward of the input training pattern, the calculation and back-propagation of the error, and updating of weights.

```
while (e<=3)
    e
    for i=1:2
        for j=1:2
            temp=temp+(v(j,i)*x(j));
        end
        zin(i)=temp+vb(i);
        temp1=e^(-zin(i));
        fz(i)=(1/(1+temp1));
        z(i)=fz(i);
        fdz(i)=fz(i)*(1-fz(i));
        temp=0;
    end

    for k=1
        for j=1:2
            temp=temp+z(j)*w(j,k);
        end
        yin(k)=temp+wb(k);
        fy(k)=(1/(1+(e^-yin(k))));
        y(k)=fy(k);
        temp=0;
    end

    for k=1
        fdy(k)=fy(k)*(1-fy(k));
        delk(k)=(t(k)-y(k))*fdy(k);
    end

    for k=1
        for j=1:2
            dw(j,k)=alpha*delk(k)*z(j);
        end
        dwb(k)=alpha*delk(k);
    end

    for j=1:2
        for k=1
            delin(j)=delk(k)*w(j,k);
        end
        delj(j)=delin(j)*fdz(j);
    end

    for i=1:2
        for j=1:2
            dv(i,j)=alpha*delj(j)*x(i);
        end
```

```
        dvb(i)=alpha*delj(i);
    end

    for k=1
        for j=1:2
            w(j,k)= w(j,k)+dw(j,k);
        end
        wb(k)=wb(k)+dwb(k);
    end
    w,wb

    for i=1:2
        for j=1:2
            v(i,j)=v(i,j)+dv(i,j);
        end
        vb(i)=vb(i)+dvb(i);
    end
    v,vb
    te(e)=e;
    e=e+1;
end
```

7.1.1 Result

eight loss	Jaundice	Irritability	GB enlargement	DVT	Swelling Lymph	DM	Loss of appetite	Point in upper abdomen	AS & abdomen	Faulty tissue abnormalities	Result(J)
0	.45	.30	.10	.55	.35	.15	.26	.80	.72	.18	1
2	.18	.17	.78	.14	.82	.50	.00	.59	.36	.47	0
2	.73	.25	.55	.61	.13	.20	.69	.33	.49	.86	0

MATLAB R2007b is used to evaluate the performance of the propose networks. Here for first input, J=1, so Disease DETECTED. For next two input, J=0, so Disease MIGHT BE DETECTED and so on.

7.2 Training of perameters using Kohonen Self Organization Map

Initially, the weight are set and learning rate are set as 0.6 . The inpur vectors to be clustered are presented to the network. Once the input vector are given, based on the winner unit is calculated either by Euclidian distance or sum of products method. Based on the winer unit selection, the weights are updated for that particular winner unit.An epoch is said to be completed once all the input vectors are presented to the network. By updating the learning rule several epochs of training is performed.

```
while(e<=4)
    i=1;
```

```
j=1;
k=1;
m=1;
disp('Epoch =');
e
while(i<=2)
    for j=1:11
        temp=0;
        for k=1:2
            temp= temp + ((w(k,j)-x(i,k))^2);
        end
        D(j)= temp
    end
    if(D(1)<D(2))
        J=1;
    else
        J=2;
    end
    disp('The winning unit is ');
    J
    disp('Weight updation');
    for m=1:2
        w(m,J)=w(m,J) + (alpha(e) * (x(i,m)-w(m,J)));
    end
    w
    i=i+1;
end
```

7.2.1 Result

Weight loss	jaund ice	Irrita bility	GB enlargem ent	DV T	Swellin g Lymph	DM	Loss of appetite	Point in upper abdome n	AS &abdo men	Faulty tissue abnormal ities	Result (ans)
.60	.45	.30	.10	.55	.35	.15	.26	.80	.72	.18	0.300
.62	.18	.17	.78	.14	.82	.50	.00	.59	.36	.47	0.150
.32	.73	.25	.55	.61	.13	.20	.69	.33	.49	.86	0.075
.28	.24	.39	.36	.23	.70	.55	.63	.08	.17	.63	0.035

MATLAB R2007b is used to evaluate the performance of the proposed networks. Here for first input answer comes 0.300 .It is much greater than 0.1. In linguistic term it can be represent as Disease DETECTED, for second two inputs it comes 0.150 and 0.075. It is near to 0.1, it can be represented as Disease MIGHT BE DETECTED and for third person answer comes 0.0350, which is much smaller than 0.1so, the person is Disease NOT DETECTED and so on.

7.3 Training of perameters usingFull counter Propagation Network Algorithm

Initially, the weight are set and learning rate α and β are set as 0.4, 0.3. The inpur vectors to be clustered are presented to the network.The clusters are formed using Euclidian Distance

method or dot product method. The weights from the cluster layer unite to the output units are tuned to obtain the desired response.

```
while(e<=3)
    m=1;
    n=1;
    for j=1:2
        temp=0;
        for k=1:4
            temp= temp + ((v(k,j)-x(k))^2);
        end
        for k=1:2
            temp= temp + ((w(k,j)-y(k))^2);
        end
        D(j)=temp
    end
    if(D(1)<D(2))
        J=1;
        disp('disease detected ')
    else
        J=2;
        disp('disease might be detected ')
    end
    disp('The winning unit is ');
    J
    disp('Weight updation');
    for m=1:4
        v(m,J)=v(m,J) + (alpha(e) * (x(m)-v(m,J)));
    end
    v
    for n=1:2
        w(n,J)=w(n,J) + (beta(e) * (y(n)-w(n,J)));
    end
    w

    temalpha=alpha(e);
    tembeta=beta(e);
    tema=a(e);
    temb=b(e);
    oe=e;
    te(e)=e;
    e=e+1;
    te(e)=e;
    te1(oe)=oe;
    alpha(e)=(0.5*temalpha);
    alpha
    beta(e)=(0.5*tembeta);
    beta

    disp('for Weight updation from cluster unit to output unit');
    for m=1:4
        v(m,J)=v(m,J) + (alpha(e) * (x(m)-v(m,J)));
    end
    v
    for n=1:2
        w(n,J)=w(n,J) + (beta(e) * (y(n)-w(n,J)));
    end
    w
    for m=1:4
        t(m)=t(m) + (b(oe) * (x(m)-t(m)));
```

```
    end
    t
    for n=1:2
        u(n)=u(n) + (a(oe) * (y(n)-u(n)));
    end
    u

    a(e)=(0.5*tema);
    b(e)=(0.5*temb);
    a
    b
end
```

3.1 Result

Weight loss	jaund ice	Irrita bility	GB enlarge ment	DVT	Swellin g Lymph	DM	Loss of appetite	Point in upper abdome n	AS &abdo men	Faulty tissue abnormal ities	Result (J)
60	.45	.30	.10	.55	.35	.15	.26	.80	.72	.18	1
62	.18	.17	.78	.14	.82	.50	.00	.59	.36	.47	2
32	.73	.25	.55	.61	.13	.20	.69	.33	.49	.86	2

MATLAB R2007b is used to evaluate the performance of the propose networks. Here for first input, J=1, so Disease DETECTED. For next two input, J=2, so Disease MIGHT BE DETECTED and so on.

Chapter 8
Scope of Futute Work

8. Future Scope

Neural Networks welcomes high quality that contributes to the full range of neural networks research. From behavioral and brain modeling, learning algorithms, through mathematical and computational analyses, to engineering and technological applications of systems that significantly use neural network concepts and techniques.

This uniquely broad range facilitates the cross-fertilization of ideas between biological and technological studies, and helps to foster the development of the interdisciplinary community that is interested in biologically-inspired computational intelligence. In includes fields like psychology, neurobiology, computer science, engineering, mathematics, and physics.

ANN has 5 sections: Cognitive Science, Neuroscience, Learning Systems, Mathematical and Computational Analysis, Engineering and Applications. Using ANN sufficient condition for the existence, uniqueness and global asymptotic stability of the equilibrium point for the class of delayed neural networks under the parameter uncertainties of the neural system can be detected (A patient is detected as a cancerarous / might be detected cancerous /not detected cancerous in the present S/W module through different algorithms)

This patient who under goes through different kind of treatments at different time slots can be again clustered / verified for getting Cured / Might be cured / Cannot be Cured

The parametric analysis of the uncertain parameters needs to be done which will enable a machine to gives the sequence of treatment at different time slots and predict in prior the condition of the patient at a specific time slot.

9. Limitations

We have a data set, but limited to few parameters used in earlier papers in ANN .We are limited to Real dataset so that parametric analysis could be done. Limited time to generate the treatment module for getting the patient cured. Some simulation results are also given to show the applicability and advantages of our result.

10. Conclusion

In this project report we have presented a neural network based approach for pancreatic cancer diagnosis. Pancreatic cancer detection in its early stage is the key of its cure. The automatic diagnosis of pancreatic cancer is an important, real-world medical problem. In this project report it has shown how BPN KSOM, Full CPN are used in actual clinical diagnosis of pancreatic cancer. Neural network model, a diagnostic system that performs at an accuracy level is constructed here. In this work, the performance of neural network structure was investigated.

Reference:

1. http://en.wikipedia.org/wiki/Backpropagation
2. http://page.mi.fu-berlin.de/rojas/neural/chapter/K7.pdf
3. http://www.share-pdf.com/f0dbb32c91d840d1a5135970e4f1c962/aKopHDcN.pdf
4. http://en.wikipedia.org/wiki/Self-organizing_map
5. http://shy.am/writings/Kohonen_SOMs.pdf
6. http://edugi.uji.es/Bacao/SOM%20Tutorial.pdf
7. http://en.wikipedia.org/wiki/Counterpropagation_network
8. http://eprints.qut.edu.au/2876/1/Aiaa-35572876.pdf
9. http://www.ncbi.nlm.nih.gov/pmc/articles/PMC331947/pdf/nar00044-0283.pdf
10. TanayaSen, Sujit Das "*An Approach to Pancreatic Cancer Detection using Artificial Neural Network*" Proc. of the Second Intl. Conf. on Advances in Computer, Electronics and Electrical Engineering -- CEEE 2013.
11. S.N. Sivanandam, S.N. Deepa "Principal of Soft Computing" Second Edition.
12. S.N. Sivanandam, S Sumathi, S.N. Deepa "Introduction To Neural Networks Using Matlab 6.0"