جمهورية مصر العربية
معهد التخطيط القومى

سلسلة قضايا التخطيط والتنمية
رقم (١٢٢)

Artificial Neural Networks Usage for Underground Water Storage & River Nile In Toshka Area

ديسمبر ١٩٩٨

Artificial Neural Networks Usage for Underground Water Storage & River Nile In Toshka Area

By

Prof. Dr. Abdalla El-Daoushy
Prof. Dr. Amani Omar
Dr. Samir Nassar

1998

Table of Content

Preface

Egypt in need to increase its agricultural production within a restricted availability of Nile water contained in Naser Lake.

The increment of agricultural production can be achieved by irrigating the land around Nile Valley, delta, and deserts.

During the last two decades, the annual runoff of the River Nile has declined to be below its average (55.5 billion m^3 / year. This brought into attention the necessity of effective usage of the available underground water inventory (4.5 billion m^3 / year) [Diab, 1992; Biswas, 1991] beside the optimum usage of the available water of River Nile.

Although underground water represent about 9% of the total water available in Egypt, only 3% are currently used [Diab, 1992 and Faried, 1988].

Therefore, the development of underground water resources plays an important role for the efficient use of River Nile and for the control of the underground water distribution.

As Nile water (Surface water) in Egypt is limitted by the present quota of the River Nile, and already water shortage occurs during the summer season, the underground water storage basin becomes an important factor in the total water management.

The aquifer can be over-pumped during the summer and replenished by subsurface drainage water during the remaining months. Anyhow, the main source of recharge the underground water storage specially in the Nile Valley and delta is the seepage from the River Nile.

Underground water in the Nile Valley and delta is an important resource for both irrigating the land of the deserts and domestic water supply for Villages.

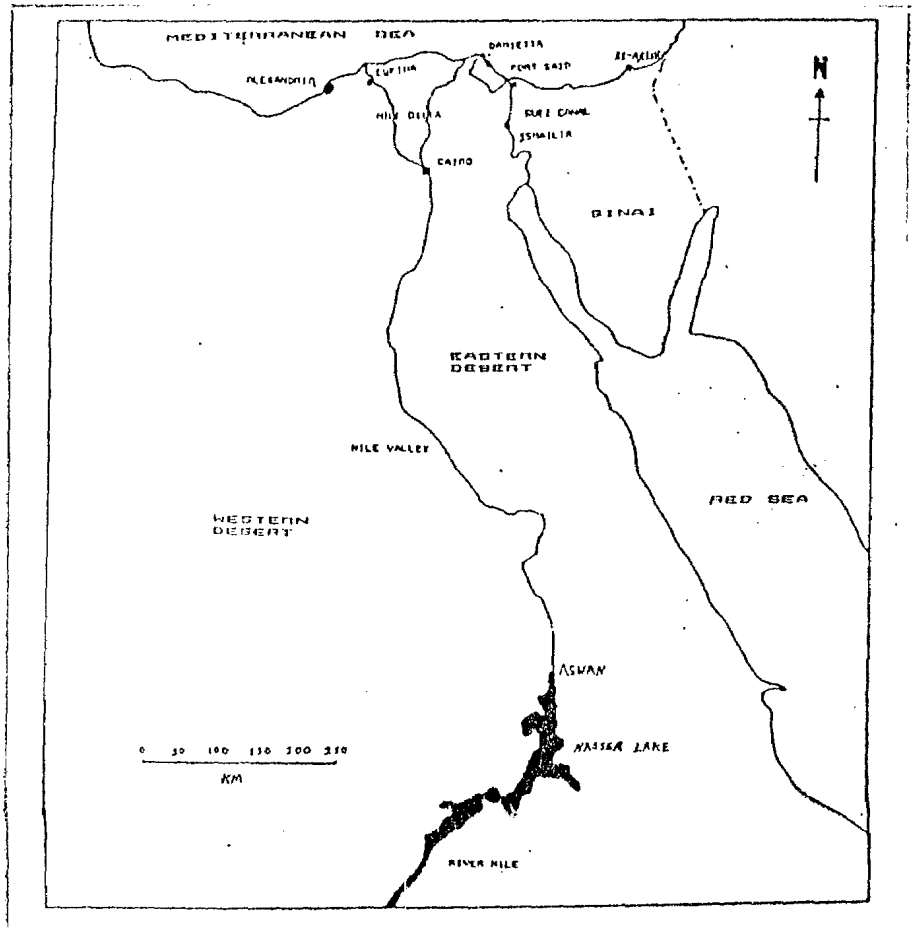
This research project investigates the Neural Networks Approach for solving this underground water problem.

Problem Statement :

Egypt could be divided into 4 major hydrologic provinces separated by natural boundaries [Diab, 1992].

These provinces are :

1. The western desert,
2. The Delta Nile and the Nile valley,
3. The Eastern Desert, and
4. The Sinai Peninsula.



1. The Western desert :

Underground water in the Western Desert, generally founded at great depths. This is not a renewable resource. Preliminary estimates indicate that the total underground water storage in this area is about 40,000 milliard m^3 (Mm^3).

Use of this fossil water depends on the cost of pumping and potential economic return over a fixed time period. Investigations at New Valley indicate that annually about 1.042 Mm^3 of underground water can be used at an economic return rate. This will allow land irrigation of about 150,000 feddans, of which 43,000 feddans are already being cultivated. Therefore, we can cultivate another 107,000 feddans, and this can be increased to about 250,000 feddans if we use the great depth underground water [Biswas, 1991].

2. The Nile Delta and the Nile valley :

The underground water basin below the agricultural lands of the Nile Valley is in fact a large storage reservoir. The total available underground water storage in the Nile valley aquifer is about 200 Mm^3. [The capacity of Naer Lake is about 130 Mm^3, Water Research Center 1990--- Biswas, 1991 ; Attia, 1987]

Another 300 Mm^3 is available in the Delta region [Bioswas, 1991].Thus, the total underground water storage in the Nile valley and Delta is about 500 Mm^3.

The current annual rate of abstracting the underground water for domestic, industrial, and agricultural irrigation was about 2.66 Mm^3 in 1992. This can be increased to 4.9 Mm^3, which is estimated to be equivalent to the annual recharge rate [Diab, 1992; Biswas, 1991].

The main source of recharge the underground water in the Nile valley is the seepage from River Nile.

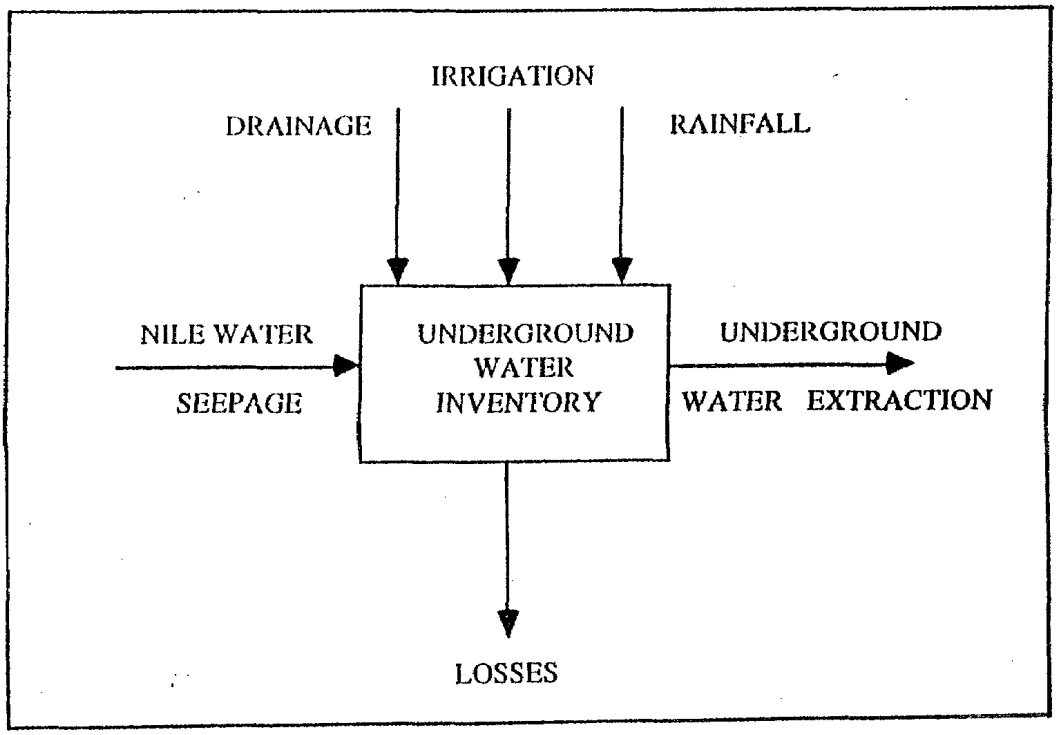
3. The Eastern Desert :

The Eastern Desert is the poorest hydrologic province in Egypt because it is dominated by crystalline igneous and metamorphic rocks. However, there are several wades which can hold amounts of rain and runoff water, as Wadi Qena and Wadi Araba [diab, 1992].

4. The Sinai :

Underground water is available in Sinai in numerous aquifers of varying sizes. More studies are necessary before a better picture emerges of underground water storage and availability in Sinai and Eastern Desert [Allam, 1992; El-Sorady, 1992].

The underground water in the aquifer is generally recharged from the seepage of the surface water, especially from the River Nile, irrigation canals, rainfall, and by down-water percolation of subsurface drainage water in traditionally cultivated lowland, and in reclaimed desert areas.



Research project Objective :

a. To determine the monthly optimum quantity of water released from Naser Lake through the Aswan Barrage and enhancing the productivity of the underground water extraction in the Nile Valley and the Nile Dela.

b. Predicting the released water from Naser lake for the next year and planning a strategy for underground water demand which satisfies the balance between the recharge and discharge of the underground water inventory.

c. Construct a comprehensive system for water distribution in Egypt.

Artificial Neural Networks Approach :

Artificial Intelligence (AI) is the part of computer science concerned with designing intelligent computer systems, i.e., systems that exhibit the characteristics we associate with intelligence in human behavior: understanding-language, learning, reasoning, solving problems, and so on [Luger, 1989].

In AI, the terms problem solving and search refer to a large body of core ideas that deal with deduction, inference, planning, commonsense reasoning, theorem proving, and related processes.

Applications of these general ideas are found in programs for natural language understanding, automatic programming, robotics, expert systems, and mathematical theorem proving.

An expert system is a knowledge-based program that provides "expert quality" solutions to problems in a specific domain. Generally, its knowledge is extracted from human experts in the domain and it attempts to emulate their methodology and performance.

Artificial Neural Networks (ANN) have emerged in recent years as an outcome of a desire to mimic the human brain.

In AI, we begin by reasoning and go down decomposing the reasoning into rules. The ANN approach ,on the contrary, begins with a small building block (the neuron). We then proceed to build a system by interconnecting a large number of neurons such that the output of the system matches the required solution to the problem.

Unlike traditional expert systems where knowledge is made explicit in the form of rules, ANN generate their own rules by learning from being shown examples. Learning is achieved through a learning rule which adapts changes the connection weights of the network in response to the example input and (optionally) the desired output of those input.

There are a number of reasons why ANN are appealing as mechanism for implementing intelligence. Traditional AI programs tend to be brittle and overly sensitive to noise; rather than degrading gracefully. Such programs tend to be either right or fail completely. Human intelligence is much more flexible; we are good at interpreting noisy input, such as recognizing a face in a darkened room from and odd angle or following a single conversation in a noisy party. Even where a human may not be able to solve some problems, we generally make a reasonable guess as to its solution. ANN, because they capture knowledge in a large number of fine-grained units, seem to have more potential for partially matching noisy and incomplete data.

Again, this research project investigate the ANN approach for solving the underground water problem.

Chapter I illustrates an idea about River Nile.

Chapter II describes the mathematical rationalization of water inventory.

Chapter III is devoted to the mathematics of ANN in general.

In Chapter IV, the Backpropagation model of ANN and its application has been discussed.

Chapter V has been devoted to the application of ANN in solving our water distribution problem.

At last, a conclusion is mentioned in Chapter VI.

Finally, due to the shortage of data and information about TOSHKA area, the ANN has been learned and trained on the available data for the Underground water in the Eastern deserts and the available water of River Nile.

When information and data of Toshka are became available, the ANN can be used (without major changes) to get the optimum distribution of the water in Toshka area.

Chapter 1
On the River Nile

Introduction

The underground water in the Egyptian east and west deserts is not a water-resource in itself; it is mainly fed and recharged by the water of the River Nile indirectly. However, the Nile water has been facing problems from time to time as a consequence of the contradiction happening in distributing the River Nile water. For instance, establishing some projects on the Nile flow in the Nile countries may decrease the quantity of the water arriving to Egypt or may delay its arrival at critical time .

Since the Nile water represents the vital artery of Egypt , and since there is a probability that in implementing some foreign projects on the Nile flow would minimize the water quota of Egypt, then it is necessary to rationalize the usage of the surface water of the River Nile and the underground water all over the Egyptian land .

1- Sources of Nile Water and Egyptian Needs of Water (1,6)

Egypt is toiling to increase the present quantity of water so as to face the increasing needs of the water year after year . It is well known that Egypt and eight other countries participate in the Nile valley water, Egypt and Sudan are considered the destination of the River Nile while the others are the source countries from which the Nile is provided by water through rains. These source countries are Ethiopia , Tanzania , Kenya , Burundi, Rwanda and Zaire. The water of the Nile originate from two main sources :

- The equatorial lakes which are Victoria Keoga and Albert lakes . The equatorial lakes provide the river with a continuous flow of water all the year around by an average of 26 .5 million cubic meter ($m.m^3$) yearly . The lakes lose a quantity of water due to evaporation and some quantity is added from Bahr El Gazal so the total quantity of water is about 15 $m.m^3$ before the mouth of the Subat on the white Nile .

- The Abbesenea Mount. The proceed of Abbesenea Mount is seasonal i.e., during the flood season only. It includes the Subat that its annual proceed is 13.5 $m.m^3$ in average, the Blue Nile that its annual proceed is 54 $m.m^3$, and the Atbara that its annual proceed is 12.$m.m^3$ in average. So the total quantity of water proceed from these rivers is 79 . 5 $m.m^3$ in average / year.

The whole proceed of the river water is 94.5 $m.m^3$ average/year and about 10.5 $m.m^3$ are considered as losses, so the net quantity of the water flow at Aswan is 84 $m.m^3$ distributed between Egypt and Sudan, 55.5 $m.m^3$ for Egypt and 18.5 $m.m^3$ for Sudan. The total feasible water proceeds for Egypt amount to 60.7$m.m^3$ yearly. The sources of this amount are as follows:

- 55.5 $m.m^3$ from Nile water.
- 2.9 $m.m^3$ from water table reservoir.
- 2.3 $m.m^3$ from water of draining canals.

On the other hand, the Egyptian needs of water can be classified as follows :

1- For cultivation, 85% of the available water is needed. the amount of water is estimated to be about 49.7 $m.m^3$ /year; this quantity varies and depends upon crop structure and the cultivated area.

2- For drinking water, the quantity of 3.3 $m.m^3$ / year is needed.

3- For industry , the quantity of 2.2 $m.m^3$ / year is needed , and.

4- Unconsumed water, the amount is estimated to nearly 5.$m.m^3$ disposed as river navigation, electricity during December, January and February to run turbines, and water balancing, i.e., water disposed for keeping safety balance, especially after the corrosion has taken place behind the bridge.

The following table illustrates the annual quantity of waters received to Egypt from the River Nile from year 1975 to 1986.

Table (I) . Egyptian Annual Receiving River Nile Water

year	Quantity of water/year	year	Quantity of water m.m3/year
1975	80 . 1 floody	1981	55. 9 unfloody
1976	57 unfloody	1982	41 . 9 unfloody
1977	61. 7 unfloody	1983	47 . 25 unfloody
1978	62. 5 unfloody	1984	33 unfloody
1979	50 .2 unfloody	1985	58 unfloody
1980	78 . 5 unfloody	1986	49 unfloody

Beside the limitation of the present Egypt's quota of the surface Nile water, it should not be ignored that there are several probable engineering projects which can be implemented and affected by the Egyptian quota of Nile water.

Such projects may be erecting dams on the Blue Nile in Ethiopia , or establishing dams on Subat and Atbare rivers , or transferring the water of Tana lake to Place River, or transferring the Nile water to the Nakab desert as Israel ambition etc . So it very essential to rationalize the usage of the available quantity of the underground water beside the surface water.

Chapter 2
Mathematical Rationalization of Water Inventory

Introduction

Egypt needs to raise its agricultural and industrial production in the coming few years. These increases in the production plus the increases of the domestic water consumption will be achieved mainly from the water of the Nile Vally and the Delta. During the last two decades, the annual runoff of the Nile River has declined to below its average of 55.5m.m^3/year.

This brought into attention the necessity of effective usage of available the underground water resources which is estimated to be 4.9 m.m^3 / year. Although these underground water resources represent 9% of the total water available in Egypt, only 3 % is currently in use (see Diab (3)).

As the surface water supplies to Egypt are limited by the present quota of Egypt from the Nile, and since the underground water is not a resource in itself, the underground water storage basin becomes an important factor in the total water management.

The inventory theory and the neural network technique may be utilized to rationalize the usage of the limited amount of the underground water in Egypt. Figure (1) shows that the underground water is generally recharged by the seepage of the surface water of the Nile River, irrigation canals, rainfall, and down water percolation of drainage water. In Egypt, recharge from rainfall is neglected.

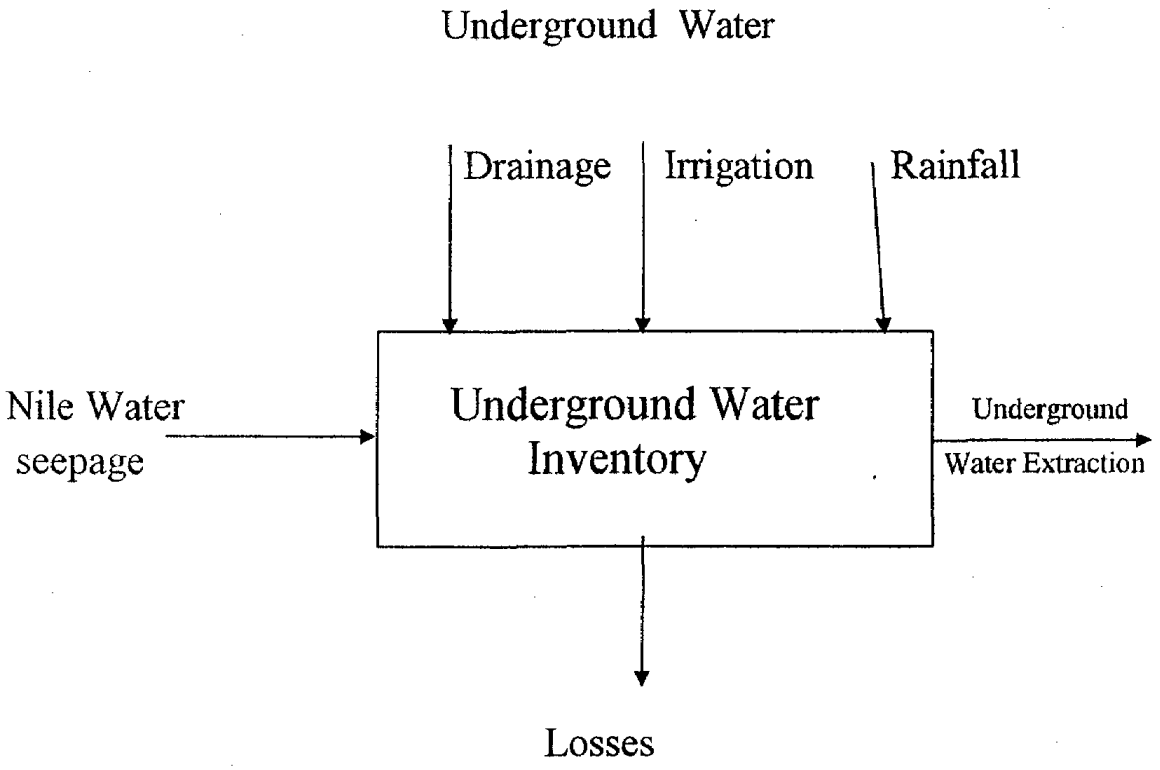


Fig. (1)

The main object of this study is to determine the optimal quantity of the water released from Lake Nasser and to enhance the productivity of the underground water extraction in TOSHKA area. This strategy for the underground water demand satisfies the balance between the recharge and discharge of the underground water inventory.

2. Some Mathematical Models for Inventory-Control Problems (5)

In this section, we will present the inventory theory approach in solving the problem of production schedule, in case of deterministic demand and in case of stochastic demand. Some practical mathematical inventory models in each case will be presented.

An inventory problem exists when it is necessary to stock physical items for the purpose of satisfying demand over a specified time horizon (finite or infinite). Let us consider the plight of the manufacturer of

seasonal items which must determine the monthly production schedule of the items of the next 12 months. The demand for the product is assumed to be fluctuated over the coming year. It is required to meet the monthly requirements as a given sales-forecast and it is important to fulfill the demands either by producing the desired demand during the month or by producing part of the desired amount and making up the difference by using the over production from the previous month, i.e., stock is available. In general, any scheduling problem has many different approaches that will satisfy the unit time-period requirements. For example, the exact number of items required by the sales forecast can be produced at each unit of time. However, a fluctuating production schedule is costly to maintain because of the overtime costs in high-production at a certain time and because of the costs associated with the release of machinery in low production at another period of time.

On the other hand, the fluctuating requirements can be faced by over producing in periods of low requirements, storing the surplus, and using the excess in periods of high requirements. However, because of the cost of keeping items in storage, such an approach may be not efficient if there are comparatively large surpluses at a certain unit of time.

Problems of this nature involve difficulties that arise whenever contradicting objectives are inherent in the models. For this kind of problems, the target is to determine an efficient production schedule which minimizes the sum of the costs due to the output fluctuations and due to inventories.

Efficient scheduling means the determination and acceptance of a compromise solution which lies between the two extreme solutions, the one that minimizes surpluses and the one that minimizes fluctuations. Obviously, the optimum production schedule will depend on the relative costs assigned to the conflicting objectives.

Let us develop some mathematical models for Inventory Scheduling Problems.

2.1 Inventory Models with Deterministic Demand

Practically, it is very difficult to formulate a single general inventory model taking into account all relations and variations in the real-world systems. Therefore, such models are usually developed for specific situations. In this section, we consider the inventory models which involve completely known and fixed demands.

2.1.1 Model (I): Uniform Demand Rate & Infinite Production Rate

Let us assume a manufacture that produces a single commodity and has to provide items at a uniform rate R / unit time, i.e., the demand is fixed and known.

Let:

R = the number of items required / unit time,

C_1 = the cost of holding one item in inventory unit time,

C_2 = shortage cost /item /unit time

C_2 = (α , since no shortage are allowed),

C_3 = the cost of setting up production run

q = the number of items produced / run , (q = Rt)

t = the interval of time between two runs , and

I_m = the number of inventory items at the beginning of time interval t (I_m is a variable) .

Since model deals with infinite rate of production, the replacement must be instantaneous, i.e., lead-time equals zero. The problem is then to determine the optimal number of units to be processed in each production run. Figure (2) illustrates how the inventory system operates with the assumptions imposed in this model.

Inventory with Infinite Rate of Production

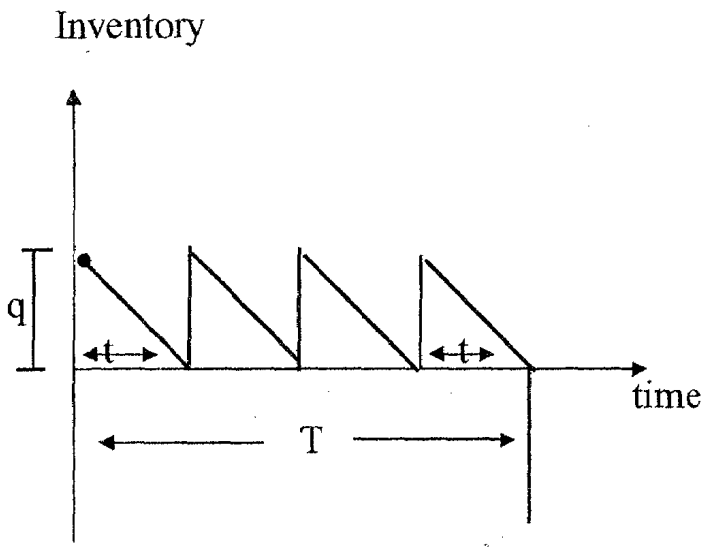


Fig. (2).

Starting at any time with an inventory q, the inventory Will decrease at the rate R until it reaches zero, then an order of q items must be placed immediately to satisfy the demand without delay. If a production run is made at intervals t, then a quantity $q = Rt$ must be produced in each run. The stock in an infinitesimal interval of time dt is Rtdt .

Hence, the stock in time interval t is

$$\int_0^t Rtdt = \frac{1}{2} Rt^2 = \frac{1}{2} qt \quad (\text{Area of inventory triangle})$$

The cost of holding inventory for production run

$= \frac{1}{2} C_1 qt$.

$\therefore$ the average total cost / unit time $= \frac{1}{2} C_1 qt + C_3$, i.e ,

$$C(q) = \frac{1}{2} c_1 q + \frac{C_3 R}{q}$$

Since the object of this model is to minimize the total cost, hence, C(q) will be minimum if

$$\frac{dc(q)}{dq} = o \quad \text{and} \quad \frac{d^2 c(q)}{dq^2} > o \quad \ldots \quad 1$$

Satisfying the two conditions of (1) , ωe deduce that the optimum number of produced items q_o /run is

$$q_o = \sqrt{\frac{2C_3 R}{C_1}} \quad ,$$

and the optimum interval of tim $t_o = \sqrt{\frac{2C_3}{C_1 R}}$, $\ldots 2$

and the minimum average cost C_o / unit time is

$$C_o = \sqrt{2C_1 C_3 R} \quad .$$

Formula (2) is known as the optimal lot size .

In this model, we have assumed that the lead time is zero, however, most practical problems involve a positive lead time from the time of the items-order until the delivery is actually satisfied. Thus if L is the lead time and R is the inventory consumption rate / unit time, then the total inventory requirements during the lead-time = LR. Therefore, an order of q quantity must be placed as soon as the stock level becomes LR, i.e., there is a reorder point p = LR

2.1.2. Model (II): Non-Uniform Demand Rate & Infinite Production Rate

All assumptions of Model (I) are supposed to be satisfied in this model, except that we are given some total demand D to be satisfied during a period of time T, and the demand rates are different for different production runs.

Now if q is the fixed quantity produced / production run, then the number of production runs is

$N = D / q.$

And so $T = t_1 + l_2 + t_n$,

where t_i is the time interval between production run i and $i+1$.

Hence, the holding costs for the time period takes the form

$$T = (\tfrac{1}{2} q t_1)c_1 + (\tfrac{1}{2} q t_2)c_1 + \ldots + (\tfrac{1}{2} q t_n)c_1 = \tfrac{1}{2} qTc_1,$$

and the setup costs $= C_3.N = C_3.D / q$.

Thus the total cost $c(q) = \frac{1}{2} q TC_1 + C_3D/q$.

The minimum cost and the optimal lot size are determined by satisfying the two conditions :

$\frac{d}{dq} c(q) = o$ and $\frac{d^2}{dq^2} c(q) > o$. Hence

The optimal lot size $q_o = \sqrt{\frac{2C_3 (D/T)}{c_1}}$, and

the minimum total cost $c_o (q_o) = \sqrt{2C_1 C_3 (D/T)}$

2.1.3 Model(III) Uniform Demand Rate and Finite Production Rate

For this model it is assumed that a new production run is started when inventory becomes zero, and the run size is constant .

Let K be the number of items produced / unit time.
Figure (3) illustrates the variation of inventory with finite rate of production .

Inventory with Finite Rate of Production

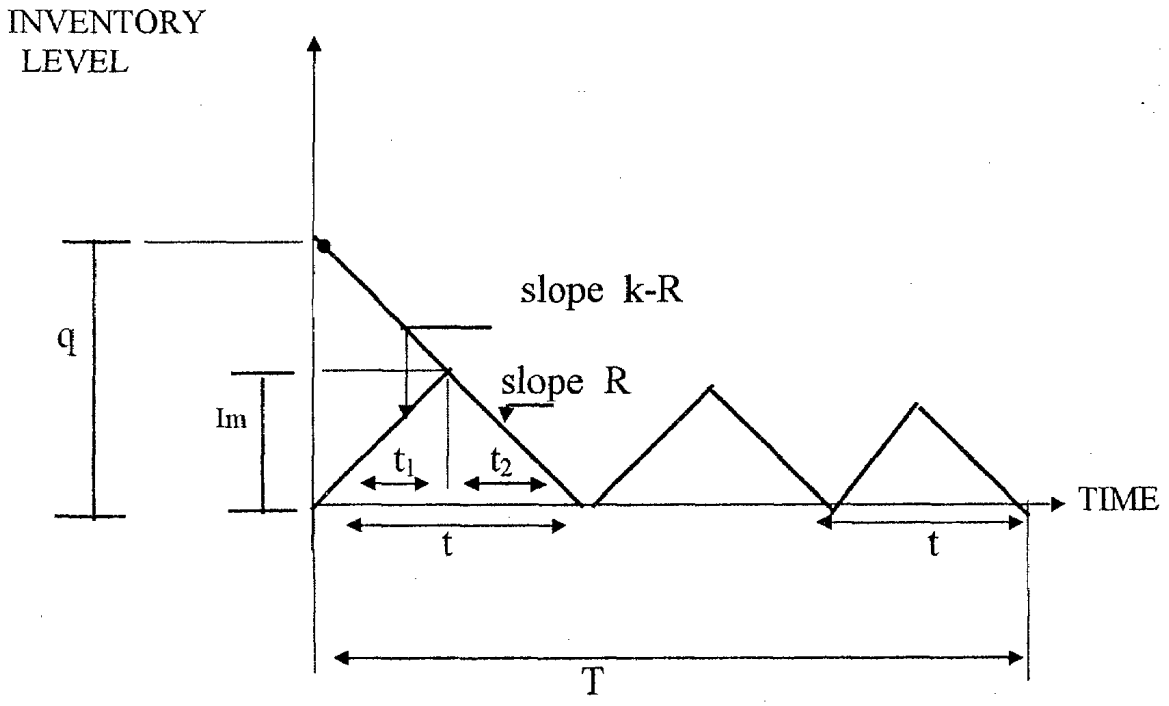


Fig. (3).

The production -run time t for this problem is divided into two periods of time:

- the time period t_1 during which the stock is accumulated up to a constant rate of K-R items / unit time , and

- the time period t_2 during which there is no production and inventory is decreasing at a constant demand rate R / unit time .

If I_m is the inventory available at the end of time t_1 ,
then $I_m = (K - R)\, t_1$.

However, the remaining quantity available at the end of t_1 is

$$I_m = q - Rt_1 ,$$

where Rt_1 represents the required quantity during t_1 .

$$\therefore \quad I_m = \frac{K - R}{K} q \quad .$$

Hence , the holding cost pre-production - run time t is
$\frac{1}{2} I_m\, t\, c_1$ and the setup cost per production run is C_3 .
Thus the average cost /unit time is ,

$$C(I_m, t) = \frac{1}{2} I_m C_1 + C_3 / t = \frac{1}{2}\frac{K - R}{K} C_1 q + \frac{C_3 . R}{q} .$$

Differentiating c(q) with respect to q and equating the result to zero , we get,
the optimum lot size $q_o = \sqrt{\frac{2C_3}{C_1} . \frac{KR}{K-R}}$,and

the optimum time interval $t_o = \sqrt{\frac{q_o}{R} . \frac{2C_3}{C_1 R} . \frac{K}{K-R}}$

and the optimum average cost $C_o = \sqrt{2C_1C_3R . \frac{K-R}{K}}$.

Note : if $K = R$, then $C_o =$ o, i, e, there will be no holding cost and no setup cost .

if $k = \infty$, then this model reduces to model (I) .

2.1.4. Model (IV) Uniform Demand Rate, Infinite Production Rate, and Allowed Shortages

In this model, we assume that the lead-time (*) is zero. The total time period T is divided into n equal time intervals each of length t; further the time interval t is subdivided into two subintervals t_1, &t_2, where t_1 is the time during which the items are being produced and accumulated but not to the q-level, and t_2 is the time during which items are drawn from inventory. Figure (4) illustrates this situation

Inventory Variations for Uniform Demand Rate and shortages

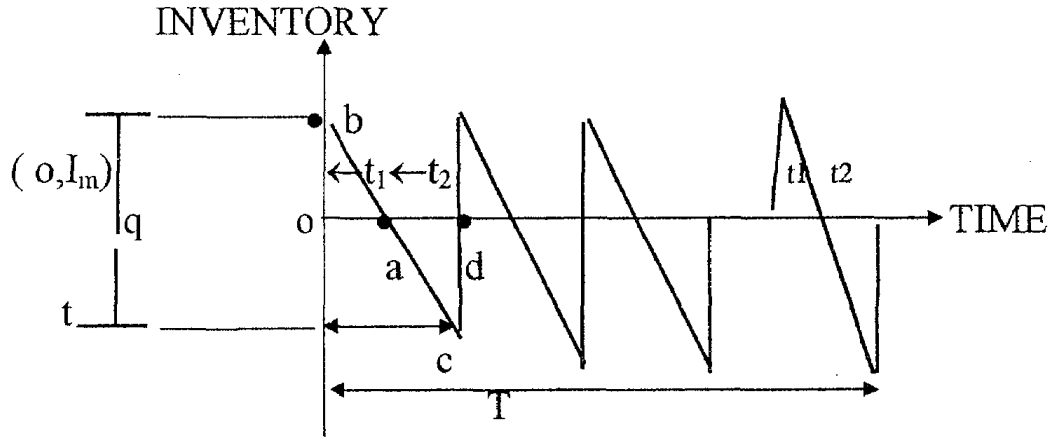


Fig (4)

The total inventory during the time period

$t = \text{area } \Delta o\ ab = \frac{1}{2} I_m . t_1$

---

* **Lead time is the time between placing an order and its receipt in stock. If the lead time is zero, there will be no need to order in advance. If it is not zero by an amount equal to lead time, provided the demand is Known.**

Hence, the inventory holding cost during time $t = \frac{1}{2} C_1 . I_m . t_1$

Also, the total shortage during time t= area Δ acd $= \frac{1}{2}(q-I_m).t_2$

The set up cost during time $t = C_3$.

$\therefore$ The total overage cost / unit time has the form:

$$C(I_m, t) = \frac{1}{t}\left[\frac{1}{2} C_1 I_m t_1 + \frac{1}{2} C_2 (q-I_m) t_2\right] + \frac{C_3}{t}, \text{ or}$$

$$C(I_m, q) = \frac{1}{2} C_1 \frac{I_m^2}{q} + \frac{1}{2} C_2 \frac{(q-I_m)^2}{q} + \frac{C_3 . R}{q} .$$

Equating the partial derivative of C (I_m, q) with respect to I_m with zero , we get

$$\frac{\partial C(I_m,q)}{\partial I_m} = \frac{C_1 . I_m}{q} - \frac{C_2(q-I_m)}{q} = o \text{ , which implies}$$

$$I_m = \frac{C2}{C_1+C_2} . q$$

Since $\frac{\partial^2}{\partial I_m^2} C(I_m, q) > o$,

then the optimal value of I_m , $I_m^o = \frac{C2}{C_1+C_2} . q$.

Similarly $\frac{\partial C(I_m,q)}{\partial q} = o$, implies

$$q = \sqrt{\frac{C_1 + C_2}{C_1 C_2}} . \sqrt{2C_3 R} .$$

Since $\Delta = \begin{vmatrix} \frac{\partial^2 C(I_m,q)}{\partial I_m^2} & \frac{\partial^2 C(I_m,q)}{\partial I_m \partial q} \\ \frac{\partial^2 C(I_m,q)}{\partial q \partial I_m} & \frac{\partial^2 C(I_m,q)}{\partial q^2} \end{vmatrix} > o$, hence ,

the optimal value of q ,

$$q^{o} = \sqrt{\frac{C_1+C_2}{C_2}} \sqrt{\frac{2C_3 R}{C_1}} .$$

Hence , the minimum average cost / unit time is

$$C^{o}(I_m , q) = \sqrt{\frac{C_2}{C_1+C_2}} \cdot \sqrt{2C_1C_3R} \text{ , and}$$

the optimum time interval t^{o} between production runs

is $t^{o} = \frac{q^{o}}{R} = \sqrt{\frac{C_1+C_2}{C_2}} \sqrt{\frac{2C_3}{C_1R}}$.

2.1.5 Model (v) Uniform Demand Rate, Finite Production Rate, and Allowed Shortages.

The time interval t is divided into intervals t_1, t_2, t_3, t_4. In the beginning, the inventory equals to zero, it increases at a constant rate (K-R) in the time interval t_1, until it reaches a level I_m. During the time interval t_2, there will be no replenishment and the inventory decreases at constant rate Run till it becomes zero. Shortage starts piling up at constant rate R during the time interval t_3 until it reaches a level S. Finally, production Starts and shortage decreases at a constant rate K-R during the time interval t_4 till the shortage becomes zero . This completes one cycle of time t, and this cycle repeats itself over and over again. Figure (5) illustrates this scheme .

Inventory Variations with Uniform Demand Rate and Finite Production Rate and Shortages .

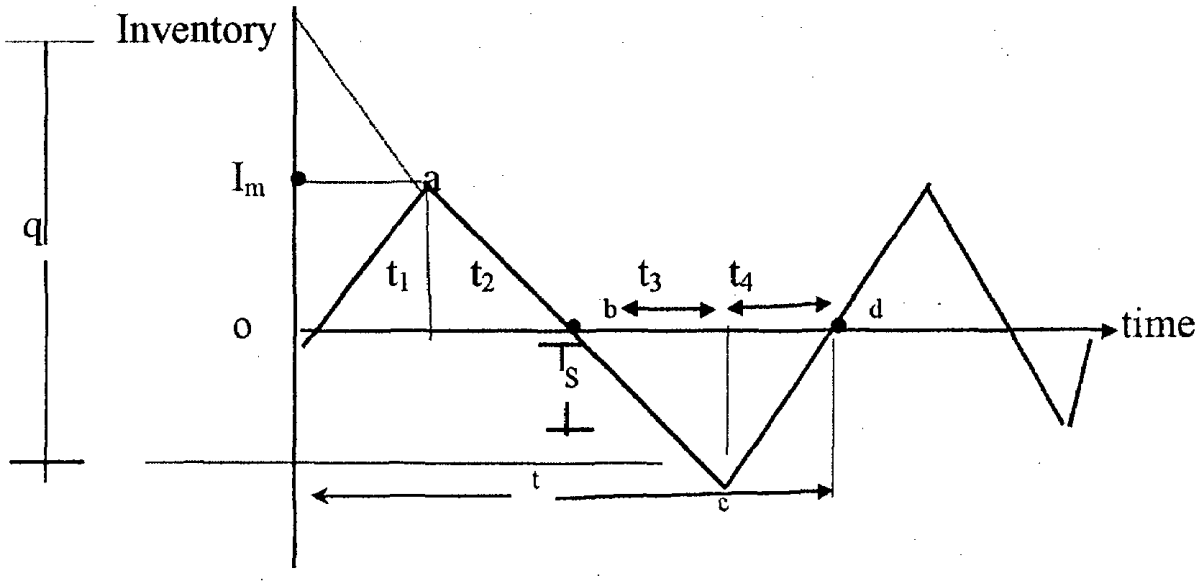


Fig (5)

Now, holding cost during the time interval t $= C_1 . \frac{1}{2} I_m (t_1+t_2)$

The shortage cost during the time interval t $= C_2 . \frac{1}{2} S (t_3+t_4)$

The set up cost $= C_3$

$\therefore$ Total average cost / unit time

$$C = \frac{\frac{1}{2}C_1 . I_m.(t_1+t_2) + \frac{1}{2} C_2.S (t_3 + t_4)+C_3}{t_1+t_2+t_3+t_4}$$

C is a function of the variables I_m ,S,t_1 ,t_2 ,t_3 , and t_4 .

We have :

$I_m = (K-R) t_1$ and $I_m=Rt_2$ implies

$(K-R) t_1=Rt_2$;

furthermore $S= Rt_3$ and $S= (K-R)\, t_4$ implies

$$(K-R)t_4 = Rt_3 .$$

Since $(K-R)\,(t_1+t_4)= R\,(t_2+t_3)$, $I_m+S=R\,(t_2+t_3)$,

and $q = K(t_1+t_4)$, then

$$I_m = q\left(1 - \frac{R}{K}\right) - S .$$

Also

$$t_1+t_2 = \frac{I_m}{K-R} + \frac{I_m}{R} = \{q\,(1 - \frac{R}{K}) - S\}\ \{\frac{1}{K-R} + \frac{1}{R}\} ,$$

and $$t_3+t_4 = \frac{S}{K-R} + \frac{S}{R} = S\{\frac{1}{K-R} + \frac{1}{R}\} . ,$$

and $$t_1+t_2+t_3+t_4 = \frac{q}{R} .$$

Substituting by I_m , t_1+t_2 ,t_3+t_4 , and $t_1+t_2+t_3+t_4$ into the equation of the total cost C , we get

$$C(q,s) = \frac{1}{2q} \cdot \frac{K}{k-R} \cdot \{C_1[q\frac{K-R}{k} - S]^2 + C_2S^2\} + \frac{R}{q}C_3 .$$

The cost function c (q,s) attains its minimum value when

$$\frac{\partial}{\partial q}[C(q,s)] = o\,,\quad \frac{\partial}{\partial s}[C(q,s)] = o \text{ and } \begin{vmatrix} \frac{\partial^2 c}{\partial q^2} & \frac{\partial^2 c}{\partial q \partial s} \\ \frac{\partial^2 c}{\partial s \partial q} & \frac{\partial^2 c}{\partial s^2} \end{vmatrix} > o \quad \partial q$$

Satisfying these conditions, we get the following optimal values :

The optimal backlog level $S^o = q\,\frac{k-R}{k}\,.\,\frac{C_1}{C_1+C_2}$,

The optimal quantity $q^o = \sqrt{\frac{(C_1+C_2)}{C_2}\,.\,\frac{k}{k-R}\,.\,\frac{2C_3R}{C_1}}$,

The minimum cost $C^o = \sqrt{\frac{2C_1C_2}{C_1+C_2}\,.\,C_3\,.\,\frac{R(k-R)}{k}}$,

The optimal time interval $t^o = \frac{q^o}{k} = \sqrt{\frac{(C_1+C_2)}{C_2}\,.\,\frac{k}{k-R}\,.\,\frac{2C_3}{C_1R}}$,

The optimal inventory quantity $I_m^o = \sqrt{\frac{C_2}{C_1+C_2}\,.\,\frac{k-R}{k}\,.\,\frac{2C_3R}{C_1}}$.

Note that if $k = \infty$, then this problem reduces to model (IV) , and to model (I) if $k=\infty \& C = \infty$, and to model (III) if $C_2 = \infty$.

2 .2 Inventory Models with Stochastic Demand [5]

Generally, in practical problems, demand is hardly known precisely, only probability distribution of future demand rather than the exact value is known, i.e., stochastic demand. For problems of stochastic demand, expected costs rather then actual costs are to be minimized. The expected costs are usually obtained by multiplying the actual costs for a

particular situation with the probability of occurrence of that situation and then either summing up or integrating according to either discrete probability distribution or continuous distribution respectively .

In the sequel, we consider four different types of inventory problems with stochastic demand .

let
R = the discrete demand rate with probability P_R ,

I_m = the discrete stock level for time interval t, and

t = constant interval time between orders.

2.2.1 Model (I) : Discrete-Stock Level and Instantaneous Demand Rate

In this model, the production is assumed to be instantaneous, lead time and set up cost be zero.

The problem is to find the optimal level of I_m when $R > I_m$ or $R \leq I_m$, at the beginning of each time interval .

When there exists surplus in stock , i.e , $R < I_m$,
the cost of oversupplying = $C_1 (I_m - R)$.

When the quantity demanded > stock quantity, i.e., R>Im ,
the cost of under-supplying = $C_2 (R - I_m)$.

There fore, the expected cost / unit time is :

$C_1 (I_m - R) . P_R$ if $R < I_m$
$C_2 (R - I_m) . P_R$ if $R < I_m$
and 0 if $R = I_m$

Hence, the total expected cost / unit time takes the form :

$$C(I_m) = C_1 \sum_{R=o}^{I_m} (I_m - R) \cdot P_R + C_2 \sum_{R=I_m+1}^{\infty} (R - (I_m+1)) \cdot P_R .$$

The solution to the problem is the value I_m which minimizes the total expected cost $C(I_m)$.

It is well known that for the probability P_R ,

$$\sum_{R=o}^{Im} P_R + \sum_{R=I_m+1}^{\infty} P_R = \sum_{R=o}^{\infty} P_R = 1 .$$

It is easy to show that

$$C(I_m+1) = [\, C_1 \sum_{R=o}^{Im} (I_m - R) \cdot P_R + C_2 \sum_{R=I_{m+1}}^{\infty} (R - (I_{m+1}) P_R \,]$$
$$+ C_1 \sum_{R=o}^{Im} P_R - C_2 \{ 1 - \sum_{R=o}^{Im} P_R \} .$$

That is $C(I_m + 1) = C(I_m) + (C_1 + C_2) \cdot P_{R \le I_m} - C_2$

Similarly , it has been deduced that

$$C(I_m-1) = C(I_m) - (C_1+C_2) P_{R \le I_m} -1+ C_2$$

Let us consider the quantity I_m such that

$$(C_1+C_2) \cdot P_{R \le I_m} - C_2 > o$$

.....(2 . 2 . 1)

and $-(C_1+C_2) \cdot P_{R \le I_{m-1}} + C_2 > o$

Since $P_{R \le I_m^{\circ}}$ is non-decreasing for increasing I_m° , then for

any integer $\bar{I}_m > I_m^{\circ}$ and any integer $I_m < I_m^{\circ}$, inequalities (2.2.1) hold.

Hence

$$C(\bar{I}_m) > C(I_m^{\circ}) \quad \text{for } \bar{I}_m > I_m^{\circ};$$

and

$$C(I_m^{*}) > C(I_m^{\circ}) \quad \text{for } I_m^{*} > I_m^{\circ}$$

that is I_m° which satisfies the inequality

$$P_{R \le I_{m-1}^{\circ}} < \frac{C2}{C_1 + C_2} < P_{R \le I_m^{\circ}}$$

is the optimal value of I_m which minimizes the expected cost $C(I_m)$.

Therefore, if the oversupply cost C_1 and the shortage cost C_2 are known, the optimal quantity I_m° can be determined when the value of cummulative probability distribution exceeds the ratio $\frac{C2}{C_1 + C_2}$.

Note that if I_m° is such that $P_{R \le I_{m-1}^{\circ}} < \frac{C2}{C_1 + C_2} = P_{R \le I_m^{\circ}}$,

then $C(I_m^{\circ}+1) = C(I_m^{\circ})$, and in this case the optimal

quantity of I_m is I_m° or $I_m^{\circ} + 1$ On the other hand,

if I^{o}_{m} is such that $P_{R \leq I^{o}_{m-1}} = \frac{C2}{C_1+C_2} < P_{R \leq I^{o}_{m}}$, then
$C(I^{o}_{m}-1) = C(I^{o}_{m})$, and in this case the optimal quantity I^{o}_{m} is $I^{o}_{m}-1$ or I^{o}_{m}.

2.2.2 Model (II) Continuous - Stock level and instantaneous Demand Rate

In this model, the stock levels are continuous, rather than discrete. The cost function for this model is similar to the one derived for model (I) with two differences: P_R is replaced by f(R) dR, where f (R) is the probability density function, and the summation is replaced by integration.

Now,
the expected quantity of oversupply $= \int_{o}^{Im} (I_m - R)\, f(R)\, dR$,
and

the expected quantity of under-supply $= \int_{Im}^{\infty} (R - I_m)\, f(R)\, dR$,
Then the total expected cost is

$$C(I_m) = C_1 \int_{o}^{Im} (Im - R)\, f(R)\, dR + C_2 \int_{Im}^{\infty} (R - I_m)\, f(R)\, dR.$$

The cost $C(I_m)$ is minimum if

$$\frac{d}{dI_m} C(I_m) = o \text{ and } \frac{d^2}{dI_m^2}(C(I_m)) > o.$$

Applying the rule :

$$\frac{d}{dI_m}\left[\int_{a(x)}^{b(x)} (R,x)\, d\,x\right] = \int_{a(x)}^{b(x)} \frac{\partial}{\partial x} f(R, x_m)dx + \left[f(R,x)\frac{dR}{dx} \right]_{a(x)}^{b(x)},$$

to the cost function $C(I_m)$, we get

$$\frac{d}{dI_m}[C(I_m)] = C_1 \int_{R=o}^{Im} (1-o)\, f(R)dR + C_1 \left[(I_m-R)f(R)\frac{dR}{dI_m}\right]_{R=o}^{Im}$$

$$+ C_2 \int_{Im}^{\infty} (o-1)\, f(R)dR + C_2 \left[(R-I_m)f(R)\frac{dR}{dI_m}\right]_{R=Im}^{\infty}.$$

The last equation reduces to

$$\frac{d}{dI_m}(C(I_m)) = (C_1 + C_2)\int_{R=o}^{Im} f(R)\, dR - C_2$$

$$\frac{d}{dI_m}(C(I_m)) = o \text{ implies that } \int_{R=o}^{Im} f(R)dR = \frac{C_2}{C_1+C_2}.$$

Since C_1 and C_2 are positive constants and $f(I_m) > o$, then

$$\frac{d^2}{dI_m^2}[C(I_m)] > o .$$

Hence, the optimal value I_m^o is the one that satisfies

$$\int_o^{I_m^o} f(R)\, dR = \frac{C_2}{C_1+C_2}.$$

2.2.3 Model (III) Discrete - Stock level and Continuous – Demand Rate

The reorder time is assumed to be fixed and known, so the set up cost equals zero; furthermore, the production is assumed to be instantaneous and so the lead time is negligibly small. There are two cases in this problem:

- the case when $R \leq I_m$, i.e , there are no shortages, in this case, the only cost considered is the holding cost

which is $= C_1 (I_m - \frac{R}{2}) t$

- the case when $R > I_m$, i.e. , the Shortage Cost is involved.

Figure (6) illustrates the inventory situation for the case $R > I_m$.

The cost involved in this case $= C_1 \frac{I_m^2 t}{2R} + C_2 \frac{(R-I_m)^2 t}{2R}$

Hence the total expected cost / time is

$$C(I_m) = \sum_{R=o}^{Im} C_1 (I_m - \frac{R}{2}) . P_R + \sum_{R=I_m+1}^{\infty} (C_1 \frac{I_m^2 t}{2R} + C_2 \frac{(R-I_m)^2}{2R}) . P_R .$$

Inventory Variation When $R > I_m$.

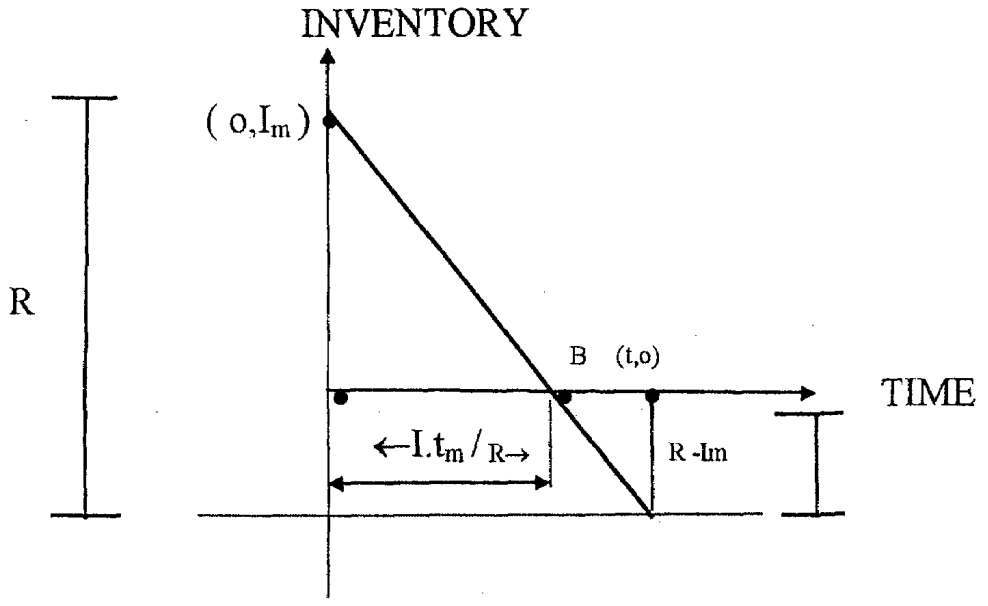


Fig . (6)

The total expected cost / unit time will be minimum if the inventory level I_m satisfies the relation :

$$\Delta C\,(I_m\text{-}1) < o < \Delta\ C\ \ (I_m)\,,$$

where $\quad \Delta C\,(I_m) = C(I_m + 1) - C(I_m)\,,$

But

$$\Delta C(I_m) = (C_1+C_2) \sum_{R=o}^{Im} P_R + (C_1+C_2) \sum_{R=Im+1}^{\infty} \frac{2Im+1}{2.R} .P_R\text{-}C_2.$$

$$\Delta C(I_m) > \text{ implies } \frac{C_2}{C_1+C_2} < \sum_{R=o}^{Im} P_R + (I_m + \frac{1}{2}) \sum_{R=I_m+1}^{\infty} \frac{PR}{R} .$$

Similarly , $\Delta C(I_m-1) < o$ implies that

$$\frac{C_2}{C_1+C_2} > \sum_{R=o}^{Im-1} P_R + \left(I_m - \frac{1}{2}\right) \sum_{R=Im}^{\infty} \frac{PR}{R}$$

Hence the relation

$$\sum_{R=o}^{Im-1} P_R + \left(I_m - \frac{1}{2}\right) \sum_{R=Im}^{\infty} \frac{PR}{R} < \frac{C_2}{C_1+C_2} < \sum_{R=o}^{Im} P_R + \left(I_m + \frac{1}{2}\right) \sum_{R=Im+1}^{\infty} \frac{PR}{R}$$

gives the range of optimum value of I_m ,i.e, the stock levels which minimize the total expected cost .

2.2.4 Model (IV) Continuous -Demand Rate and Continuous Stock Levels

The expected Cost function for this model takes the form :

$$C(I_m) = \int_o^{Im} C_1\left(I_m - \frac{R}{2}\right) f(R)\, dR + \int_{Im}^{\infty} \left\{\frac{C_1 I_m^2}{2R} + \frac{C_2 (R-I_m)^2}{2R}\right\} f(R) dR.$$

Differentiating $C(I_m)$ with respect to I_m , we get

$$\frac{d}{dI_m}(C(I_m)) = (C_1+C_2) \int_o^{Im} f(R)\, dR + (C_1+C_2) \int_{Im}^{\infty} I_m \frac{f(R)}{R} dR - C_2$$

The cost will be minimum if $\frac{d}{dI_m}(C(I_m)) = o$, thus ,

$$\int_o^{Im} f(R)\, dR + \int_{Im}^{\infty} I_m \frac{f(R)}{R} dR = \frac{C_2}{C_1+C_2} \qquad \ldots\ldots (2.2.2)$$

Since $\frac{d^2}{dI_m^2}(C(I_m)) = (C_1 + C_2) \int_{Im}^{\infty} \frac{f(R)}{R} dR > o$,

Hence , equation (2. 2 .2) determines the optimum value of I_m which minimizes the total expected cost / unit time .

Chapter 3
Artificial Neural Networks Application
The Case of Water Distribution Problem in Toshka Area

Introduction

Artificial Neural Networks have emerged in recent years as a desire to mimic the human brain thinking. Neural networks are human attempts to simulate and understand what goes on in nervous systems, with the hope of capturing some of the power of the biological systems. Generally, the artificial neural networks approach as a technique for solving real-world problems begins with a small building node, then a process is carried out to build a system by interacting a certain number of nodes such that the output of the built system achieves a required solution to the problem to a certain degree of accuracy.

In the 1950's and 1960's, a group of researchers produced the first artificial neural networks. Initially, these neural networks were implemented as electronic circuits, then later converted to the more flexible models of computer simulation. Mervim Minsky, Frank Rosenblatt, Bernard Widrow, and others developed neural networks consisting of a single layer of neurons which were applied to diverse problems such as weather forecast, artificial vision, operations research problems, etc. Discoveries, by several researchers independently in the 1980's, and the widespread of an effective general method of training multi-layers neural networks played a major role in the dissemination of neural networks as a tool for solving wide variety of complex problems*.

Scientists do their best to answer the question: - Why do electronic computers and humans differ so significantly in the way they execute the tasks? The answer has been found in the biological nervous system. A computer usually consists of one processor operating alone, executing instructions one at a time, which were written by a programmer. On the contrary, the human-processor is the nervous system which consists of billions of brain cells (neurons) interconnected to each other and carries

---

For a review of artificial neural networks, see Fausett (4) .*

out millions of processes simultaneously. In this direction, scientists have made many attempts to simulate the functions of the human brain by developing sophisticated neural networks, expert systems and fuzzy logic which collectively make up the field of artificial intelligence.

Neural networks have applied to hundreds of different applications. Among them are:

- Robotics.
- Medical diagnosis.
- Targeted marketing.
- Speech synthesis.
- Pattern recognition on an assembly line.
- Process control.
- Financial prediction.
- Noise filtering.
- Linear programming problems.
- Classification.
- Multi-objective programming problems, and many others.

3.1 Neural Networks: Structures and Classifications

A neural network is built of a certain number of elements called neurons, nodes, or in computer terminology: Processing Elements (PE ^{s}). These nodes are usually arranged in layers and each node is connected to many nodes in other layers by means of direct communication links or arcs. Sometimes nodes are connected to nodes in their own layer, or even to themselves. Each direct link is associated with some weights and each node processes the net input it receives via the connected links by an activation function and provides an output value to other nodes via its outgoing connections. Deciding how the nodes in a network are connected, how the nodes process their input and how the connection weights are modified creates a neural network. Now a day, there are so many different types of neural networks, and are being added every day. Figure (7) illustrates a typical neural net work.

A Typical Neural Network, Showing Layers and Neurons

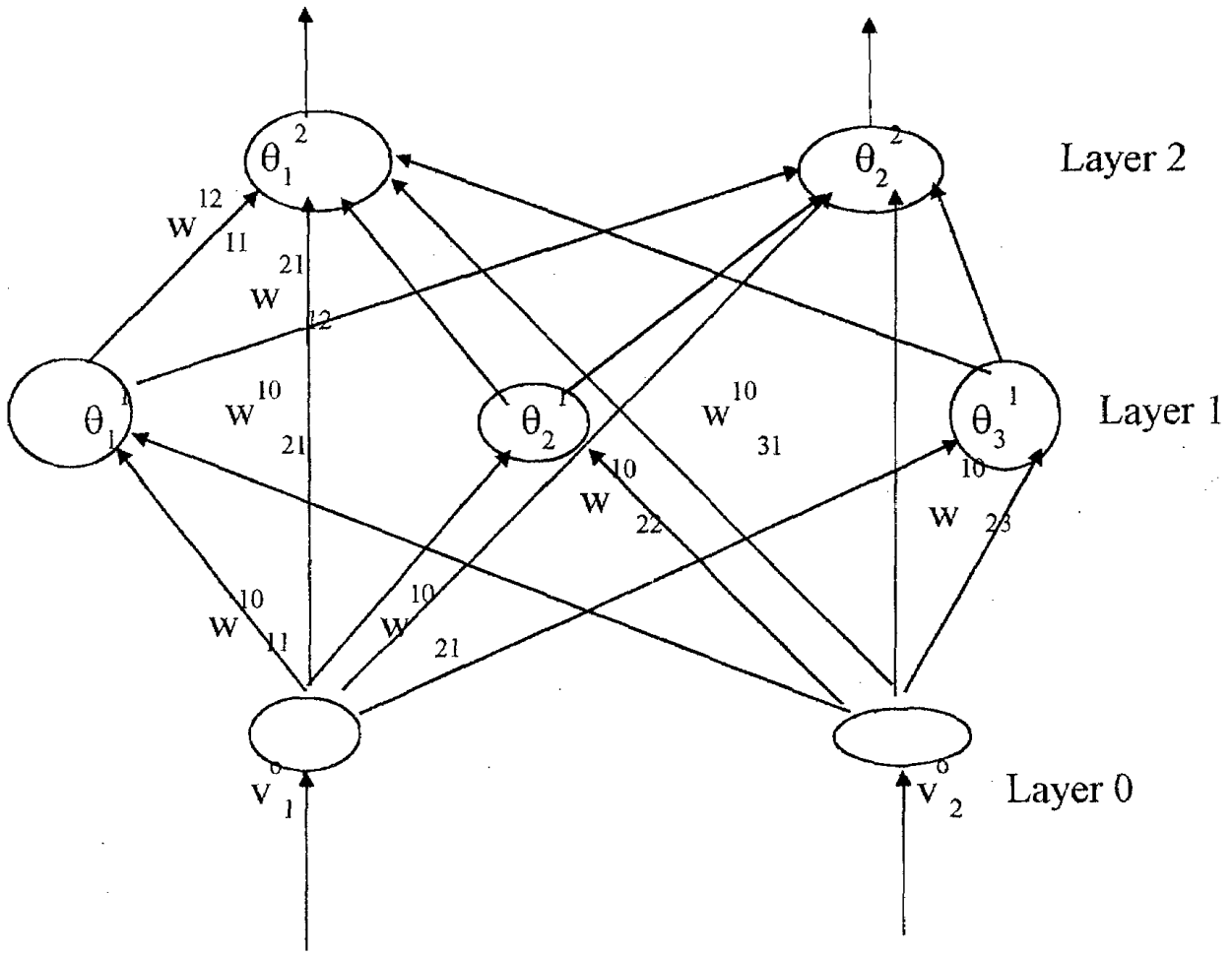


Fig. (7)
A neural network is usually characterized by:

- Its pattern of connections between the nodes, i.e., its architecture, or structure,
- Its activation functions, and
- Its method of determining the weights on connections, i.e., the learning/ training algorithm.

Training the neural network is achieved through learning rules, which adapt the connection weights in response to the connected inputs and the desired outputs.

There are two characteristics that help to divide neural networks into a few basic categories:

- whether the data flows through a network in the forward direction only, or in both forward and backward directions,

- Whether the network is given the correct required solution during training, or the network figures this output by itself.

Networks in the forward direction are called feed-forward networks. Networks with connections that allow data to flow forward and backward are called feedback networks. Back-propagation is an example of a feed-forward network, and a recurrent network is an example of a feedback network type.

In this work, we will be using a back propagation network as a new procedure for solving the problem of optimal utilization and distribution of underground water in TOSHKA ZONE.

The second major characteristic used to classify a network type is how the network is trained. The most common technique for training a network is to feed the network with the input and desired output or result. Some learning rules and mathematical functions are applied to the network producing a current output to be compared with the desired output. The difference between these two values is called the error. The error is then used to adjust the weights of the weighted arcs so that the network will produce a result a bit closer to the desired output in the next iteration, i.e., the network is being trained. This type of learning or training is called supervised; back-propagation network uses the supervised learning technique; an example of supervised learning is the Hebbian learning.

3.2 A Feed-Forward Neural Network Training Back-Propagation Algorithm

Let us assume that we have a feed-forward Neural Network (FFNN) where the nodes are organized into layers, and the weighted arcs only link nodes in lower layers to nodes in higher layers. Nodes in the input layer, called input nodes, accept input from outside world and nodes in the output layer, called output nodes, generate output to the outside world. Nodes in the input layer are used to distribute inputs only and do not serve any processing or computational function. Nodes in layers between the input layer and the output layer are called hidden nodes, and these layers are called hidden layers.

Let the input layer be called layer 0 and let the number of layers be m. If two nodes are not connected, the connectivity weight between them is o. Associated with each υ_k^i is a node bias, also called threshold, θ_k^i.

Let υ_k^i denote node K in layer i ,

η_i be the number of nodes in layer i ,

ω_{kr}^{ij} be the connectivity weights from υ_r^j to υ_k^i,

$w^{kr} = \{\omega_{kr}^{ij}, \theta_k^i\}$ denote the set of weights and node biases.

A (FFNN) can be considered as FFNN: $R^{no} \rightarrow R^{n_m}$,i.e., FFNN is a map mapping vectors from input space R^{no} to the output space R^{n_m}. FFNN is a dynamic process in which the inputs and outputs of the nodes are updated sequentially from the input layer to output layer. In each iteration, the computation process composes of two parts, the summation computations and the transfer computations.

For the summation computations, the input to υ_k^i , denoted by y_k^i is computed as the weighted sum of the outputs of all nodes connected to it from all other lower layers plus θ_k^i, i.e .

$$y_k^i = \sum_{j=o}^{i-1} \sum_{r=1}^{nj} \omega_{kr}^{ij} \times_r^j + \theta_k^i \quad , \qquad i > o \quad .... \quad (3.2.1)$$

where $\times_r^j$ is the output of υ_r^j.

For the transfer computations, the output node is computed, based upon its input, by its activation function. The most frequently used activation function is the sigmoid function defined as *.

$$x_r^j = \frac{1}{1 + e^{y_r^j / T}} \quad , \quad .....(3.2.2.)$$

where T is a user-selected scalar determines the stepness of the activation function.

The purpose in training a FFNN is to determine the values of the elements in W so the FFNN can closely represent the desired outputs. The training of a FFNN is accomplished by:

- Mapping input vectors from the training set by the current topology of the FFNN to their computed output vectors,
- Comparing the computed output vectors with their respective desired output vectors in the training set, and then ,
- Adjusting the values of the components of W so as to reduce any differences between the computed and desired out put vectors .

After a number of iterations, the connectivity weights and node biases of the FFNN will converge to a set of values that minimize the differences between the computed and desired output vectors, and the FFNN will organize itself internally, constructing a model to represent the unknown mapping from the input space to the output space. Thus any new input vector presented to an appropriately trained FFNN will produce an output vector similar to the one that would have been given by the actual mapping .

An algorithm for training FFNNs with multiple layers, which can be used in the optimal distribution of underground water, is presented .

---

* **Other forms for the activation function may be the linear function ,the step function, the tanh function, etc.**

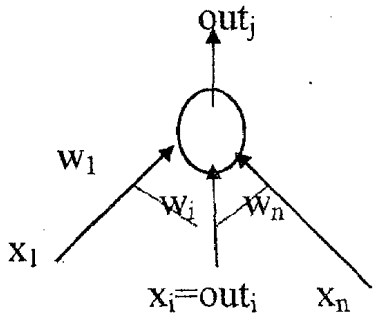
The algorithm is based on the error of the back-propagation technique. Generally, the learning rule of this technique is based on the current weights, values of input, the actual output, and the desired output, and the following is its formula :

$\omega_{ij}(n+1) = \omega_{ij}(n) + \Delta\ \omega_{ij}$, where

$\Delta\ \omega_{ij} = \eta\ \delta_j\, out_i$. $\delta_j = \frac{}{}\ (target_j - out_j)$,

$\delta_j = \frac{}{} = \frac{dx}{dy} = \frac{1}{T} \times (1-x)$, $x = \frac{1}{1+e^{-y/T}}$ and η is a constant called the learning coefficient.

The main aim of training a FFNN is to iteratively adjust the weights in order to minimize the differences ($Target_j$-out_j), for all $_j$, with some permissible error.



Generally the training process proceeds as follows :

- It starts with some initial weights ω^h , h = 1 .
- The FFNN is fed with an input vector representing the input values, and an input vector t^h representing the desired output values.
- The input is processed forward through the network layers till a computed output vector y^h is created.
- $\Delta = y^h - t^h$ is computed.
- The weights are updated according to the rule
- $\omega^{h+1} = \omega_h + \Delta_h$, where
  h is the number of iteration .

- The previous two steps are repeated until the desired output vector, with certain accuracy, is obtained.

The following back-propagation algorithm uses a combination of the Golden section Method and a "doubling and halving " line search strategy, and the Polak and Ribiere conjugate gradient direction . (See Wasserman (8) Lahedi (9)) .

Let $\times_q \varepsilon R^{no}$ be the q^{th} input vector and $t_q \varepsilon R^{nm}$ be the associated desired output vector in the training set.
The pair $(\times_q , t_q) \varepsilon R^{no+nm}$ is called a training pattern. Let the number of patterns in the training set be denoted by Q .

When $\times_q$ is fed the network, the FFNN maps it to an output vector y_q based on 3.2.1.& 3.2.2. We define the error measure E_q for the q^{th} training pattern as

$$E_q = \frac{1}{2} \sum_{j=1}^{n_m} (t_{qj} - y_{qj})^2 .$$

The overall error measure over all Q is given by :

$$E(\omega) = \sum_{q=1}^{Q} E_q(\omega) = \frac{1}{2} \sum_{q=1}^{Q} \sum_{j=1}^{n_m} (t_{qj} - y_{qj})^2$$

Now, the steps of the training back-propagation algorithm can be as follows:

Step o. Initialize the connectivity weights ω_i to small values.
Let $\varepsilon_1 > o$ and $\varepsilon_2 > o$ be small real values. Let $\alpha_o > o$ be a pre - determined constant. Set the iteration number h=1

Step 1. Compute G_h, the gradient of $E(\omega)$ with respect to as follows:

$$\frac{\partial E_q(\omega)}{\partial \omega_{kr}^{ij}} = - \delta_{qk}^{i} \; y_{qr}^{j} \qquad o<i<m ,$$

where y_{qr}^{j} is the computed output of node υ_r^j and δ_{qk}^{i} is the error signal of node υ_k^i and is computed recursively in terms of the error signals of all nodes to which it directly connects, i.e.,

$$\delta_{qk}^{i} = \frac{1}{T} \times_{qk}^{i} (1 - \times_{qk}^{i}) \sum_{s=i+1}^{m} \sum_{t=1}^{n_s} \delta_{qt}^{s} \; \omega_{tk}^{si} .$$

$$G_h = \nabla E(\omega) = \{ g_{kr}^{ij} \}, i=1,2,\ldots,m, j = o, \ldots, m-1, K = 1,\ldots,n$$

and $r = 1,\ldots, n_j$;

$$g_{kr}^{ij} = \frac{\partial E(\omega)}{\partial \omega_{kr}^{ij}} = \sum_{q=1}^{Q} \frac{\partial Eq(\omega)}{\partial \omega_{kr}^{ij}} = - \sum_{q=1}^{Q} \delta_{qk}^{i} \times_{qr}^{j}$$

Let the search direction $O_h = - G_h$

Step 2. Update the connectivity weights by the rule

$$\omega_{h+1} = \omega_h + \eta^* D_h,$$

where η^* is the learning rate .

Let η^* be the value corresponding to the minimum of $E(\omega_h + \eta D_h)$ (*)

Step 3. If $E(\omega_h) - E(\omega_{h+1}) < \varepsilon_1$, then stop .

Step 4. Let $h=h+1$. If $(h \bmod |\omega| 1) = o$, go to step1 , otherwise , ($|\omega|$ is the Cardinality of ω) go to step 5 .

Step 5. Compute G_h . If the norm $||G_h|| < \varepsilon_2$, then stop otherwise, compute

$$\alpha_h = \frac{(Gh - Gh\text{-}1) . Gh\text{-}1}{|| G_{h-1} ||^2}$$

if $\alpha_h > \alpha_o$, then set $\alpha_h = \alpha_o$ and compute a new search direction

$$D_h = - G_h + \alpha_h D_{h-1} .$$

(*) **note that for a given training set and w_h and a given search direction D_h ,**

E becomes a function of η

Chapter 4.

Back-Propagation Neural Network with Momentum

Introduction

In backpropagation with momentum, the weight change is in a direction that is a combination of the current gradient and the previous gradient. This is a modification of gradient descent whose advantages arise chiefly when some training data are very different from the majority of the data (and possibly even incorrect). It is desirable to use a small learning rate to avoid a major disruption of the direction of learning. However, it is preferable to maintain training at a fairly rapid pace as long as the training data are relative similar (*Wasserman, 1989*).

Convergence is sometimes faster if a momentum term is added to the weight to update the formulas on the net. In order to use momentum, the weight (or weight updates) of different trials must be saved. For example, in the simplest form of backpropagation with momentum, the new weights for training step t+1 are based on the weights at training steps t and t-1. The weight update formulas for backpropagation with momentum are

$$w_{jk}(t+1) = w_{jk}(t) + \alpha\delta_k z_j + \mu\left[w_{jk}(t) - w_{jk}(t-1)\right],$$

or

$$\Delta w_{jk}(t+1) = \alpha\delta_k z_j + \mu\Delta w_{jk}(t)$$

and

$$v_{ij}(t+1) = v_{ij}(t) + \alpha\delta_j x_i + \mu\left[v_{ij}(t) - v_{ij}(t-1)\right],$$

or

$$\Delta v_{ij}(t+1) = \alpha\delta_j x_i + \mu\Delta v_{ij}(t),$$

where the momentum parameter μ is constrained to be in the range from 0 to 1, exclusive of the end points.

Momentum allows the network to make reasonably large weight adjustments as long as the corrections are in the same general direction for several patterns, using a smaller learning rate to provent a large response to the error from any one training pattern. It also reduces the likelihood that the net will find weights that are a local, but not global, minimum. When using momentum, the network is proceeding not in the direction of the gradient, but in the direction of a combination of the current gradient and the previous direction of weight correction.

Choices

Random Initialization. The choice of initial weights will influence whether the network reaches a global (or only a local) minimum of the error and, if so, how quickly it converges. The update of the weight between two units depends on both the derivative of the upper unit's activation function and the activation of the lower unit. For this reason, it is important to avoid choices of initial weights that would make it likely that either activations or derivatives of activations are zero. The values for the initial weights must not be too large, or the initial input signals to each hidden or output unit will be likely to fall in the region where the derivative of the sigmoid function has a very small value (the so- called saturation region). On the other hand, if the initial weights are too small, the network

input to a hidden or output unit will be close to zero, which also causes extremely slow learning.

Our procedure, as shown in Chapter 3, is to initialize the weights tc random values between -i and i (for i=0.1) and increase i by 0.1 in each step until the training process reaches a suitable interval.

How long train the network. Since the usual motivation for applying a backpropagation network is to achieve a balance between correct responses to training patterns and good responses to new input pattern (i.e., a balance between memorization and generalization), it is not necessarily advantageous to continue training until the total squared error actually reaches a minimum. The available data is normally broken down into two sets of data: a set of training patterns and a set of testing patterns. These two sets are disjoint. Weight adjustments are based on the training patterns; however, at intervals during training, the error is computed using the testing patterns. As long as the error for the testing patterns decreases, training continues. When the error begins to increase, the network is starting to memorize the training patterns too specifically (and starting to lose its ability to generalize). At this point, training is terminated.

4.1. Min Max Tables and Network Ranges

The raw data is typically in the original units of measurement and may have values that are too large or small, or are not centered. For example, in training a backpropagation network, desired range of values for input values might be

Input minimum value = -1,

Input maximum value = 1.

Choosing such values may help keep the network nodes from saturating and being unable to learn. If we present a backpropagation network with input value such as 55.5, as seen later in our problem in the next chapter, then even with small weights in the network, the summations will be huge and the activation functions will become saturated. When saturated, the derivative of the sigmoid (or hyperbolic tangent) is close to zero at large (either positive or negative) summation values. Since the derivative is a multiplier in the weight update equation, learning stops for processing elements with such large summation values.

NWorks addresses this problem with an easy to use pre-processing facility. The pre-processing facility computes the lows and highs of each data field for all the input data files to be used with a given network. These lows and highs are stored in a table called a Min Max table. The user then decides what ranges the input and output should be scaled to the presentation to the network. NWorks then computes the proper scale and offset for each data field. Real world values are then scaled to network ranges for presentation to the network. After the network has produced a network scaled result, the result is de-scaled to real world units.

Table 4. 1

File layout	Start Field	Network	Ranges
Input	1	-1	1
Output	14	-0.8	0.8

When we select the Min Max table check box and the hyperbolic tangent transfer function, as seen later in Chapter 5, NWorks automatically created a Min Max table from the data files and set the network input and output ranges as shown in Table 4.1. In case of a sigmoid network the settings are:

Input Network Ranges 0 to 1

Output Network Ranges 0.2 to 0.8

The choice of Network Ranges is closely linked to the number of inputs, the type of transfer function in use, and the initial weight values in the network.

The basic idea is to choose a range that produces summations, which will not initially saturate the transfer function.

4.2. Learning Schedule

One way in which researchers have attempted to improve the speed of training for backpropagation is by changing the learning rate during training. This is handled by the use of learning schedule. Only one column from the schedule is used at a time. All others are ignored. The row at top of the learning schedule is labeled "Learn Count", as shown in table 4.2. The values in this row should increase from column. The column following

the last one should have a value of zero. The learn counter is compared to the "Learn Count" at the top of each column. The column chosen is the first one in which the learn count is greater than or equal to the value in the learn counter. In other words, the "Learn Count" is the high inclusive boundary for a column.

Table 4.2

Column	1	2	3	4	5
Learn Count	10000	30000	70000	150000	310000
Learning Rate	0.5	0.25	0.25	0.00391	0.00002
Momentum	0.4	0.2	0.05	0.00313	0.00001
Tolerant Error	0	0	0	0	0

In the following table (Table 4.3) we show that which column would be chosen for different values in the learn counter.

Table 4.3

Column	1	2	3	4	5
Learn Count	50	100	200	300	0
Learn Counter	Column	Reason for choice			
200	3	$100 < 200 \leq 200$			
210	4	$200 < 210 \leq 300$			
350	4	Past the end of the table			

The parameters in the selected column are the ones used by the processing element

During learning, the counter is incremented one for each training cycle and holds its value unless the network is reinitialized; in which case it is reset to 1. Each time a new training example is presented to the network, the learn counter is incremented.

The learning rates can be made to change dynamically as the learning process proceeds.

The primary purpose of the learning schedule is to control the learning parameters as learning proceeds.

The parameters in this schedule can be adjusted to allow one layer to reach equilibrium (or for its learning to stabilize) before another layer begins learning.

This is accomplished by using a different learning schedule for each layer, and setting the learning rates in each column appropriately.

Learn Temperature

This allows noise to be introduced during the learning process. In many instances, it is useful to introduce noise into the training set to help the network avoid local Minami. As the training proceeds, the amount of noise is reduced to zero (NeuralWar, 1991).

For example, uniform function adds a random number within a specified to each unit summation value in the layer. The range for random to each unit summation value in the layer. The range for random plus or minus one percent of the temperature value, as shown in table 4.2. The

random number for the "noise" value is different for each unit in the layer. Tolerant error value to be considered as zero (when this value is reached stops).

Chapter 5

Application of Neural Networks for Solving Water Distribution Problems

5.1. Designing Neural Network Architectures

This section discusses the difficult problem of deciding on the structure and training of a backpropagation network.

It is assumed that the problem to be solved indicates that a multi-layer backpropagation network is the most appropriate model. Given that assumption, we must design the network. While there are no hard and fast rules for defining network parameters, we can generally be safe if we follow three guidelines (*Taha et al., 1995; Masters, 1993*):

(1) Use one hidden layer.

(2) Use very few hidden neurons.

(3) Train until you can't stand it any more.

Choosing an appropriate number of hidden neurons is extremely important. Using too few will starve the network of the resources which it needs to solve the problem.

Using too many will increase the training time, perhaps so much that it becomes impossible to train it adequately in a reasonable period of time. Also, an excessive number of hidden neurons may cause a problem called over fitting. The network will have so mach information processing capability that will learn insignificant aspects of the training set, aspects that are irrelevant to the general population. If the performance of the network is evaluated with the training set, it will be excellent.

The best approach to finding the optimal number of hidden neurons is time-consuming, but should always be followed for important tasks. Start with a number of neurons which is definitely too small. If coming up with a good guess for "too small" is impossible, start two neurons! Choose an appropriate criterion for evaluating the performance of the network. Train and test the network, recording its performance. Then slightly increase the number of hidden neurons, and train and test again. Repeat this until the error is acceptably small, or no significant improvement is noted, whichever comes first. It's brute force and it's slow, but it works. If validation sets are easily obtained, we may want to try adding neurons past the point of acceptable results, counting on the validation procedure to warn us of over fitting. Otherwise, we should be content with using the bare minimum number of hidden neurons necessary to achieve acceptable performance. Increasing the number beyond that minimum will often cause deterioration in generalization ability.

5.2. Optimal Network Architectures

We have seen that the network architecture is very important, and each application requires its own architecture.

To obtain good generalization ability, one has to build into the network as much knowledge about the problem as possible (e.g., the topology of the input space) and limit the number of connections appropriately. It is therefore desirable to find algorithms that not only optimize the weights for a given architecture, but also optimize the architecture itself. This means in particular optimizing the number of layers and the number of units per layer.

Of course there are various different criteria for optimal, including generalization ability, learning time, number of units, and so on. In fact, giving various hardware restrictions, there may be quite a complicated cost function for the architecture itself. We focus mainly on using as few units as possible; this should not only reduce computational costs and perhaps training time, but should also improve generalization (*Burattini, 1993*).

It is of course possible to mount a search in the space of possible architectures. We have to train each architecture separately by (say) back-propagation, and then evaluate it with an appropriate cost function that incorporates both performance and number of units. Such a search can be carried out by a genetic algorithm, so that good building blocks found in one trial architecture are likely to survive and be combined with good building blocks from other trials (*Hertz, 1991*). However, this kind of search seems unlikely to be practical for applications requiring large networks, where training just one architecture often requires massive CPU power.

More promising are approaches in which we construct or modify an architecture to suit a particular task, proceeding incrementally. There are two such ways to reach as few units as possible: Start with too many and take some away; or start with too few and add some more. In section 5.4 we follow the second way of optimizing the architecture. In this case it is necessary to retrain the network after each update of the network parameter, through this, retraining is usually rather fast.

The back-propagation algorithm described in Chapter 4 which is used for training networks has two major problems:

1. The algorithm is sometimes trapped in the local minima of the squared error function.
2. The weight coefficients after training are different, depending on the network configuration, and the initial values of weights.

Our solution to these problems is training a number of networks each operating with different initial weights distributed random range, different hidden nodes number, and using the learning schedule of Table 4.1 of the previous Chapter with uniform noise function and then compute the minimum value of the mean square error of their outputs.

Suppose that we have a realistic model with 13 inputs and 2 outputs, and in the next sections we construct the optimal network architectures for this model, as seen latter in Fig. 5.33 of Section 5.5.

5.3. Network Construction Algorithm

We used the following network parameters to construct the network:

1. Learning Rule (Delta) .
2. Activation Function .
3. Learning Rate (α) .
4. Momentum Coefficient (μ) .
5. Epoch Size .
6. Initial Weights Distributed Random Range (IWDRR)
7. Hidden Layers Number .
8. Hidden Nodes Number (H).
9. Tolerance Error.

The following procedure can be used to get the desired network, see Fig. 5.1. :

1. Construct the model .
2. Select the network architecture .
3. Prepare the training and test files .
4. Adjust the network parameters .
5. Training the network .
6. Test the network, compare the outputs with the desired outputs and calculate a measure of their difference. A common used method is the Mean Square Error (MSE) :

$$MSE = \frac{\sum_{i=1}^{n} (t_i - o_i)^2}{n}$$

where t_i is the target output

o_i is the desired output, and

n is the number of cases.

The object of training is to minimize this MSE, often called the objective function .

7. If the MSE is acceptable, then go to step 9.
8. If the situation needs a change of parameter, then go to step 4, otherwise go to step 2 .
9. Stop .

5.4. A Parametric Analysis

In this section, we conduct a parametric analysis, whose results should further improve the design and the performance of the architecture

of the model (*Lodewyck et al., 1993; Tam, 1992; Wilson, 1992; Eberlein et al., 1991*).

Our procedure based on the minimization of the objective function (MSE).

Experiments were conducted to determine the effect of several key network parameters of I networks (for I = 1 , ... , n ; n = 12). To simplify the presentation and concentrate on going insight , we illustrate our analysis in case of I = 1 , and hence we illustrate the trials of I cases (for I = 1 , ... , n; n = 12). We have two directions . In the first direction we used the sigmoid activation function and changing the other network parameters. In the second direction we used the tanh activation function with the adjusted network parameters of the first direction .

The network failed to converge if the testing tolerance is less than 0.002; this would imply an infinitely large number of runs.

We illustrate our analysis of the two directions in the following two subsections:

5.4.1. Sigmoid Architectures

In this section we illustrate our work to construct an optimal network architecture to solve our problem using the sigmoid activation function. Figure 5.2. illustrates the main processes to construct the architecture which minimize the objective function. As shown in this figure, we start with the delta learning rule, 0.4 momentum coefficient, 0.4 learning rate, one epoch size, and IWDRR = [- 0.1, 0.1] at step 1 and go through step 6 by changing one parameter in each step. We explain these

steps in detail in the following procedure for case I = 1. We initialize, train, and test the network after changing any parameter .

Figures 5.1, 5.2, 5.3, 5.4, and 5.5 are graphs of MSE as a function of hidden nodes number, epoch size, learning rate, momentum coefficient and activation function, respectively.

Step 1 : In this step we changed the number of hidden nodes from 0 - 100, see Fig. 5.1, we seen that the MSE (= 0.056882) is minimum at 4 hidden nodes and increased otherwise. The graph suggests that the hidden nodes number should be 4.

Step 2 : From the previous step , we construct the architecture 13-4-2 (this means that, 13 input nodes, 4 hidden nodes, and 2 output nodes) and changing the epoch size as shown in Fig. 5.2. The epoch size which minimize the MSE (= 0.056667) is equal to 2 .

Step 3 : By using the result of step 2 and by changing the learning rate from 0.02 - 0.75 we see, as shown in Fig. 5.3, that the minimum of MSE is equal to (0.058875) at 0.5 learning rate.

Step 4 : From step 2 and by changing the momentum coefficient from 0.2 - 0.9 we can see that the momentum, which minimize the MSE (= 0.056763), is equal to 0.4, as shown in Fig. 5.4.

Step 5 : By using the result of step 2 and by changing the activation function we get the MSE's minimum at sigmoid activation function with MSE = 0.064674, as shown in Fig. 5.5.

Step 6 : From the previous steps, we see that, the minimum value of the MSE is equal to (0.056664).

We try to reduce the MSE by increasing the IWDRR from [-0.1,0.1] to [-1, 1], see Fig. 5.6 for I = 1. It is clear that, MSE has the minimum value, which equal to (0.055331), at IWDRR = [-0.4, 0.4].

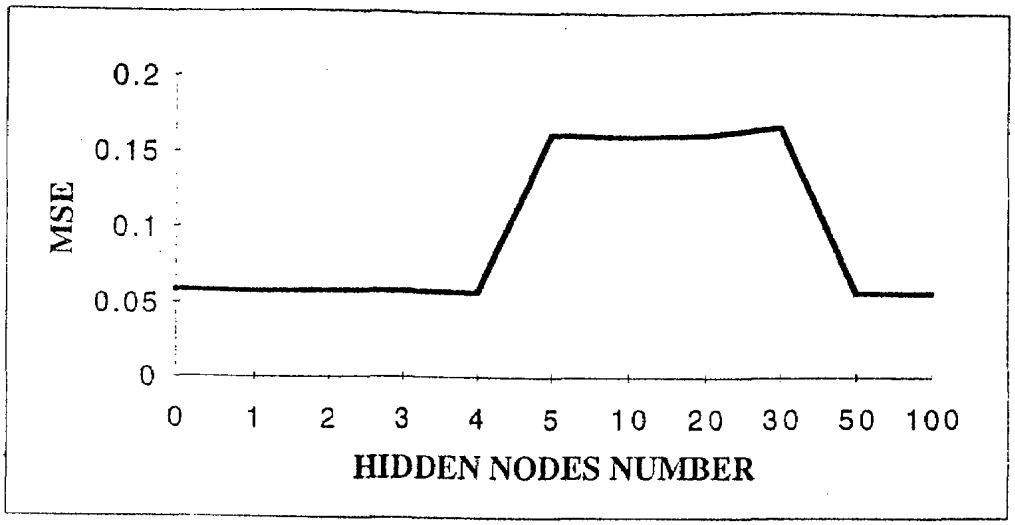


Fig. 5.1 MSE versus hidden layer size.

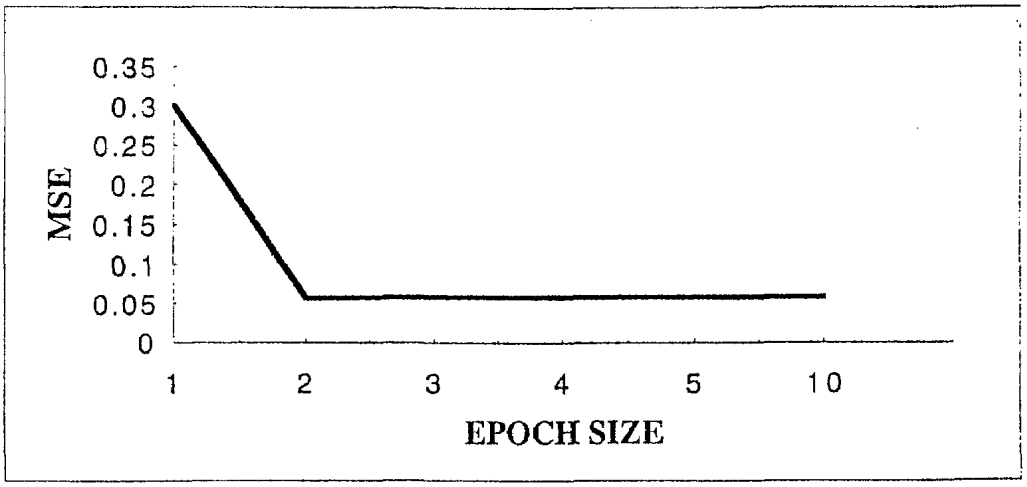


Fig. 5.2 MSE versus epoch size.

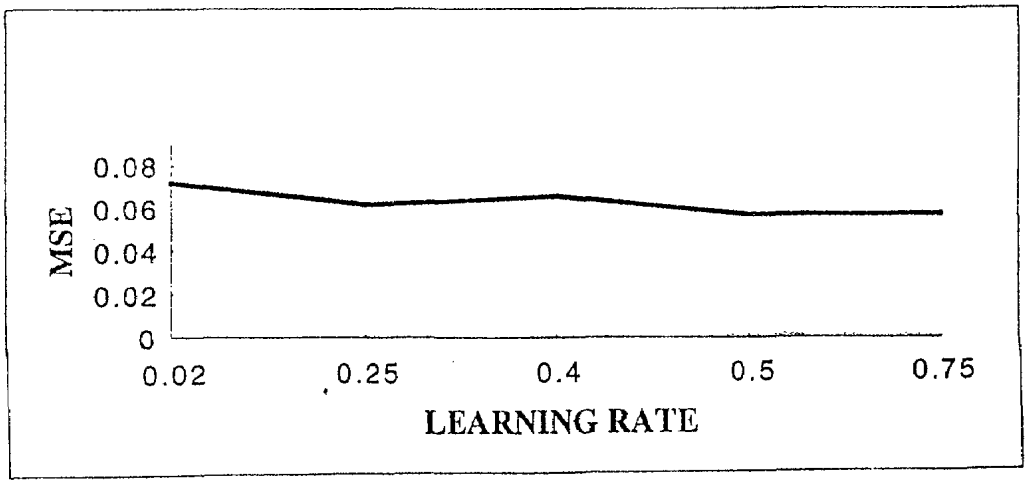


Fig. 5.3 MSE versus learning rate.

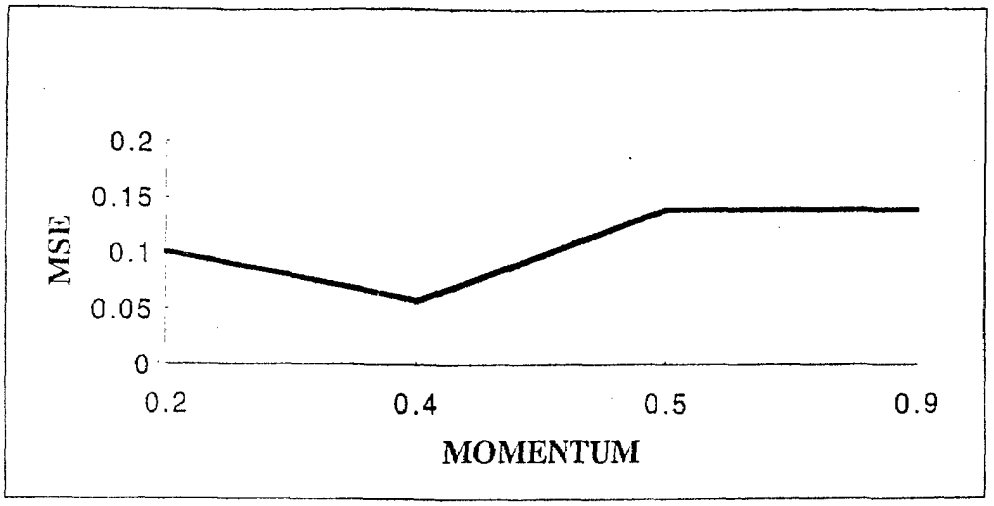


Fig. 5.4 MSE versus momentum coefficient.

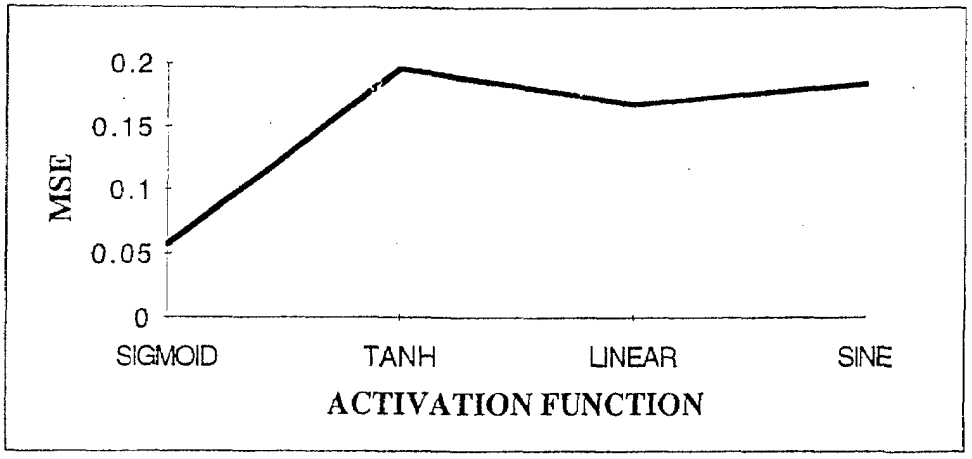


Fig. 5.5 MSE versus activation function.

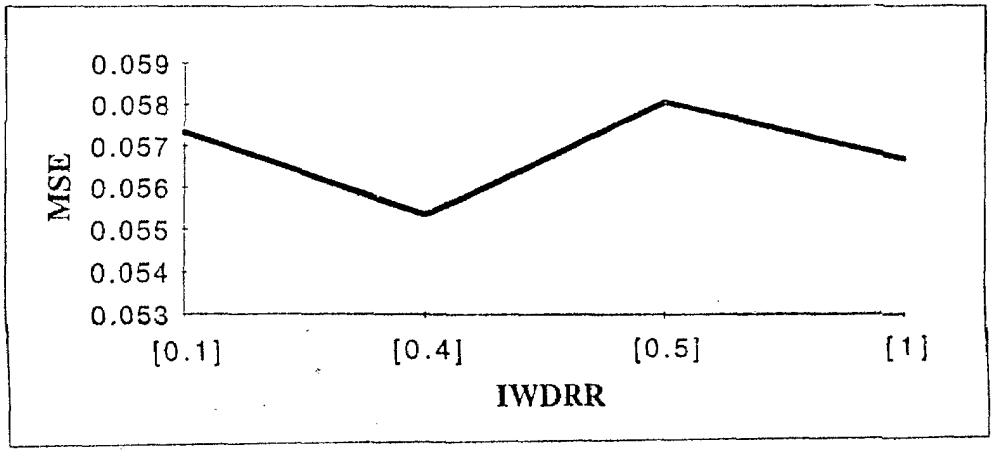


Fig. 5.6 MSE versus IWDRR.

We applied these values of the network parameters for the cases from I = 2 to I = 12 and plots the MSE of the results, as shown in Fig. 5.7 . From this figure, we see that, some cases (I = 5, 7, 8, 9) have MSE less than 0.03 and other cases (I = 11, 12) have MSE greater than 0.20.

The network of architecture 13-4-2 was trained after increasing the IWDRR over the range [-0.1, 0.1] - [-1, 1] with

- hidden layer size fixed at 4,
- learning rate fixed at 0.5,
- momentum coefficient fixed at 0.4,
- training tolerance fixed at 0.002,
- epoch size fixed at 2, and
- sigmoid activation function.

Figures 5.7 and other trails in case of IWDRR equal to [-0.1, 0.1], [-0.4,0.4], [-0.5,0.5] and [-1,1], we see that, there is successively decreasing of the MSE. The average of MSE is (0.078) in case of IWDRR = [-1,1].

For I=1, by increasing the IWDRR [-5, 5], the MSE reduced to (0.036798) at [-3, 3].

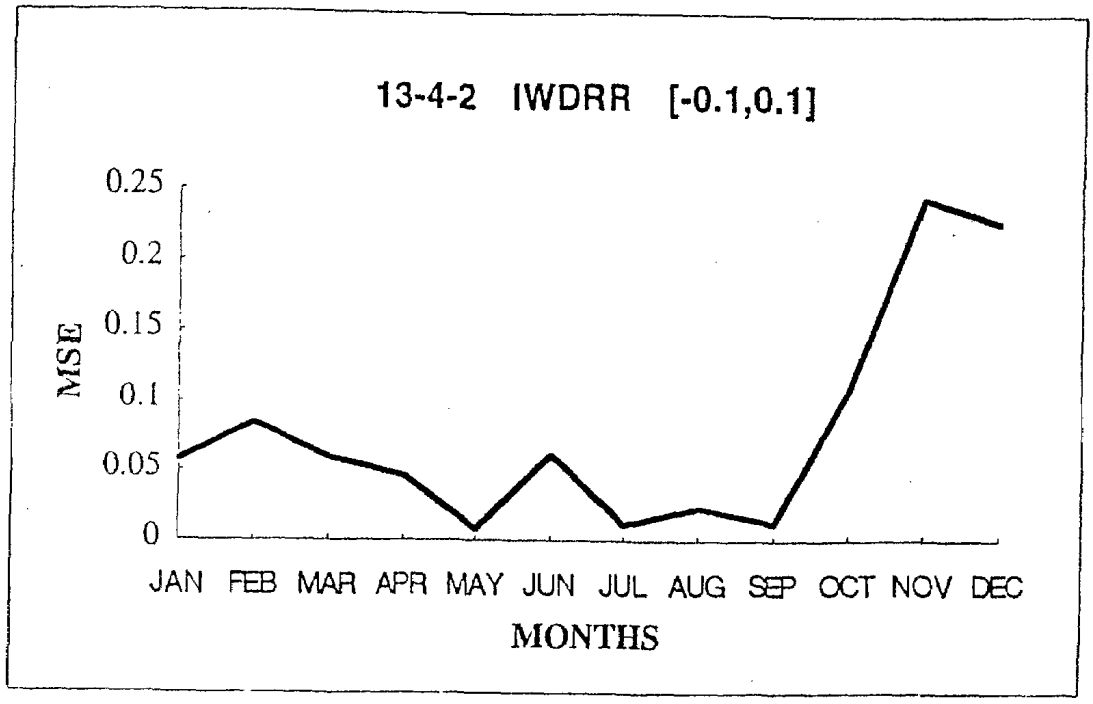


Fig. 5.7 MSE versus months in case of IWDRR=|-0.1, 0.1|
and architecture 13-4-2.

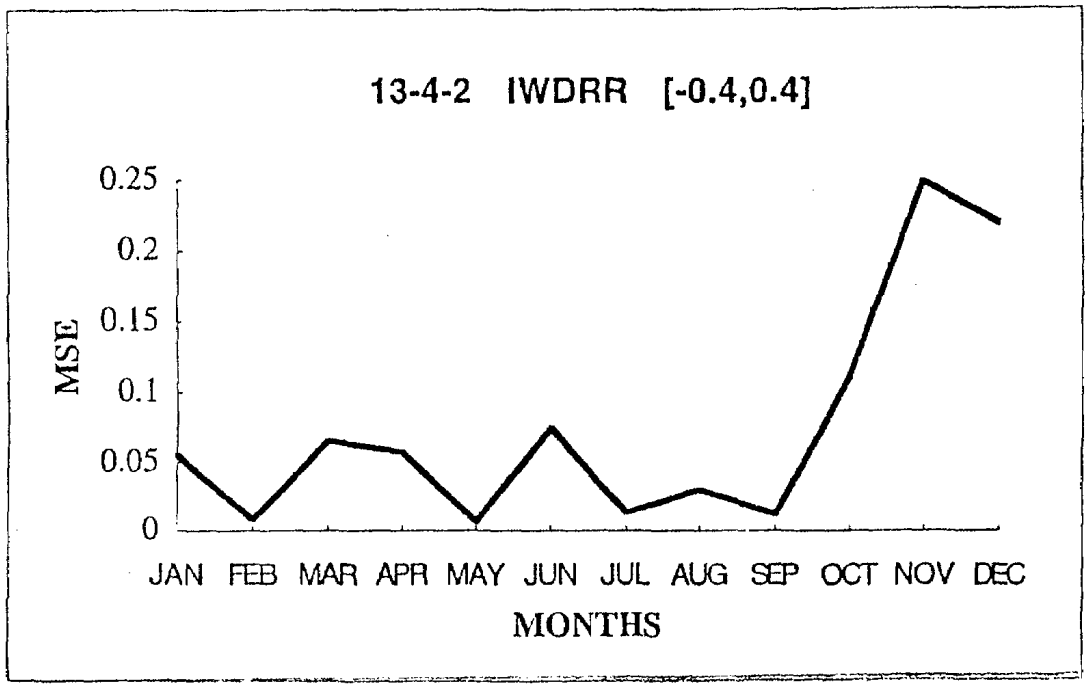


Fig. 5.8 MSE versus months in case of IWDRR=|-0.4, 0.4|
and architecture 13-4-2.

5.4.2. Tanh Architectures

In this section we used the tanh activation function, and adjusted the other network parameters as shown in the last step of Fig. 5.2 .

(13-10-2) Architecture

In this case we started with 10 hidden nodes and increase the initial weights random range from [-0.1,0.1] to [-1,1] for case I (I = 1 , ... , 12). We initialize, train, and test the network for each case of I in each range of the initial weights. Then, we compute the MSE for each case and plot its results as shown in figures 5.9, 5.10.

The network of architecture 13-10-2 was trained after increasing the IWDRR over the range [-0.1, 0.1] - [-1, 1] with

- hidden layer size fixed at 10,
- learning rate fixed at 0.5,
- momentum coefficient fixed at 0.4,
- training tolerance fixed at 0.002,
- epoch size fixed at 2, and
- tanh activation function.

From fig. 5.9, 510 and other trails in case of IWDRR equal to [-0.1,0.1], [-0.4, 0.4], [-0.5,0.5], and [-1,1], we see that, the average of the MSE is 0.09 in case of IWDRR=[-0.1,0.1] and decreased to 0.06 in case of IWDRR = [-0.4,0.4] and IWDRR= [-1,1].

It is clear, from Fig. 5.9, that the MSE is less than 0.05 in case of (I = 5, 7, 8, 9, 10, 11) and greater than 0.25 in case of (I = 1, 3). But in Fig. 5.10 the MSE is less than 0.02 in case of (I = 5, 6, 8, 9) and greater than 0.08 in case of (I = 1, 3, 7).

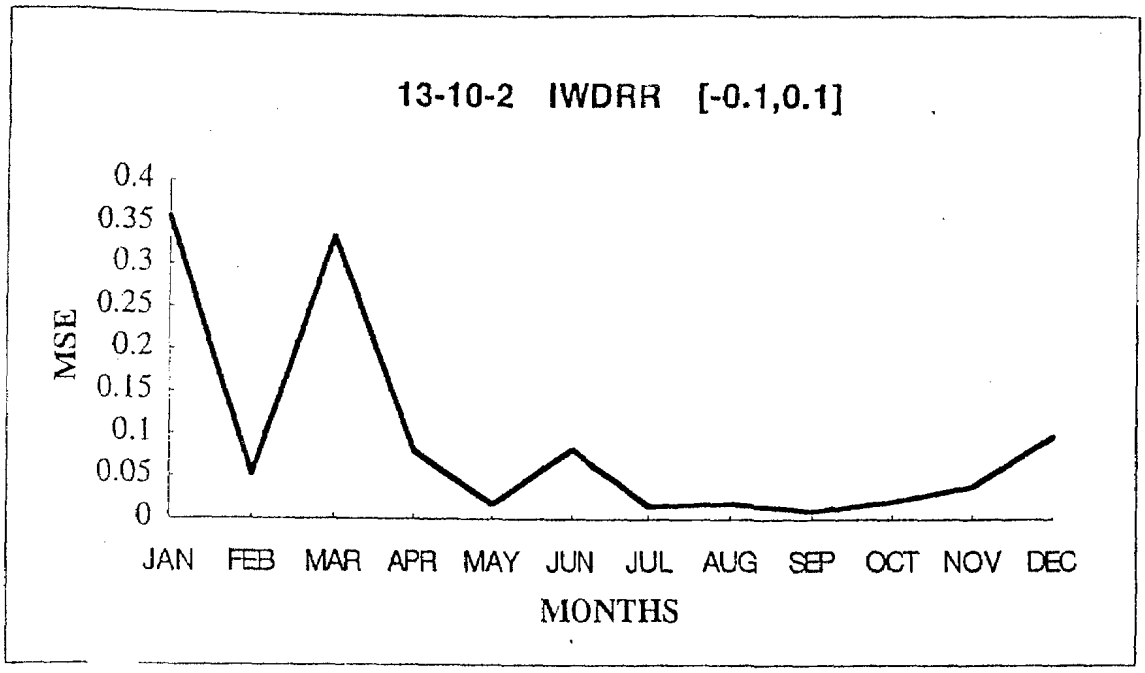


Fig. 5.9 MSE versus months in case of IWDRR=[-0.1, 0.1]
and architecture 13-10-2.

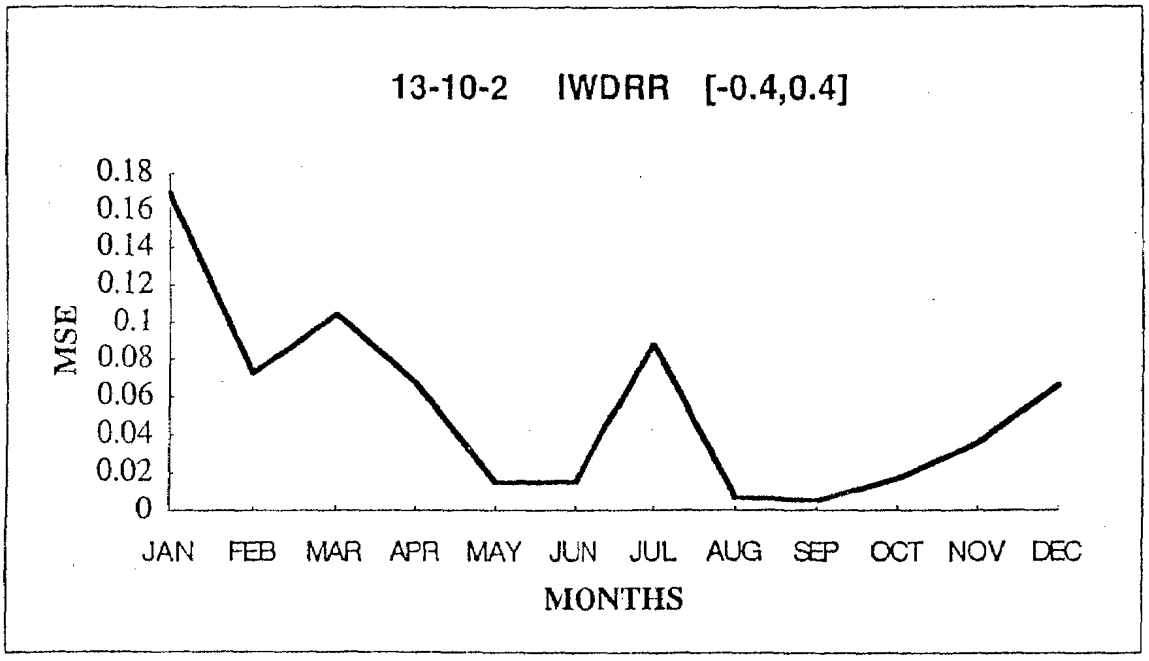


Fig. 5.10 MSE versus months in case of IWDRR=[-0.4, 0.4].
and architecture 13-10-2.

(13-20-2) Architecture

In this case, we increased the hidden nodes to 20 hidden nodes with the same network parameters of the above case. We illustrated the graphs of the MSE of this architecture, 13-20-2, in Figures 5.18, 5.19, 5.20 and 5.21.

The network of architecture 13-20-2 was trained after increasing the IWDRR over the range [-0.1, 0.1] - [-1, 1] with

- hidden layer size fixed at 20,
- learning rate fixed at 0.5,
- momentum coefficient fixed at 0.4,
- training tolerance fixed at 0.002,
- epoch size fixed at 2, and
- tanh activation function.

From Fig. 5.9, 5.10 and other trails in case of IWDRR equal to

[-0.1, 0.1], [-0.4, 0.4], [-0.5, 0.5], and [-1, 1], we say that, the average of the MSE is 0.08 in case of IWDRR=[-0.1,0.1] and decreased to 0.058 in case of IWDRR = [-0.5, 0.5].

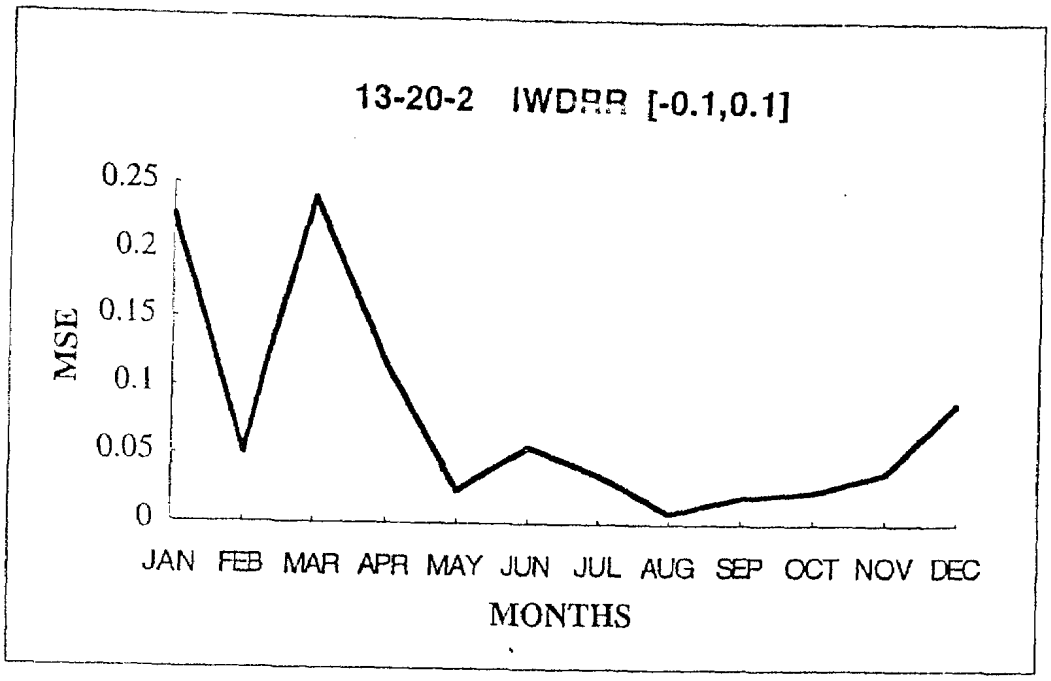


Fig. 5.11 MSE versus months in case of IWDRR=|-0.1, 0.1|
and architecture 13-20-2.

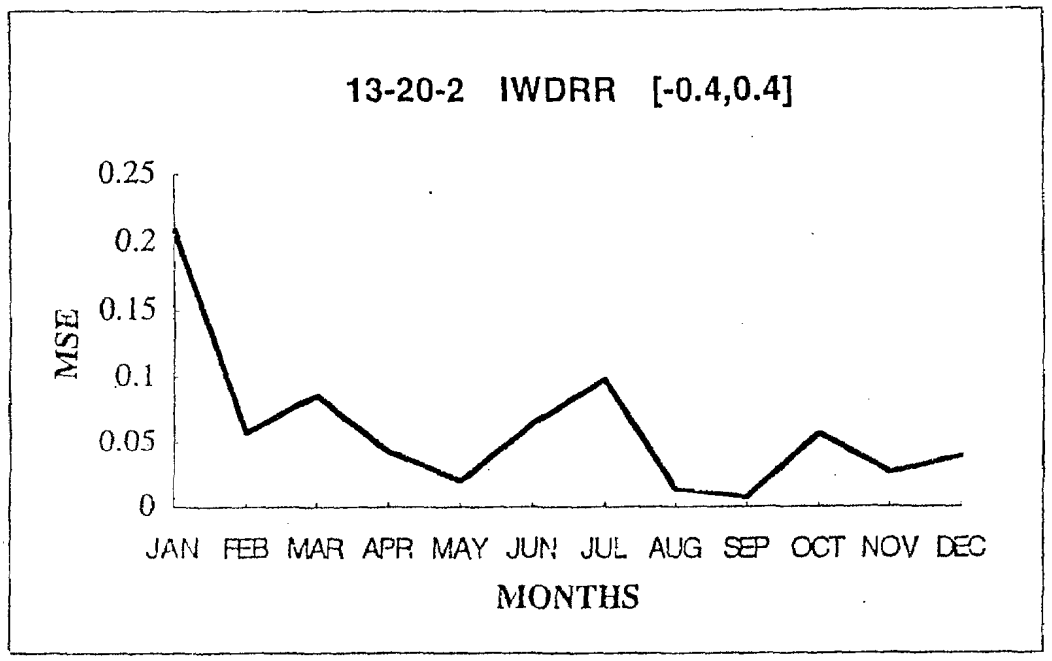


Fig. 5.12 MSE versus months in case of IWDRR=|-0.4, 0.4|
and architecture 13-20-2.

(13-30-2) Architecture

In this case, we used 30 hidden nodes and also the same network parameters as in the previous case. The results of this architecture are shown in Figures from 5.22 to 5.26.

The network of architecture 13-30-2 was trained after increasing the IWDRR over the range [-0.1, 0.1] - [-1, 1] with

- hidden layer size fixed at 30,
- learning rate fixed at 0.5,
- momentum coefficient fixed at 0.4,
- training tolerance fixed at 0.002,
- epoch size fixed at 2, and
- tanh activation function.

Figures 5.22, 5.23, 5.24, 5.25 and 5.26 are graphs of MSE as a function of month in case of IWDRR equal to [-0.1, 0.1], [-0.4, 0.4], [-0.5, 0.5], [-0.6, 0.6], and [-1, 1], respectively.

The average of the MSE is 0.07 in case of IWDRR = [-0.1, 0.1] and decreased to 0.0568 in case of IWDRR = [-1, 1].

It is clear that there is some similarity in patterns of the MSE in Figures 5.14, 5.18, and 5.22 of the architectures 13-10-2, 13-20-2, and 13-30-2 in case of IWDRR = [-0.1, 0.1].

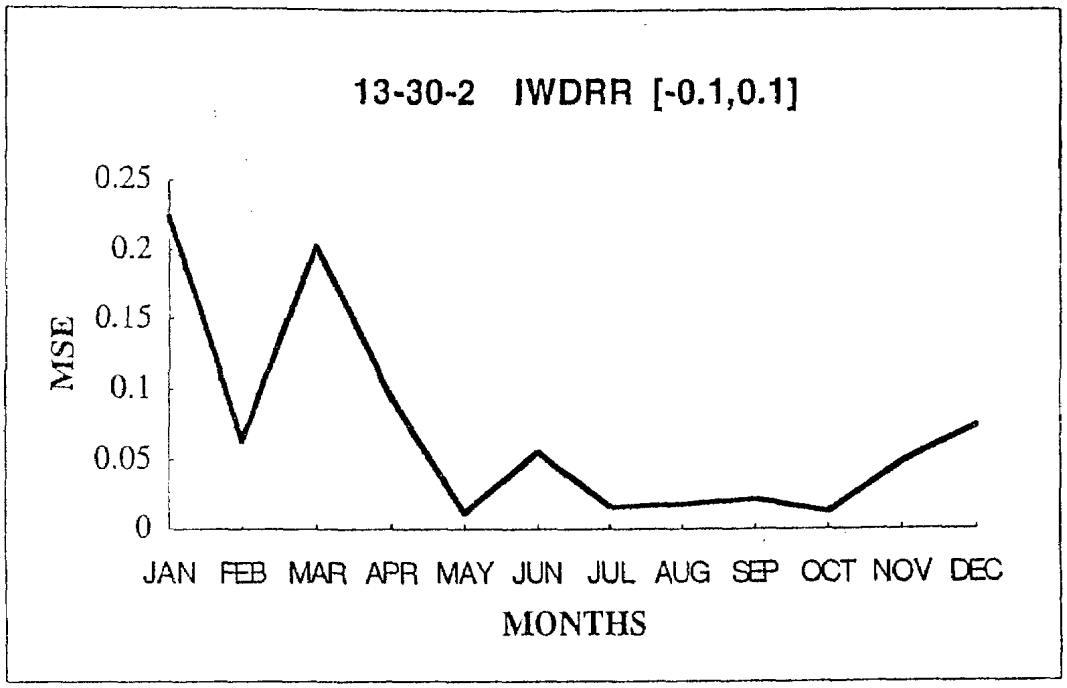


Fig. 5.13 MSE versus months in case of IWDRR=[-0.1, 0.1] and architecture 13-30-2.

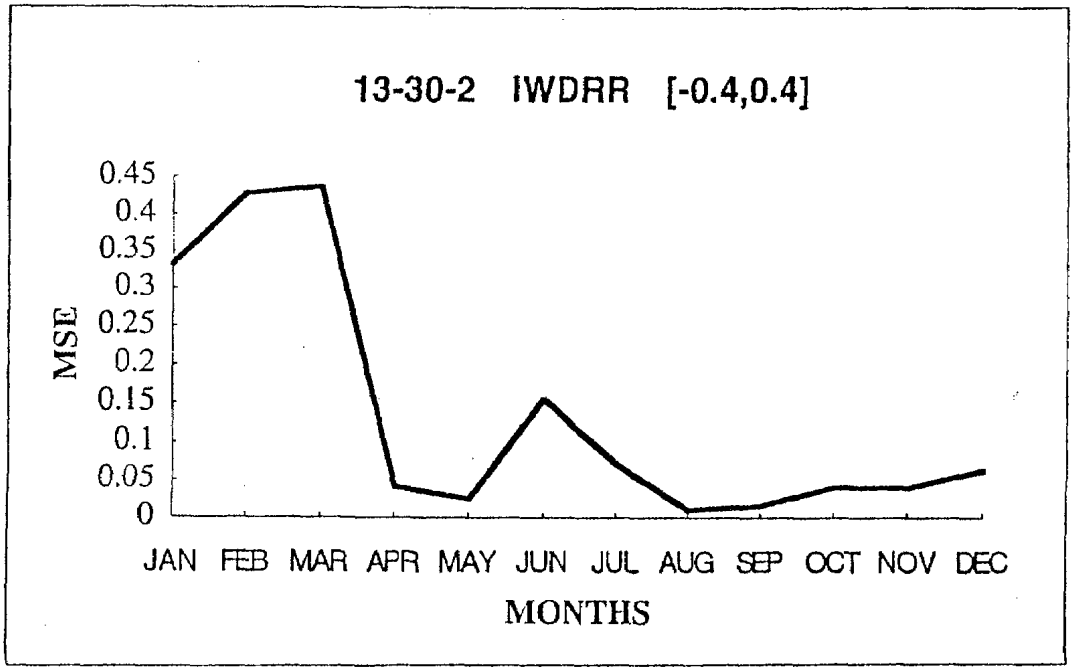


Fig. 5.14 MSE versus months in case of IWDRR=[-0.4, 0.4] and architecture 13-30-2.

5.5. The Result Networks

We summarized the results of the previous two sections in Table 5.1. This table illustrates the minimum values of the MSE of all architectures for each case I (for I = 1 , ... , 12). We write the MSE and IWDRR of each case in each cell in the table. The last column represents the minimum values of the MSE of each row. The average of the MSE of each architecture is written in the last row. It is clear that the average of the last column is equal to 0.0154, and these architectures are powerful to obtain an excellent prediction for case I (I = 2, ... , 12), as seen later in the prediction section in Chapter 6.

In case of I = 1 the MSE is equal to 0.0367 at IWDRR = [-3,3], it is clear that the MSE is a good result, but the prediction of this case may will not be acceptable, because the IWDRR must be small.

For this reason, we increased the hidden nodes of the architecture 13-10-2 one hidden node in each time until we get a good result at minimum IWDRR. In this case we see that the desired result is at hidden nodes number equal to 12, and the MSE = 0.0356 at IWDRR = [-1, 1].

The final result networks for I cases (I = 1, ... , 12) are illustrated in Table 5.1. We also illustrated the graph of the MSE of the final results in Fig. 5.15.

Table 5.1

NO.	MONTHS	*MSE* *ARCHITECTURE* *IWDRR*	NO.	MONTHS	*MSE* *ARCHITECTURE* *IWDRR*
1	JAN.	0.0356 13-12-2. [-1,1]	7	JUL.	0.0107 13-4-2. [-0.1,0.1]
2	FEB.	0.0079 13-4-2. [-0.5,0.5]	8	AUG.	0.0065 13-10-2. [-0.4,0.4]
3	MAR.	0.0074 13-10-2. [-1,1]	9	SEP.	0.0051 13-10-2. [-0.4,0.4]
4	APR.	0.0313 13-30-2. [-0.6,0.6]	10	OCT.	0.012 13-30-2. [-0.1,0.1]
5	MAY.	0.0045 13-20-2. [-1,1]	11	NOV.	0.0086 13-30-2. [-0.6,0.6]
6	JUN.	0.0152 13-10-2. [-0.4,0.4]	12	DEC.	0.0389 13-20-2. [-0.4,0.4]
MSE AVERAGE					0.0153

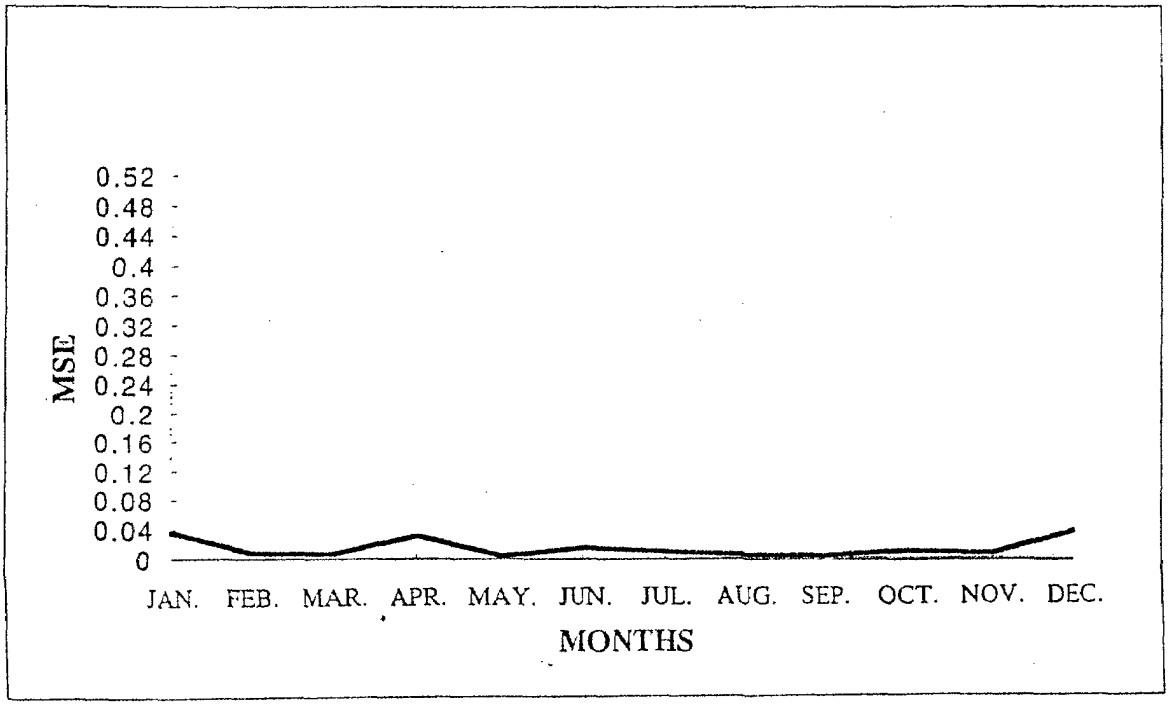


Fig. 5.15 MSE versus months of the networks of the final results.

5.6. Nile Water Distribution and Underground Water Extraction : A Case Study

The decision of specifying the amount of water that should be released from Nasser lake is a complicated process. The MPWWR* determines this decision for each month based on the current level of the water in the Nasser lake, the needs of the agriculture, and many other factors. We consider the Nile water distribution (released water) problem and the underground water extraction problem as one problem. A neural networks approach can be used for solving this problem.

Planning for water distribution system development may be viewed as a neural network with a three-layer architecture consisting of input, hidden, and output layers. Though the architecture of multiple hidden layers is also possible, all of its functions can be accomplished by a three-layered network (*Wang, 1994; Jung, 1994; Burattini, 1993; Schocken et al., 1991*).

Each node in one layer is connected in the forward direction to every node in the next layer.

In this model, the continuous and differential tanh function is adopted as a node's transfer mechanism; it translates the activation value into an output value. We also used the sigmoid activation function.

Our system uses the back-propagation learning algorithm to modify the weight associated with the connection nodes. This strategy generates multiple output values : A set of values for planning end user development.

---

*) Ministry of public works and water resources .

We illustrated the approach with a neural network consisting of 13 input nodes, H hidden layer nodes, and 2 output nodes. The input nodes are the attributes we identified as important to this planning task.

A data set representing 28 patterns was gathered from the historical records for 28 years. The acquired data set was divided into two sets: The first one had 23 patterns and was used as the training set, the second served as the validation set. Each pattern includes the measured of the Nile released water and the extraction of the underground water.

The resulting architectures have 13 input nodes, number of hidden nodes (as illustrated in Table 5.1), and 2 output nodes, as shown below.

During the training phase, the back-propagation algorithm with momentum stochastically searched the weight space to find the optimal weights .

After the training phase, the validation set (5 patterns) was used to observe how the architecture will perform in real situations.

This dynamic model is used to predict the water distribution system. The output prediction taken by 12 architectures, and the results of this prediction are illustrated in the next chapter.

In the following we illustrate the network inputs and outputs.

Each of the input attributes is denoted by XJI; they are defined as follows :

The Network Inputs

The following inputs for case I , (I = 1 , ... , n; n = 12) .

1- Total water arrived Nasser lake in previous year (X1I) .

2- Total water prediction arrive Nasser lake in this year (X2I) .

3- Total water prediction arrive Nasser lake of the next year (X3I) .

4- Total water prediction released downstream Aswan barrage in previous year (X4I) .

5- Total water prediction released downstream Aswan barrage in this year (X5I) .

6- Total water prediction released downstream Aswan barrage of the next year (X6I) .

7- Water arrived Nasser lake in previous year of month I (X7I) .

8- Water released downstream Aswan barrage in previous year of month I (X8I) .

9- Total underground water actually demanded until month I (X9I) (Which is start with 0).

10- Residual underground water inventory until month I (X10I) . (Which is start with 500 (Total size of the underground water in the Nile Valley and the Nile Delta)).

11- Water demanded in previous year of month I (X11I) .

12- Demand lower bound of month I (X12I), (which is equal to $\frac{2.66}{12}$).

13- Demand upper bound of month I (X13I), (which is equal to 10).

Output from the layer is received by the hidden layer, which allows the hidden layer to develop complex feature detectors or internal representations. The number of nodes in the hidden layer is treated as a parameter of the network in Section 5.4.

We used a set of representative values to train the system. While the inputs may range from -1 to 1 we restricted them to a range from -0.8 to 0.8 in order to increase the likelihood convergence of the learning algorithm.

The output layer includes a set planning strategies for system development in an end user development environment. We have used Y1I and Y2I as symbols to represent two values, as follows (*NSSAR,1996*)

The Network Outputs

The following outputs for case for case I,(I = 1 , ... , n ; n = 12).

1- Water released downstream Aswan barrage of month I of next year (Y1I).
2- Water demand for month I of the next year (Y2I) .

5.7 Prediction by Using the Idea Models

In many situations in science and technology we want to predict the future evaluation of a system from past measurements on it. This is in fact central procedure of neural networks: we make a neural networks model of a system and shift the output to predict future state.

In a back- probagation network, data is required both at the input buffer and at the output buffer, as shown in chapter 4. Typically, each presentation of data is represented by a data record consisting of a set of input fields followed by a set of desired output fields.

If we construct this data record such that the input part is of year T and the desired output is of year T+Y, then training network gives the prediction of a year after Y years. This is our procedure to obtain the prediction of the water distribution for the future.

In table 5.2, we present prediction after one year (1996) (using the data of year 1995 as the start point), and after two years (1997) for the released water and the underground water extraction. We obtain the predictions by applying the models in table 5.1 in the previous chapter.

In table. 5.3, we present prediction after three years (1998), and after four years (1999) for the released water and the underground water extraction. We obtain the predictions by applying the models in table 5.1 in the previous chapter.

In fig 5.16, we illustrated the relation between predicted values and the actual values of the released water in year 1996. The graphs in this figure shown that the response of the networks will be correct. Fig. 5.17 presents the planning of the underground water extraction in year 1997.

Table 5.2

NO.	MONTHS	1996		1997		ACTUAL RELEASED WATER 1996
		PREDICTED RELEASED WATER	PLANNING UNDERG. WATER EX.	PREDICTED RELEASED WATER	PLANNING UNDERG. WATER EX.	
1	JAN.	2.626878	0.246565	2.756745	0.256199	2.58
2	FEB.	3.610272	0.280485	3.704278	0.284524	3.425
3	MAR.	4.442094	0.375451	4.476484	0.36359	4.635
4	APR.	4.454094	0.368185	4.519665	0.404914	4.5
5	MAY.	5.351599	0.455938	5.621789	0.46626	5.795
6	JUN.	7.72671	0.694053	7.783246	0.694078	7.75
7	JUL.	7.139198	0.632253	7.158447	0.635098	7.17
8	AUG.	6.297251	0.556621	6.26923	0.553584	6.442
9	SEP.	4.581002	0.403706	4.514484	0.396192	4.210
10	OCT.	3.603343	0.312458	3.702872	0.3268	3.46
11	NOV.	3.353824	0.289562	3.378147	0.307414	3.375
12	DEC.	2.69025	0.259395	2.328234	0.251193	2.21
SUM		55.87652	4.874672	56.21362	4.939846	55.552

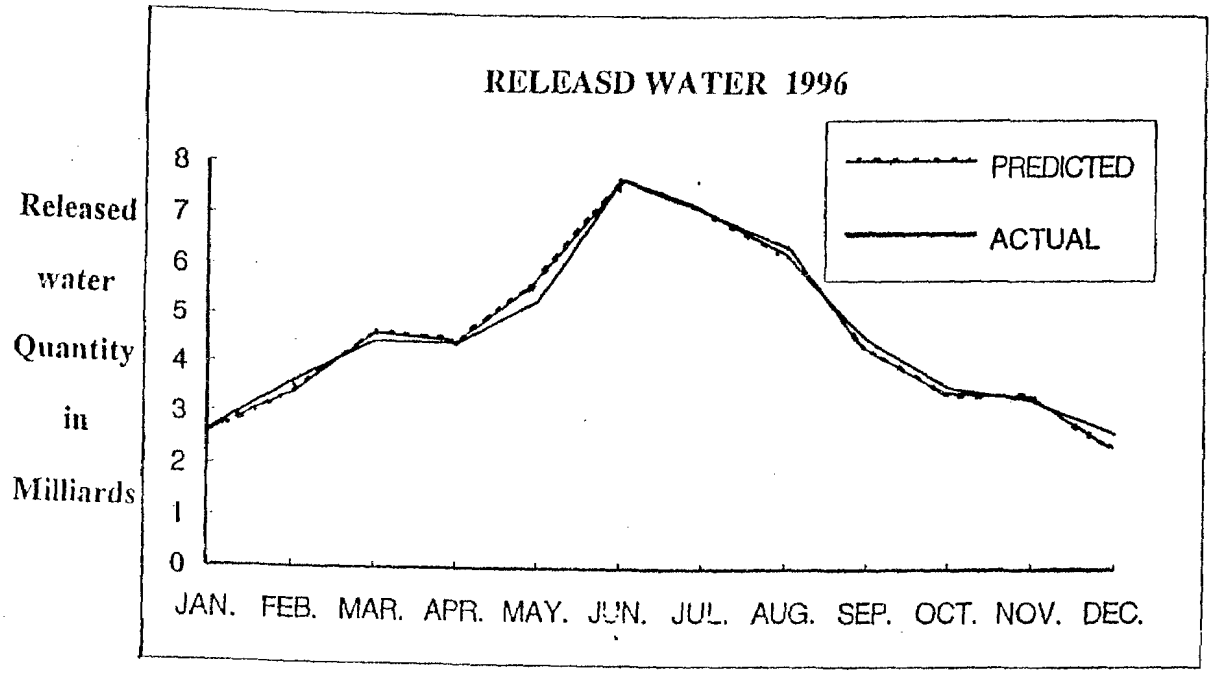


Fig. 5.16 Released water quantity versus months of year 1996.

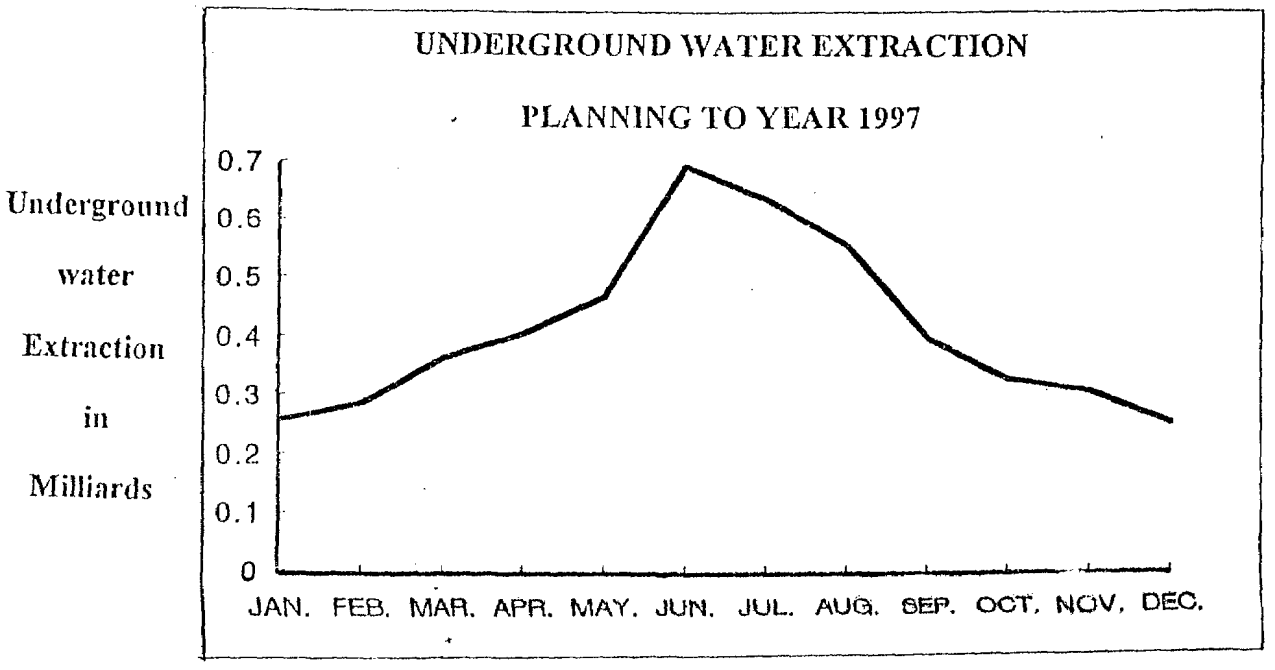


Fig. 5.17 Underground water extraction versus months to year 1997.

Table 5.3

NO	MONTHS	1998		1999	
		PREDICTED RELEASED WATER	PLANNING UNDERG. WATER EX	PREDICTED RELEASED WATER	PLANNING UNDERG. WATER EX
1	JAN	2.647591	0.325077	2.60544	0.323586
2	FEB	3.70427	0.28452	3.04558	0.303597
3	MAR	3.83081	0.344168	3.95497	0.352244
4	APR	4.07258	0.36177	4.064364	0.360839
5	MAY	5.595089	0.499109	5.514745	0.494028
6	JUN	7.10726	0.62323	6.04315	0.534581
7	JUL	6.82551	0.605799	6.3392	0.614928
8	AUG	6.38048	0.565648	6.387271	0.566424
9	SEP	4.54386	0.404579	4.541783	0.404093
10	OCT	3.74957	0.334418	3.766	0.335791
11	NOV	3.220359	0.29407	3.167121	0.287326
12	DEC	2.788137	0.248979	2.892439	0.256256
TOTAL		54.465516	4.891367	52.322063	4.833693

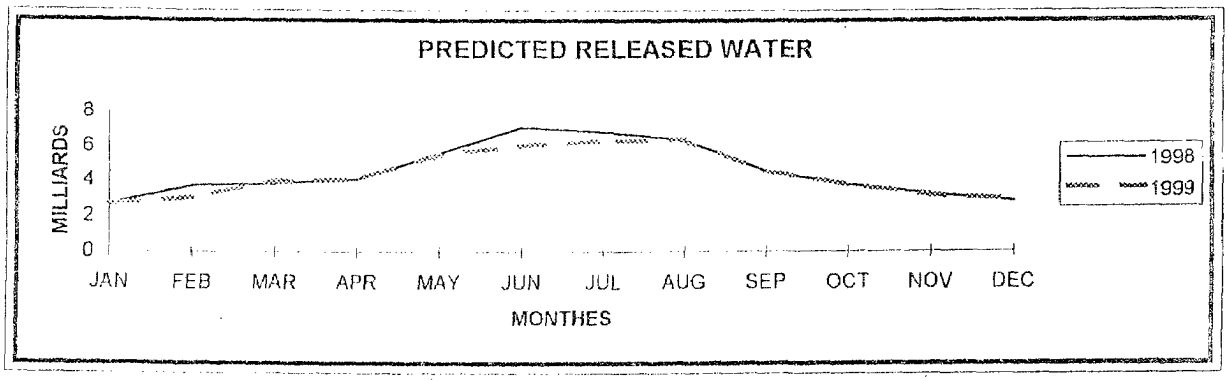


Fig. 5.16 Released water quantity versus months of year 1998

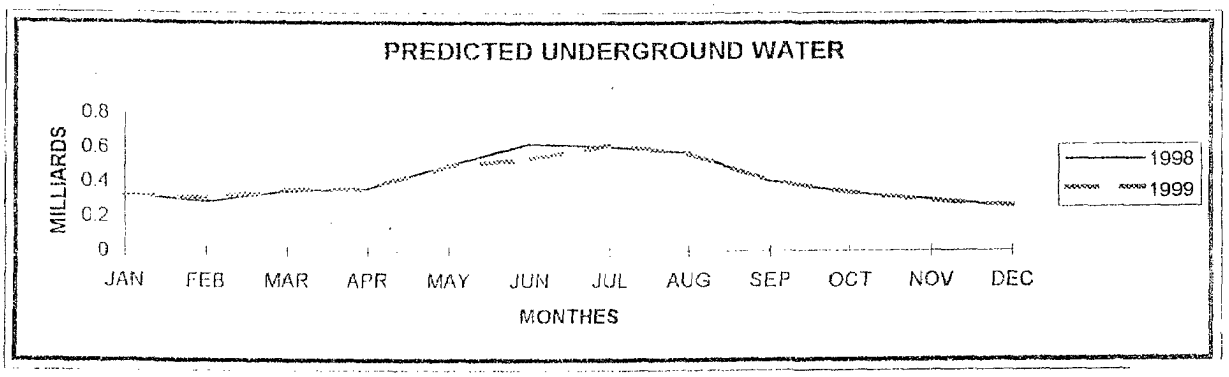


Fig. 5.17 Underground water extraction versus months to year 1998

Conclusion

The Nile Valley and Delta underground water reservoirs can assist in saving irrigation water for some new reclaimed areas located at the eastern and western desert fringes. The underground water can be pumped from suitable supply areas of these fringes. The pumped water may be used directly in the site or indirectly by conveying it to these demand areas.

The results which show that the optimum solution of this policy are:

1. The current annual rate of extraction of underground water in the Nile Valley and Delta for domestic, industrial and agricultural uses is 2.66 billion m^3/year (*Diab, 1992; Biswas, 1991*). This can be increased to 4.9 billion m^3/year (see the resulted table 5.2 in Chapter 5) which is estimated to be equivalent to annual recharge rate. Hence, this value increasing underground water extraction, 2.24 billion m^3/year will yield an irrigation of 350000 Feddans.
2. The total underground water inventory in the Nile Valley and Delta is 500 billion m^3. This inventory of underground water can be used for the irrigation of 750000 Feddans for 100 years.
3. In addition there is 40000 billion m^3 under the western desert, which can be utilized to irrigate 250000 Feddens for 100 years.

The proposed model (Chapter 3) as well as the neural network model (Chapter 5) could be applied to irrigate 1.35×10^6 Feddans.

In addition to these results we can use about 1.4 billion m^3 /year rainfall for irrigating new lands as in El-Arish city (*Allam et al., 1992*).

The neural network model, in chapter 5, could be applied to predict the response of Nile Valley, Delta, and the desert region aquifers system to any underground water extraction plan based on the following development criteria: The planned extraction policies might be based on the strategy suitable for the Nile Valley, Delta, and the desert underground water reservoirs.

Additional Water Supply:

In order to compensate the future water demands, desalination techniques may be the unique solution, as the following sequence:

1. Reuse of the brackish drainage water (16-18 billion m^3) after removing salts by using suitable membrance systems such as reverse osmosis or electrodialysis .
2. Use of desalted sea water, whether distilled by nuclear power or by conventional fuel .

From the given presentation, it could be concluded that the use of the desalted brackish drainage water is urgently needed. As a second step, the use of the desalted sea water seems to be a must, as far as the agriculture development, in Egypt, is concerned.

Underground water loss from natural springs in the Western Desert Oases occurs through evaporation, which is exceptionally high in our region .

There is a water loss from the Nubian artesian aquifer estimated at 1.32 x 108 m^3/year (Diab, 1992) .

A good part of this water could be saved by channelizing many of the springs, which issue its water to the ground surface, from which it is lost to the atmosphere through natural evaporation .

Sinai is an important region for the future development of Egypt, but little is known about its water resources. The major drainage basin of Sinai is Wadi El-Arish which occupies one third of the surface area of the peninsula.

A complete geophysical study for these subbasins is required to identify the underground aquifers in size and extend to show the possibility of the underground water recharge and the storage capability .

Installation of a complete network of meteorological and hydrological automatic stations to cover the mentioned subbasins to collect the required records for accurate studies .

Also, establishing a computer database for these records is essential to facilitate the storage and the analysis of these records .

Appendix

This appendix contains code listing of Nworks of the backpropagation network. The following is list of all neural network parameters of case I=1.

Title: Network

Display Mode : Network
Type : Hetero-Associative
Display Style : default
Control Strategy: bkpfast
L/R Schedule : backprop
31646 Learn 1 Recall 0 Layer
2 Aux 1 0 Aux 2 0 Aux 3

L/R Schedule: backprop

Recall Step	1	0	0	0	0
Input Clamp	0.0000	0.0000	0.0000	0.0000	0.0000
Firing Density	100.0000	0.0000	0.0000	0.0000	0.0000
Temperature	0.0000	0.0000	0.0000	0.0000	0.0000
Gain	1.0000	0.0000	0.0000	0.0000	0.0000
Learn Step	5000	0	0	0	0
Coefficient 1	0.9000	0.0000	0.0000	0.0000	0.0000
Coefficient 2	0.6000	0.0000	0.0000	0.0000	0.0000
Coefficient 3	0.0000	0.0000	0.0000	0.0000	0.0000
Temperature	0.0000	0.0000	0.0000	0.0000	0.0000

IO Parameters
Learn Data: File Rand. (toshk1a) Binary
Recall Data: File Seq. (toshk1b)
Result File: Desired Output, Output

UserIO Program: userio
I/P Ranges: 0.0000, 1.0000
O/P Ranges: 0.2000, 0.8000
I/P Start Col: 1 MinMax Table: toshk1a
O/P Start Col: 14 Number of Entries: 15

MinMax Table <toshk1a>:

Col:	1	2	3	4	5	6
Min:	32.6120	32.6120	32.6120	52.0800	52.0800	52.1000
Max:	88.41	88.41	88.41	62.15	62.15	62.2
Col:	7	8	9	10	11	12
Min:	1.2620	2.0000	0.2200	0.0000	500.0000	0.2210
Max:	3.595	4.45	0.4	0	500	0.221
Col:	13	14	15			
Min:	2.0000	0.2600	0.2600			
Max:	2.9	4.45	0.39			

Layer: 1

PEs: 1 Wgt Fields: 2 Sum: Sum
Spacing: 5 F' offset: 0.00 Transfer: Linear
Shape: Square Output: Direct
Scale: 1.00 Low Limit: -9999.00 Error Func: standard
Offset: 0.00 High Limit: 9999.00 Learn: --None--
Init Low: -0.100 Init High: 0.100 L/R Schedule: (Network)
Winner 1: None Winner 2: None
PE: Bias
1.000 Err Factor 0.000 Desired
0.000 Sum 1.000 Transfer 1.000 Output
0 Weights -1.692 Error 0.000 Current Error

Layer: In
PEs: 13 Wgt Fields: 1 Sum: Sum
Spacing: 5 F' offset: 0.00 Transfer: Linear
Shape: Square Output: Direct
Scale: 1.00 Low Limit: -9999.00 Error Func: standard
Offset: 0.00 High Limit: 9999.00 Learn: --None--
Init Low: -0.100 Init High: 0.100 L/R Schedule: (Network)
Winner 1: None Winner 2: None

PE: 2

1.000 Err Factor	0.294 Desired	
0.294 Sum	0.294 Transfer	0.294 Output
0 Weights	0.000 Error	0.000 Current Error

PE: 3

1.000 Err Factor	0.192 Desired	
0.192 Sum	0.192 Transfer	0.192 Output
0 Weights	0.000 Error	0.000 Current Error

PE: 4

1.000 Err Factor	1.000 Desired	
1.000 Sum	1.000 Transfer	1.000 Output
0 Weights	0.000 Error	0.000 Current Error

PE: 5

1.000 Err Factor	0.302 Desired	
0.302 Sum	0.302 Transfer	0.302 Output
0 Weights	0.000 Error	0.000 Current Error

PE: 6

1.000 Err Factor	0.232 Desired	
0.232 Sum	0.232 Transfer	0.232 Output
0 Weights	0.000 Error	0.000 Current Error

PE: 7

1.000 Err Factor	0.000 Desired	
0.000 Sum	0.000 Transfer	0.000 Output
0 Weights	0.000 Error	0.000 Current Error

PE: 8

1.000 Err Factor	0.327 Desired	
0.327 Sum	0.327 Transfer	0.327 Output
0 Weights	0.000 Error	0.000 Current Error

PE: 9

1.000 Err Factor	0.241 Desired	
0.241 Sum	0.241 Transfer	0.241 Output
0 Weights	0.000 Error	0.000 Current Error

PE: 10

1.000 Err Factor	0.111 Desired	
0.111 Sum	0.111 Transfer	0.111 Output
0 Weights	0.000 Error	0.000 Current Error

```
    1.000 Err Factor     0.500 Desired
    0.500 Sum          0.500 Transfer       0.500 Output
      0 Weights        0.000 Error          0.000 Current Error
  PE: 12
    1.000 Err Factor     0.500 Desired
    0.500 Sum          0.500 Transfer       0.500 Output
      0 Weights        0.000 Error          0.000 Current Error
  PE: 13
    1.000 Err Factor     0.500 Desired
    0.500 Sum          0.500 Transfer       0.500 Output
      0 Weights        0.000 Error          0.000 Current Error
  PE: 14
    1.000 Err Factor     0.000 Desired
    0.000 Sum          0.000 Transfer       0.000 Output
      0 Weights        0.000 Error          0.000 Current Error

Layer: Hidden1

     PEs: 12     Wgt Fields: 2              Sum: Sum
  Spacing: 5     F' offset: 0.00         Transfer: Sigmoid
    Shape: Square                       Output: Direct
    Scale: 1.00   Low Limit: -9999.00   Error Func: standard
   Offset: 0.00   High Limit: 9999.00       Learn: Delta-Rule
 Init Low: -0.100  Init High: 0.100    L/R Schedule: hidden1
 Winner 1: None                      Winner 2: None

L/R Schedule: hidden1

  Recall Step          1        0        0        0        0
    Input Clamp      0.0000   0.0000   0.0000   0.0000   0.0000
    Firing Density 100.0000   0.0000   0.0000   0.0000   0.0000
    Temperature      0.0000   0.0000   0.0000   0.0000   0.0000
    Gain             1.0000   0.0000   0.0000   0.0000   0.0000
    Gain           1.0000   0.0000   0.0000   0.0000   0.0000
  Learn Step         10000    30000    70000   150000   310000
    Coefficient 1    0.3000   0.1500   0.0375   0.0023   0.0000
    Coefficient 2    0.4000   0.2000   0.0500   0.0031   0.0000
    Coefficient 3    0.1000   0.1000   0.1000   0.1000   0.1000
    Temperature      0.0000   0.0000   0.0000   0.0000   0.0000
```

PE: 15

1.000 Err Factor	0.000 Desired	
-1.850 Sum	0.136 Transfer	0.136 Output
14 Weights	0.000 Error	-0.001 Current Error

PE: 16

1.000 Err Factor	0.000 Desired	
1.785 Sum	0.856 Transfer	0.856 Output
14 Weights	0.000 Error	0.010 Current Error

PE: 17

1.000 Err Factor	0.000 Desired	
-0.662 Sum	0.340 Transfer	0.340 Output
14 Weights	0.000 Error	0.005 Current Error

PE: 18

1.000 Err Factor	0.000 Desired	
-1.101 Sum	0.250 Transfer	0.250 Output
14 Weights	0.000 Error	-0.004 Current Error

PE: 19

1.000 Err Factor	0.000 Desired	
-1.470 Sum	0.187 Transfer	0.187 Output
14 Weights	0.000 Error	-0.001 Current Error

PE: 20

1.000 Err Factor	0.000 Desired	
-0.675 Sum	0.337 Transfer	0.337 Output
14 Weights	0.000 Error	-0.006 Current Error

PE: 21

1.000 Err Factor	0.000 Desired	
-0.712 Sum	0.329 Transfer	0.329 Output
14 Weights	0.000 Error	0.002 Current Error

PE: 22

1.000 Err Factor	0.000 Desired	
0.645 Sum	0.656 Transfer	0.656 Output
14 Weights	0.000 Error	-0.011 Current Error

PE: 23

1.000 Err Factor	0.000 Desired	
0.268 Sum	0.567 Transfer	0.567 Output
14 Weights	0.000 Error	-0.004 Current Error

PE: 24

1.000 Err Factor	0.000 Desired	
-0.026 Sum	0.493 Transfer	0.493 Output
14 Weights	0.000 Error	0.009 Current Error

PE: 25
1.000 Err Factor 0.000 Desired
0.236 Sum 0.559 Transfer 0.559 Output
14 Weights 0.000 Error -0.005 Current Error
PE: 26
1.000 Err Factor 0.000 Desired
0.295 Sum 0.573 Transfer 0.573 Output
14 Weights 0.000 Error 0.000 Current Error

Layer: Out

PEs: 2 Wgt Fields: 2 Sum: Sum
Spacing: 5 F' offset: 0.00 Transfer: Sigmoid
Shape: Square Output: Direct
Scale: 1.00 Low Limit: -9999.00 Error Func: standard
Offset: 0.00 High Limit: 9999.00 Learn: Delta-Rule
Init Low: -0.100 Init High: 0.100 L/R Schedule: out
Winner 1: None Winner 2: None

L/R Schedule: out

Recall Step	1	0	0	0	0
Input Clamp	0.0000	0.0000	0.0000	0.0000	0.0000
Firing Density	100.0000	0.0000	0.0000	0.0000	0.0000
Temperature	0.0000	0.0000	0.0000	0.0000	0.0000
Gain	1.0000	0.0000	0.0000	0.0000	0.0000
Gain	1.0000	0.0000	0.0000	0.0000	0.0000
Learn Step	10000	30000	70000	150000	310000
Coefficient 1	0.1500	0.0750	0.0188	0.0012	0.0000
Coefficient 2	0.4000	0.2000	0.0500	0.0031	0.0000
Coefficient 3	0.1000	0.1000	0.1000	0.1000	0.1000
Temperature	0.0000	0.0000	0.0000	0.0000	0.0000

PE: 27
1.000 Err Factor 0.578 Desired
0.259 Sum 0.564 Transfer 0.564 Output
13 Weights 0.578 Error 0.014 Current Error
PE: 28
1.000 Err Factor 0.200 Desired
-0.036 Sum 0.491 Transfer 0.491 Output
13 Weights 0.200 Error -0.291 Current Error

References

1 إبراهيم ندا.(١٩٨٧)"الإستخدام الأمثل لنهر النيل لصالح الاقتصاد المصرى" ، كلية الدفاع مجلد الدورة ١٥ – أكاديمية ناصر العسكرية العليا ، القاهرة.

2.Biswas, A.K. (1991) "land and Water Management for Sustainable Agricultural Development in Egypt: Opportunities and Constraints." Report Submitted to Food and Agricultural Organization. Rome, Italy, Under Project TCP/ EGY/ 0052.

3. Diab , M.S., and Risk, Z.S. (1992) " Management of Ground Water Resources in Egypt." Geology of the Arab World, Cairo University, pp. 147-157.

4. Fausett, L .(1994) "Fundamentals of Neural Network, Architectures, Algorithms, and Applications." Prentice Hall, Englewood Cliffs.

5. NSSAR, S. EL-Mohammady.(1996) " Comparison between Optimum Solutions Using the Inventory Techniques AI Techniques by Building a Mathematical Model for Optimum Utility of Inventory Underground Water in Egypt." Ph. D. Thesis. Faculty of Science, Menoufiya University, Shiben El-Kom, Egypt.

6. Saad, M.M.(1990) " Crises Management and its Applications." M.SC. Thesis, Submitted to Military Technical College, Cairo.

7. Tank, D.W. and J.J. Hopfield. (1986) " Simple Neural Optimization Networks: An A/D Converter, Single Decision Circuit, and a linear Programming Circuit." IEEE. On Circuits and Systems, CAS 33, 5, PP. 533-541.

8. Wasserman, P. D. (1989) "Neural Computing, Theory and Practice." Van Nostrand Reinhold, N.Y.

9. Zahedi, F.(1991) " An Introduction to Neural Networks and a Comparison with Artificial Intelligence and Expert Systems." Interfaces, 21,2, P 25-38.

References (2)

10.Allam, G.I.; Grigg, N.S.; and Ibrahim, H. (1992), "Water Resources of Wadi El- Arish", Water Science-11th Issue, pp. 9-13.

11.Report Submitted to Food and Agricultural Organization", Rome, Italy, Under project TCP/EGY/0052.

12.Burattini, E.: and Tamburrini, G. (1993), " A Neural Knowledge Representation for Diagnostic Expert Systems", Artificial Neural Networks (North Holland), pp. 453-458.

13.Eberlein, S.; and Yates, G. (1991), " Neural Network-Based System for Autonomous Data Analysis and Control", Neural Networks, Volume 1, pp.25.

14.Hertz, J.; Krogh, A.; and Palmer, R. G. (1991), " Introduction to the Theory of Neural Computation", Addison- Wesley Publishing Company California.

15.Heyman, D. P.; and Sobel M.T. (1982), "Stochastic Models in Operations Research", Vol. I, Mc Graw- Hill, New York.

16.Jung, D.; and Burns, J.R. (1993), "Connectionist Approaches to Inexact Reasoning and Learning Systems for Executive and Decision Support", Decision Support Systems", North-Holland, pp. 37-66.

17.Lodewyck, R.W.; and Deng, P. (1993), "Experimentation with a back- Propagation Neural Network", Information & Management 24 North-Holland, pp.-18.

18.Masters, T. (1993), "Practical Neural Recipes in C++", Academic Press, Inc. New York.

19.Neural Ware (1991), "Reference Guide, Neural Ware Inc.", Neural Works Professional II/PLUS and Neural Works Explorer.

20.N Works (1991), " Neural Computing", Neural War. Inc. New York.

21.Onigteit, D.:Kin, C.; and Sehmid, B. (1969), "Operation Design of an Irrigation System", International Commission on Irrigation and Drainage r.6, Mexico City, PP. S. 95-s. 110.

22.Schocken, S.; and Ariav, G. (1991), " Neural Networks for Decision Support: Problems and Opportunities", Working Paper Series STERN IS-91-35, Information Systems Dept. NYU'S Stern School of Business.

23.Taha, I.; and Ghosh, J. (1995), "Controlling Water Reservation Using A Hybrid Intelligent Architecture", Neural Computation Vol.5, pp. 1-11.

24.Tam, K.Y.; and Kiang, M.Y. (1992), " Management Applications of Neural Networks: The Case of Bank Failure Predictions", Management Science, Vol. 38, No. &July, 1992 pp. 926-947.

25.Wang, S.; and Archer, N.P (1994), " A Neural Network Technique in Modeling Multiple Criteria Multiple Person Decision Making", Computers Ops. Res. Vol. 21, No. 2. Pp. 127-142.

26.Wilson, R.; and Sharda, R. (1992), "Neural Networks", OR / MS Today August 1992, pp. 36-42.

إستخدام الشبكات العصبية
فى وضع إستراتيجية للإستخدام الأمثل
للمياه الجوفيه ومياه السد العالى
فى توشكى

ملخص البحث :

تهتم هذه الدراسة بمشكلة الإستخدام الأمثل لمياه النيل والمياه الجوفية التى تعتمد عليها مصر فى تغطية عجز المياه فى الفترة القادمة .

وتقوم الدراسة بتقنين سحب المياه الجوفيه بحيث يكون السحب مناسباً لعملية إعادة ملىء خزانات المياه الجوفية التى يتم تعويضها عن طريق تسرب مياه النيل بالإضافة إلى الأمطار و السيول حتى نتمكن من زراعة مايمكن إستصلاحه من الأراضى الصحراوية بمصر .

وحيث أن جزءاً كبيراً من المياه الجوفية يرجع إلى مياه النيل فقد تم وضع نظام شامل لتوزيع مياه النيل القادمة من بحيرة ناصر توزيعاً مناسباً على مدار السنة حسب الإحتياجات وربطها بمعدلات سحب المياه الجوفية .

والفصل الأول من هذه الدراسة يشرح الوضع القائم لنهر النيل وكذلك المياه الجوفية حيث تبين لنا أن حصة مصر من مياه النيل هى55.5 بليون متر مكعب سنوياً ومخزون المياه الجوفية يتم تعويضه سنوياً بمقدار 4.9 بليون متر مكعب بالإضافة الى الكمية الموجودة أصلاً فى الخزان الأرضى لوارد النيل والدلتا والذى يبلغ حوالى 500 بليون متر مكعب بالإضافة إلى 40,000 بليون متر مكعب تحت الصحراء الغربية .

وفى الفصل الثانى تعرضنا للنماذج الرياضية الإحصائية ونظرية المخزون كأسلوب جرى العرف على إستخدامه فى هذا المجال خاصة فى غياب علوم الذكاء الإصطناعى و الشبكات العصبية موضوع الدراسة فى هذا البحث .

أما فى الفصل الثالث، فعرضت الدراسة رياضيـــات الشبكـــات العصبيـــة و إمكانيـــة إستخدامها فى التوزيع الأمثل لمياه النيل والمياه الجوفية فى مصر .

أما فى الفصل الرابع فيتعوض لإختيار أحد أساليب الشبكات العصبية وهو أسلوب :

Backpropagation Neural Network Model

لدراسة المشكلة محل الدراسة

والباب الخامس (الباب الرئيسى فى الدراسة) فهو تطبيق الأسلوب المختار للشبكـــات العصبيـــة فى وضع إستراتيجية مثلى لتوزيع المياه القادمة من السد العالى والمياه الجوفية فى الأراضى الصحراوية ويعتمد هـــذا الأسلوب على تصميم وبناء الشبكة العصبية و تعليمهـــــــــــــــا و تدريبها ليعطـــى المخرجـــات المطلوبه ، ولقد إستخدم أسلوب إرجاع الخطأ **Backpropagation Algorithm** لتعليم الشبكـــة بتعديل أوزانها الواصله بين عناصرها **Processing Element** وذلك عن طريق الفرق بين المخرجـــات المحسوبـــــــــه **Actual Output** والمخرجات المطلوبه **Desired Output** .

وفى هذا المجال إستخدم مايعرف بجدول القيم العظمى و الصغرى لمدخلات و مخرجـــات الشبكة وتتبع سرعة تعليم الشبكة بإستخدام طريق تعديل معدل التعليم **learning rate** وذلك عن طريق جدول التعليم **learning schedule** .

ولقد إستخدم نموذج مقترح لشبكة ديناميكية وهى تتكون من 13 وحـــدة إدخـــال و2 وحدة إخراج إحداهما تمثل كمية المياه الخارجة من السد العالى لشهر معين فى الســـــنة القادمـــة والأخرى تمثل السحب المخطط للمياه الجوفية لنفس الشهر فى العام القادم .

وفى هذا النموذج يمكن التحكم فى الشبكة بواسطة 2 من المدخلات **inputs** وهما الحد الأعلى و الحد الأدنى لكمية المياه الجوفيه المطلوبه .

ولقد تم بناء الشبكة بواسطة إختيـــــــــــــــار وأختبار **Parameters** للشبكة و أول هذه الـــــ **Parameters** هـــو (**Hidden limits**) .

ولقد قمنا بعدد من المحاولات لتحديد العدد المناسب من هذه المستويات والوحدات وتم تدريب الشبكة بواسطة بيانات معده عن المياه الجوفيه ومياه النيل لسنوات سابقه عددها 23 سنه تم أختبار الشبكة بواسطة بيانات أخرى غير التى تم التدريب عليها وتلك البيانات هى بيانات الخمس سنوات السابقة و بذلك الأسلوب وصلنا إلى الحل الأمثل لتحديد الـ Parameters والـ Hidden layers .

كما قمنا فى بداية التجارب بإستخدام عدد صغير من PE s تم زيادة العدد تدريجياً لتحقيق أفضل النتائج فى حالة دالة Tanh Activation Function بينما فى حالة Sigmoid Activation Function تحققت نتائج جديدة عند عدد 4 وحدات وسطية ولقد تم تغيير Initial Weights لتدريب الشبكة و الوصول إلى أفضل الحلول .

وبنهاية التجارب وصل البحث إلى الشبكة العصبية التى تعطى توافقاً قريباً إلى الواقع لسحب المياه فى المستقبل وتعطى تخطيطاً لسحب المياه الجوفيه حيث يعتمد عليه لتحديد الكميات التى يجب سحبها فى كل شهر على حده لإمكانيه زياده الرقعة الزراعية المستصلحه من الأراضى الصحراوية .

ولكن نظراً لقصور البيانات وعدم توافرها عن منطقة توشكى ، فتم تدريب وتعليم الشبكة على بيانات المياه الجوفية فى الصحراء الغربيه ومياه النيل ومع توافر البيانات عن منطقة توشكى فى المستقبل يمكن تطبيقها على الشبكة بدون تعديل .

سلسلة من القضايا صدر منها:

(١) دراسة الهيكل الاقليمى للعمالة فى القطاع العام فى جمهورية مصر العربية. (ديسمبر ١٩٧٧)

(2) Adverse Economic Effects Resulting From Israeli A ggressions and Continued Occupation of Egyptian Territories, April 1978.

(٣) الدراسات التفصيلية لمقومات التنمية الاقليمية بمنطقة جنوب مصر (ابريل ١٩٧٨)

(٤) دراسة تحليلية لمقومات التنمية الاقليمية بمنطقة جنوب مصر (يوليو ١٩٧٨)

(٥) دراسة اقتصادية فنية لافاق صناعة الاسمدة و التنمية الزراعية فى جمهورية مصر العربية حتى عام ١٩٨٥. (ابريل ١٩٧٨)

(٦) التغذية والغذاء والتنمية الزراعية فى البلاد العربية. (اكتوبر ١٩٧٨)

(٧) تطور التجارة الخارجية وميزان المدفوعات ومشكلة تفاقم العجز الخارجى وسلبيات مواجهتة (٦٩/١٩٧٠-١٩٧٥). (اكتوبر ١٩٧٨)

(8) Improving the position of Third World Countries in the International Cotton Economy, June 1979.

(٩) دراسة تحليلية لتفسير التضخم فى مصر(١٩٧٠-١٩٧٦) (اغسطس ١٩٧٩)

(١٠) حوار حول مصر فى مواجهة القرن الحادى والعشرين. (فبراير ١٩٨٠)

(١١) تطوير اساليب وضع الخطط الخمسية باستخدام نماذج البرمجة

الرياضية فى جمهورية مصر العربية. (مـــارس ١٩٨٠)

(١٢) دراسة تحليلية للنظام الضريبى فى مصر (١٩٧٠/٧١-١٩٧٨) (مـــارس ١٩٨٠)

(١٣) تقييم سياسات التجارة الخارجية والنقد الاجنبى وسبل ترشيدها (يوليـــو ١٩٨٠)

(١٤) التنمية الزراعية فى مصر ماضيها وحاضرها(ثلاثة اجزاء) (يوليـــو ١٩٨٠)

(15) A study on Development of Egyptian National Fleet, June 1980.

(١٦) الانفاق العام والاستقرار الاقتصادى فى مصر ١٩٧٠-١٩٧٩ (ابريـــل ١٩٨١)

(١٧) الابعاد الرئيسية لتطوير وتنمية القرية المصرية. (يونيـــو ١٩٨١)

(١٨) الصناعات الصغيرة والتنمية الصناعية.
(التطبيق على صناعة الغزل والنسج فى مصر). (يوليـــو ١٩٨١)

(١٩) ترشيد الادارة الاقتصادية للتجارة الخارجية والنقدية الاجنبية (ديسمـــبر ١٩٨١)

(٢٠) الصناعات التحويلية فى الاقتصاد المصرى.(ثلاثة اجزاء) (ابريـــل ١٩٨٢)

(٢١) التنمية الزراعية فى مصر (جزئين) (سبتمبـــر ١٩٨٢)

(٢٢) مشاكل انتاج اللحوم والسياسات المقترحة للتغلب عليها. (اكتوبـــر ١٩٨٣)

(٢٣) دور القطاع الخاص فى التنمية. (نوفمبـــر ١٩٨٣)

(٢٤) تطور معدلات الاستهلاك من السلع الغذائية واثارها على السياسات الزراعية فى مصر. (مـــارس ١٩٨٥)

(٢٥) البحيرات الشمالية بين الاستغلال النباتى والاستغلال السمكى (اكتوبـــــر ١٩٨٥)

(٢٦) تقييم لاتفاقية التوسع التجارى والتعاون الاقتصادى بين مصر والهند ويوغوسلافيا (اكتوبـــــر ١٩٨٥)

(٢٧) سياسات وامكانات تخطيط الصادرات من السلع الزراعية (نوفمـــــبر ١٩٨٥)

(٢٨) الافاق المستقبلية فى صناعة الغزل والنسج فى مصر. (نوفمبـــــر ١٩٨٥)

(٢٩) دراسة تمهيدية لاستكشاف افاق الاستثمار الصناعى فى اطار التكامل بين مصر والسودان. (نوفمبـــــر ١٩٨٥)

(٣٠) دراسة تحليلية عن تطور الاستثمار فى ج.م.ع مع الاشارة للطاقة الاستيعابية للاقتصاد القومى. (ديسمبـــــر ١٩٨٥)

(٣١) دور المؤسسات الوطنية فى تنمية الاساليب الفنية للانتاج فى مصر (جزئين). (ديسمبـــــر ١٩٨٥)

(٣٢) حدود وامكانات مساهمة ضريبية على الدخل الزراعى فى مواجهة مشكلة العجز فى الموازنة العامة للدولة واصلاح هيكل توزيع الدخل القومى. (يوليـــــو ١٩٨٦)

(٣٣) التفاوتات الاقليمية للنمو الاقتصادى والاجتماعى وطرق قياسها فى جمهورية مصر العربية. (يوليـــــو ١٩٨٦)

(٣٤) مدى امكانية تحقيق اكتفاء ذاتى من القمح. (يوليـــــو ١٩٨٦)

(35) Intergrated Methodology for Energy Planning in Egypt, Sept. 1986.

(٣٦) الملامح الرئيسية للطلب على تملك الاراضى الزراعية الجديدة

والسياسات المتصلة باستصلاحها واستزراعها. (نوفمبـــــر ١٩٨٦)

(٣٧) دراسة بعنوان مشكلات صناعة الالبان فى مصر (مـــارس ١٩٨٨)

(٣٨) دراسة بعنوان آفاق الاستثمارات العربية ودورها فى خطط التنمية المصرية (مـــــارس ١٩٨٨)

(٣٩) تقدير الايجار الاقتصاد ى للأراضى الزراعية لزراعة المحاصيل الزراعية الحقلية على المستوى الاقليمى لجمهورية مصر العربية عامى ١٩٨٥/٨٠. (مارس ١٩٨٨)

(٤٠) السياسات التسويقية لبعض السلع الزراعية وآثارها الاقتصادية (يونيـــــة ١٩٨٨)

(٤١) بحث الاستزراع السمكى فى مصر ومحددات تنميتة (اكتوبـــــر١٩٨٨)

(٤٢) نظم توزيع الغذاء فى مصر بين الترشيد والألغاء (اكتوبـــــر ١٩٨٨)

(٤٣) دور الصناعات الصغيرة فى التنمية دراسة استطلاعية لدورها فى الاستيعاب العمالى. (اكتوبـــــر ١٩٨٨)

(٤٤) دراسة تحليلية لبعض المؤشرات المالية للقطاع العام الصناعى التابع لوزارة الصناعة. (اكتوبـــــر ١٩٨٨)

(٤٥) الجوانب التكاملية وتحليل القطاع الزراعى فى خطط التنمية الاقتصادية والاجتماعية (فبرايـــــر ١٩٨٩)

(٤٦) امكانيات تطوير الضرائب العقارية لزيادة مساهمتها فى الايرادات العامة للدولة فى مصر. (فبرايـــــر ١٩٨٩)

(٤٧) مدى امكانية تحقيق اكتفاء ذاتى من السكر. (سبتمبـــــر ١٩٨٩)

(٤٨) دراسة تحليلية لاثر السياسات الاقتصادية والمالية والنقدية على تطور وتنمية القطاع الزراعى. (فبرايـــــر ١٩٩٠)

(٤٩) الانتاجية والاجور والاسعار - الوضع الراهن للمعرفة النظرية والتطبيقية مع اشارة خاصة للدراسات السابقة عن مصر. (مـــــارس ١٩٩٠)

(٥٠) المسح الاقتصادى والاجتماعى والعمرانى لمحافظة البحر الاحمر وفرص الاستثمار المتاحة للتنمية. (مـــــارس ١٩٩٠)

(٥١) سياسات اصلاح ميزان المدفوعات المصرى للمرحلة الاولى (مايـــــو ١٩٩٠)

(٥٢) بحث صناعة السكر وامكانيات تصنيع المعدات الراسمالية فى مصر. (سبتمبـــــر ١٩٩٠)

(٥٣) بحث الاعتماد على الذات فى مجال الطاقة من منظور تنموى وتكنولوجى (سبتمبـــــر ١٩٩٠)

(٥٤) التخطيط الاجتماعى والانتاجية. (اكتوبـــــر ١٩٩٠)

(٥٥) مستقبل استصلاح الاراضى فى مصر فى ظل محددات الأرض والمياه والطاقة. (أكتوبـــر ١٩٩٠)

(٥٦) دراسات تطبيقية لبعض قضايا الانتاجية فى الاقتصاد المصرى. (نوفمبـــــر ١٩٩٠)

(٥٧) بنوك التنمية الصناعية فى بعض دول مجلس التعاون العربى. (نوفمبـــــر ١٩٩٠)

(٥٨) بعض آفاق التنسيق الصناعى بين دول مجلس التعاون العربى. (نوفمـــبر ١٩٩٠)

(٥٩) سياسات اصلاح ميزان المدفوعات المصرى(مرحلة ثانية) (نوفمبـــر ١٩٩٠)

(٦٠) بحث اثر تغيرات سعر الصرف على القطاع الزراعى وانعكاساتها الاقتصادية. (ديسمبـــر ١٩٩٠)

(٦١) الامكانيات والافاق المستقبلية للتكامل الاقتصادى بين دول مجلس التعاون العربى فى ضوء هياكل الانتاج والتوزيع. (ينايـــر ١٩٩١)

(٦٢) امكانيات التكامل الزراعى بين مجلس التعاون العربى. (ينايـــر ١٩٩١)

(٦٣) دور الصناديق العربية فى تمويل القطاع الزراعى. (ابريـــل ١٩٩١)

(٦٤) بعض القطاعات الانتاجية والخدمية بمحافظة مطروح(جزئين) الجزء الاول : القطاعات الانتاجية. (اكتوبـــر ١٩٩١)

(٦٤) بعض القطاعات الانتاجية والخدمية بمحافظة مطروح(جزئين) الجزء الثانى: القطاعات الخدمية والبيئية الاساسية. (أكتوبـــر ١٩٩١)

(٦٥) مستقبل انتاج الزيوت فى مصر (اكتوبـــر ١٩٩١)

(٦٦) الانتاجية فى الاقتصاد القومى المصرى وسبل تحسينها- مع التركيزعلى قطاع الصناعة (الجزء الاول) الاسس والدراسات النظرية. (اكتوبـــر ١٩٩١)

(٦٦) الانتاجية فى الاقتصاد القومى المصرى وسبل تحسينها- مع التركيز على قطاع الصناعة (الجزء الثانى) الدراسات التطبيقية. (اكتوبـــر ١٩٩١)

(٦٧) خلفية ومضمون النظريات الاقتصادية الحالية والمتوقعة بشرق اوروبا. ومحددات انعكاساتها الشاملة على مستقبل التنمية فى مصر والعالم العربى. (ديسمبـــــر ١٩٩١)

(٦٨) ميكنة الانشطة والخدمات فى مركز التوثيق والنشر. (ديسمبـــــر ١٩٩١)

(٦٩) ادارة الطاقة فى مصرفى ضوء ازمة الخليج وانعكاساتها دوليا واقليميا ومحليا. (ديسمبـــــر ١٩٩١)

(٧٠) واقع وافاق التنمية فى محافظة الوادى الجديد. (ينايـــــر ١٩٩٢)

(٧١) انعكاسات ازمة الخليج (١٩٩١/٩٠) على الاقتصاد المصرى. (ينايـــــر ١٩٩٢)

(٧٢) الوضع الراهن والمستقبلى لاقتصاديات القطن المصرى. (مايـــــو ١٩٩٢)

(٧٣) خبرات التنمية فى الدول الاسيوية حديثة التصنيع وامكانية الاستفادة منها فى مصر . (يوليـــــو ١٩٩٢)

(٧٤) بعض قضايا تنمية الصادرات الصناعية المصرية. (سبتمبـــــر ١٩٩٢)

(٧٥) تطور مناهج التخطيط وادارة التنمية فى الاقتصاد المصرى فى ضوء المتغيرات الدولية المعاصرة. (سبتمبـــــر ١٩٩٢)

(٧٦) السياسة النقدية فى مصر خلال الثمانيات"المرحلة الاولى" ميكانيكية وفعالية السياسة النقدية فى الجانب المالى والاقتصاد المصرى. (سبتمبـــــر ١٩٩٢)

(٧٧) التحرير الاقتصادى وقطاع الزراعة (سبتمبـــــر ١٩٩٢)

(٧٨) احتياجات المرحلة المقبلة للاقتصاد المصرى ونماذج التخطيط واقتراح بناء نموذج اقتصادى قومى للتخطيط التأشيرى - المرحلة الأولى. (يناير ١٩٩٣)

(٧٩) بعض قضايا التصنيع فى مصر من منظور تنموى تكنولوجى (فبراير ١٩٩٣)

(٨٠) تقويم التعليم الاساسى فى مصر (مايو ١٩٩٣)

(٨١) الآثار المتوقعة لتحرير سوق النقد الأجنبى على بعض مكونات ميزان المدفوعات المصرى (مايو ١٩٩٣)

(82) The Current development in the methodology and applications of operations research obstacles and prospects in develping countries, Nov. 1993.

(٨٣) الآثار البيئية للتنمية الزراعية. (نوفمبر ١٩٩٣)

(٨٤) تقييم البرامج للنهوض بالانتاجية الزراعية. (ديسمبر ١٩٩٣)

(٨٥) اثر قيام السوق الأوربية المشتركة على مصر والمنطقة العربية. (يناير ١٩٩٤)

(٨٦) مشروع انشاءقاعدة بيانات الأنشطة البحثية بمعهد التخطيط القومى "المرحلة الاولى" (يونيو ١٩٩٤)

(٨٧) الكوارث الطبيعية وتخطيط الخدمات فى ج.م.ع (دراسة ميدانية عن زلزال اكتوبر ١٩٩٢ فى مدينة السلام). (سبتمبر ١٩٩٤)

(٨٨) تحرير القطاع الصناعى العام فى مصر فى ظل المتغيرات المحلية والعالمية. (سبتمبر ١٩٩٤)

(٨٩) استشراف بعض الاثار المتوقعة لسياسات الاصلاح الاقتصادى بمصر(مجلدان) (سبتمبـــــر ١٩٩٤)

(٩٠) واقع التعليم الاعدادى وكيفية تطويره (نوفمبـــــر ١٩٩٤)

(٩١) تجربة تشغيل الخريجين بالمشروعات الزراعية وافاق تطويرها. (ديسمبـــــر ١٩٩٤)

(٩٢) دور الدولة فى القطاع الزراعى فى مرحلة التحرير الاقتصادى (ديسمبـــــر ١٩٩٤)

(٩٣) الابعاد الاقتصادية والاجتماعية لتحرير القطاع الصناعى المصرى فى ظل الأصلاح الأقتصادى. (ينايـــــر ١٩٩٥)

(٩٤) مشروع انشاء قاعدة بيانات الانشطة البحتية بمعهد التخطيط القومى (المرحلة الثانية) (فبرايـــــر ١٩٩٥)

(٩٥) السياسات القطاعية فى ظل التكيف الهيكلى (ابريـــــل ١٩٩٥)

(٩٦) الموازنة العامة للدولة فى ضوء سياسة الاصلاح الاقتصادى (يونيـــــة ١٩٩٥)

(٩٧) الستجدات العالمية (الجات واوروبا الموحدة) وتأثيراتها على تدفقات رووس الاموال والعمالة والتجارة السلعية والخدمية (دراسة حالة مصر). (اغسطـس ١٩٩٥)

(٩٨) تقييم البدائل الأجرائية لتوسيع قاعدة الملكية فى قطاع الأعمال العام (ينايـــــر ١٩٩٦)

(٩٩) أثر التكتلات الأقتصادية الدولية على قطاع الزراعة (ينايـــــر ١٩٩٦)

(١٠٠) مشروع أنشاء قاعدة بيانات الأنشطة البحثية بمعهد التخطيط القومى (المرحلة الثالثة) (مايـــــو ١٩٩٦)

(١٠١) دراسة تحليلية مقارنة لواقع القطاعات الأنتاجية والخدمية بمحافظات الحدود. (مايو ١٩٩٦)

(١٠٢) التعليم الثانوى العام فى مصر: واقعة ومشاكله واتجاهات تطويره. (مايو ١٩٩٦)

(١٠٣) التنمية الريفية ومستقبل القرية المصرية:المتطلبات والسياسات (سبتمبر ١٩٩٦)

(١٠٤) دور المناطق الحرة فى تنمية الصادرات (أكتوبر ١٩٩٦)

(١٠٥) تطوير اساليب وقواعد المعلومات فى ادارة الأزمات المهددة لأطراد التنمية (المرحلة الأولى) (نوفمبر ١٩٩٦)

(١٠٦) المنظمات غير الحكومية والتنمية فى مصر (دراسة حالات) (ديسمبر ١٩٩٦)

(١٠٧) الابعاد البيئية المستدامة فى مصر (ديسمبر ١٩٩٦)

(١٠٨) تطوير التعليم العالى فى مصر من اجل التنمية ومواجهة مشكلة البطالة (مارس ١٩٩٧)

(١٠٩) التغيرات الهيكلية فى مؤسسات التمويل الزراعى ومصادر ومستقبل التمويل الزراعى فى مصر (أغسطس ١٩٩٧)

(١١٠) ملامح الصناعة المصرية فى ظل العوامل الرئيسية المؤثرة فى مطلع القرن الحادى والعشرون (ديسمبر ١٩٩٧)

(١١١) آفاق التصنيع وتدعيم الأنشطة غير المزرعية من آجل تنمية ريفية مستدامة فى مصر (فبراير ١٩٩٨)

(١١٢) الزراعية المصرية والسياسة الزراعية فلى اطار نظام السوق الحرة.(فبراير ١٩٩٨)

(١١٣) الزراعة المصرية فى مواجهة القرن الواحد والعشرين (فبراير ١٩٩٨)

(١١٤) التعاون بين الشرق الأوسط وشمال افريقيا (مايو ١٩٩٨)

(١١٥) تطوير اساليب وقواعد المعلومات فى ادارة الازمات (يونيو ١٩٩٨)
المهددة بطرد التنمية (المرحلة الثالثة)

(١١٦) حول اهم التحديات الاجتماعية فى مواجهة القرن ٢١ (يونيه ١٩٩٨)

(١١٧) محددات الطاقة الادخارية فى مصر (يوليو ١٩٩٨)
دراسة نظرية وتطبيقية

(١١٨) تصور حول تطوير نظم المعلومات الزراعية (يوليو ١٩٩٨)

(١١٩) التوقعات المستقبلية لامكانيات (سبتمبر ١٩٩٨)
الاستصلاح والاستزراع بجنوب الوادى

(١٢٠) استراتيجية استغلال البعد الحيزى فى مصر (ديسمبر ١٩٩٨)
فى ظل الاصلاح الاقتصادى

(١٢١) مدخل محاسبى مقترح للقياس الكمى (ديسمبر ١٩٩٨)
للاهمية النسبية لحسابات الاصول والخصوم
فى القطاع الصناعى