NEURAL NETWORKS BASED DATA MINING APPLICATIONS FOR MEDICAL INVENTORY PROBLEMS

Kanti Bansal
Brandeis University
MB 2368; Waltham, MA 02254-9110
Email: kanti@mit.edu

Sanjeev Vadhavkar
Vice President (Technology)
Gupta Consultancy Incorporated
14 Pulsifer Street, Newtonville, MA 02160
Tel: 617-964-3466 Fax: 617-527-5324 Email: vada@mit.edu

Amar Gupta
Co-Director, Productivity from Information Technology (PROFIT)
Sloan School of Management
E53-311 MIT, Cambridge MA 02139
Tel: 617-253-8306 Fax: 617-527-5324 Email: agupta@mit.edu

ABSTRACT

One of the main requirements for agile organizations is the development of information systems for effective linkages with their suppliers, customers, and other channel partners involved in transportation, distribution, warehousing and maintenance. Agility increasingly depends on the quality of decision-making, and companies are continuously trying to improve the quality of decisions by learning from past transactions and decisions. An efficient inventory management system based on contemporary information systems is a first step in this direction. This paper discusses the use of neural networks to optimize the inventory in a large medical distribution organization. The paper defines the inventory patterns and elaborates on the method for constructing and choosing an appropriate neural network to solve the problem. As an extension to the neural network models, statistical procedures and assumptions used to augment the neural network model are explained in detail. With the large number of neural network classes, it is difficult to identify a particular class and model which offers the best inventory model. The paper describes the use of traditional statistical techniques to help determine the best neural network type for a particular application. The paper concludes with a detailed evaluation of the "neural network solution". Using the method proposed in this paper, the total inventory level of the concerned medical distribution organization could be decreased from over a billion dollars to about half-a-billion dollars (reduction by 50 percent).

INTRODUCTION

Organizations are gaining agility by learning to merge products with services and information and to design, build and market products concurrently. For retailers, today's business challenge is to understand the new reality, to leverage that knowledge to become agile, to learn to operate profitably in a competitive environment of continually and unpredictably changing customer opportunities [14]. As we turn the corner into the 21st century, companies are differentiating themselves with new and different products and services. This effort is ultimately successful if they can provide significant and unique value to the customer. Typically customers will appreciate the value of the effort if three expectations are met: price, quality and adaptation to customer need. Understanding the dynamic world of the customer by leveraging information and technology, with the right "customer centered" attitude throughout the company and with cooperation of value chain partners, is a definite first step in the direction of agility [14]. Thus, the key focus for agile retail organizations is the use of technology to give individual customers the sense of buying a unique or personalized product, while retaining most of the cost advantages of mass production [10]. Agility assumes consummate customer service, but it moves beyond placing the customer first, to making him or her central figure in a company-wide dialogue. Agile organizations do not work for or to customers: they work with customers. Organizational agility--the ability to move quickly and decisively and thereby exploit market opportunities--may provide a greater competitive advantage than huge financial resources, talent, or even superior products. An agile organization recognizes that successful strategies are more likely to emerge through a confluence of market conditions than from the efforts of centralized planners and analysts [11].

1 <u>Defining the Problem</u>

With hundreds of chain stores and with revenues of several billion dollars per annum, "Medicorp" is a large retail distribution company that dispenses pharmaceutical drugs to customers in a number of states in the United States. Just as any other retailer in its position, Medicorp is forced to have a large standing inventory of products ready to deliver on customer demand. The problem is how much quantity of each drug should be kept in the inventory at each store and warehouse. Medicorp incurs significant financial costs if it carries excess quantities of drugs relative to the customer demand. Unsatisfied customers frequently turn to competing stores, and Medicorp loses potential profits in such cases. Because of negative experiences, unsatisfied customers may switch company loyalties, relying on other pharmaceutical chains to serve them. On the other hand, Medicorp incurs a financial cost if it carries excessive inventories. Pharmaceutical drugs have a short expiration date and must be renewed periodically. Inventories take a lot of money to maintain. Historically, Medicorp has maintained an inventory of approximately a billion dollars on a continuing basis, using traditional regression models to determine inventory levels for each item.

The best way to manage an inventory is to be able to develop better techniques for predicting customer demands and stock inventories accordingly. In this way, the size of the drug inventory can be optimized to keep up with demand.

2 <u>Preliminary Statistical Analysis</u>

To find the best solution to the inventory problem, we looked at the transactional data warehouse at Medicorp. The Medicorp database is hundreds of gigabytes in size containing all sales information from 1986 to the present. From this vast database, we extracted a portion of the recent fields (Jan. 1995 - Sept 1996) which we felt would provide adequate raw data for a preliminary statistical analysis:

1) Date field – Indicates the date of the drug transaction
2) Customer number – Uniquely identifies a customer (useful in tracking repeat customers)
3) NDC number – Uniquely identifies a drug (equivalent to a drug name)
4) Quantity number – Identifies the amount of the drug purchased
5) Days of Supply -- Identifies how long that particular drug purchased will last

6) Sex field – Identifies the sex of the customer
7) Cost Unit Price – Establishes the per unit cost to Medicorp of the particular drug
8) Sold Unit Price – Identifies per unit cost to the customer of the particular drug

The preliminary statistical analysis was utilized to help search for seasonal trends, correlation between field variables and significance of variables, etc. Our preliminary statistical data provided evidence for the following conclusions:

- Women are more careful about consuming medication than men.

- Most drugs' sales showed no or little correlation to seasonal changes.

- Drug sales are heaviest on Thursdays and Fridays.

- Drug sales (in terms of quantity of drug sold) show differing degrees of variability:
  Maintenance type drugs (for chronic ailments) show low degrees of sales variability.
  Acute type drugs (for temporary ailments) show high degrees of sales variability.

Based on this statistical preliminary analysis, we created and tested a number of types of neural networks. Our initial focus was on acute type drugs (which show a higher variability).

3 Building Neural Network—General Information

For building neural networks, there is no general theory that provides specifications for type of neural network, number of layers, number of nodes (at various layers), or learning algorithm. Without such a theory, researchers have only general rules based on observations by previous network designers. As a consequence, today's network builder must try hundreds, even thousands, of neural networks before s/he hits upon the appropriate neural network to solve her/his problem.

To build these neural networks, we used a freeware product called SNNS version 4.0. Using an advanced feature called *batchman*, this off-the-shelf product enables quick and automated generation of neural networks.

Prior to investing large amounts of time refining certain neural network architectures, we investigated each of the major network classes: Feed-forward or Multi-Layer Perceptron (MLP), Time Delay Neural Network (TDNN), and Recurrent Neural Networks. Most of the major learning algorithms were tried: Hebbian learning (Hebb, 1949), backpropagation momentum learning, time delay network learning, and topographic learning. In each of these categories, we generated a few types of networks to get a preliminary idea of the suitability of the particular approach in addressing the inventory problem.

Investigations into recurrent neural networks, as well as topographic learning, and Hebbian learning were discontinued because of poor initial results. Since MLP and TDNN looked especially promising in forecasting sales demand, we focused on them.

For the time series prediction problem, we had to choose the optimal size of the time interval. Accurate predictions for short time periods (such as daily sales) are more difficult to obtain as compared to those for long intervals of time (such as monthly sales). Short intervals of times require a greater number of forecast points, show greater sales demand variability, and exhibit a lesser dependence on previous sales history. As a result, short-interval predictions can be very difficult. However, short-interval predictions can be more useful than long-interval predictions because they allow one to see sales demand with greater clarity than long-interval predictions. Using MLP architectures and antibiotics sales data, we initially attempted to forecast sales demand on a daily basis. The neural net results were unsatisfactory – the nets produced predictions with very low Pearson Correlation (generally below 20 %) and very high Absolute Error Values (above 80 %). [See section below on statistical measurement of Neural Network Performance]. Such large errors rendered the forecast values useless. Therefore, larger time intervals were tried. Forecasting for a week proved more accurate than for a day; forecasting for a month proved more accurate than for a week; and forecasting for a year proved even more accurate. Indeed, if we had to predict aggregate sales demand for a year, we obtained average absolute errors of only 2 %. In Medicorp's judgment, the best compromise between prediction accuracy and prediction clarity was provided by a weekly prediction interval.

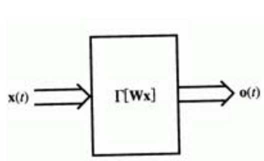
There were two methods by which we presented the neural network historic sales data: the standard method and the rolling method. The difference between these two methods is best shown with an example. Assume that weekly sales data (in units sold) were: 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, etc. In the standard method, we would present the historic data: "10, 20, 30" and ask the network to predict the fourth value: "40". Then, we would present the network with "40, 50, 60" and ask it to predict the next value:" 70". We would continue this process until all training data were exhausted. On the other hand, using the rolling method, we would present historic data as "10, 20, 30" and ask the network to predict the fourth value: "40"; then, we would present the network with "20, 30, 40" and ask it to predict the fifth value: "50". We would continue using the rolling method until all the training data were exhausted.

Either method can be used to produce the training sets. The rolling method has an advantage over the standard method in that the former produces a greater quantity of training examples at the expense of training data quality. Often, the rolling method can confuse the network because of the close similarity between training examples. Using the previous example for instance, the rolling method would produce "20, 30, 40"; "10, 20, 30"; "30, 40, 50". Each of these training examples only differs from each other by a single number only. This minuscule difference may confuse the network and destroy its ability to forecast numbers. On the other hand, with the standard method, one produces a greater quality of training examples over quantity. The differentiation problem is never encountered; training sets, though fewer in number, are adequately different to avoid confusion.

4 <u>MLP Architecture</u>

A single layer (*n* inputs – *m* outputs) MLP model is shown in Figure 1, with *x(t)* denoting the input vector, *o(t)* denoting the output vector, *w(i,j)* denoting the weights and **W** denoting the connection

weight. Held in the relation:



, where x(t) is the input vector and o(t) is the output vector. For more information on the specifics of the MLP neural network, please see [8]. Note that the MLP has neither time-delay elements nor any recurrent input. The class of MLP networks is among the simplest and most powerful type of neural networks.

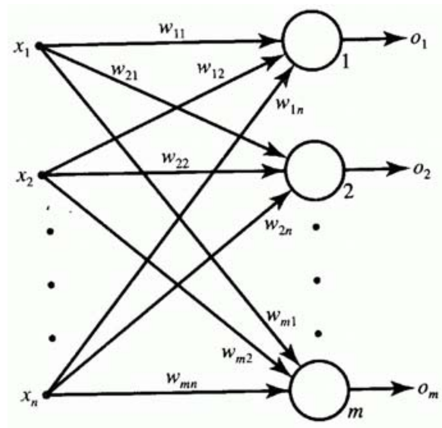


Figure 1: MLP diagram

Two matrices characterize an MLP neural network:

$$\mathbf{W} \triangleq \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1n} \\ w_{21} & w_{22} & \cdots & w_{2n} \\ \vdots & \vdots & \cdots & \vdots \\ w_{m1} & w_{m2} & \cdots & w_{mn} \end{bmatrix}$$

$$\Gamma[\cdot] \triangleq \begin{bmatrix} f(\cdot) & 0 & \cdots & 0 \\ 0 & f(\cdot) & \cdots & 0 \\ \vdots & \vdots & \cdots & \vdots \\ 0 & 0 & \cdots & f(\cdot) \end{bmatrix}$$

The MLP search was constrained to the three-layer model where one layer was the input layer, the second layer was the hidden layer, and the third layer was the output layer. This three-layer constraint enabled developers to limit vast search architecture space while maintaining universal approximation between one finite dimensional space to another.

5 <u>Time Delay Neural Network Architecture</u>

A Time Delay Neural Network (TDNN) model is shown in the figure 2. This type of neural networks is especially adept at handling temporal patterns. In this case, the input data are entered into the network one at a time (look at the x(t) on the left-hand side of the diagram). The network delays handing the data to the next node before the new data value is entered. In this way, the TDNN maintains a history of the time sequence within the network. This history enhances its sales forecasting capabilities. See [9] for more details.

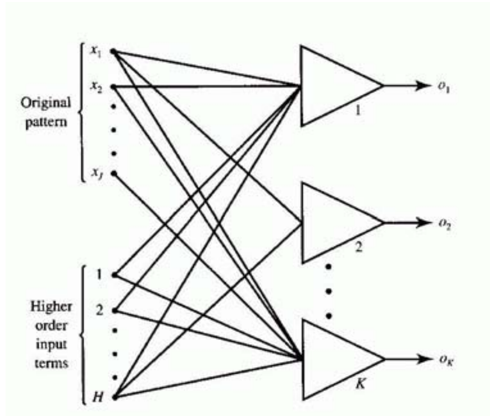


Figure 2: TDNN architecture

We used the "add-and-evaluate" procedure to differentiate useful parameters from those that were irrelevant. First, the neural networks were tried with historical sales data. These networks were statistically evaluated for quality of performance. Then, we entered both historical sales data and an additional parameter (such as sex, days of supply, or customer number) into the network and evaluated the quality of the network's output. If the quality of the network output was significantly better with the additional parameter, we continued to enter the same parameter in the network on all subsequent trials. However, if the network quality was found to be degraded or unchanged, we avoided the use of the particular parameter in further iterations. This procedure enabled the identification of useful network parameters.

To train neural networks, we first generated the neural network patterns. Network patterns are those which allow one to "feed" training data into a network. For all networks tested, we used 1994 and 1995 sales data to train the networks. A neural network needs to be fed all relevant available information pertaining to sales forecasting. However, it is virtually impossible to determine what these parameters are *a priori*. In particular, a network developer must be careful in presenting the correct number of parameters. " . . . An increase in the number of parameters will lessen output errors for given training inputs, but will raise additional errors for inexperienced inputs . . ." [1].

Backpropagation is frequently a very slow method. Indeed, most initial neural network training sessions took up to 3 hours on a Sun SPARC 5 or a SGI INDY machine. In order to speed simulation, a variety of techniques were used. (For more details on speeding backpropagation, please see [3] & [4]).

One particular method used to speed back propagation methods was preprocessing of data. Feed forward networks converge to global minimas faster when data are centered around 1 and -1. Accordingly, all training pattern files contained data that was pre-normalized to values between 1 and -1. [2]

Memory is an important property of neural networks, especially in forecasting neural networks. Only by training with adequate amounts of historical sales data can the neural network be expected to predict future sales demand properly. However, too much past data can confuse a neural network. Therefore, a delicate balance exists between memory and overwhelming. Once again, no sure way exists for knowing *a priori* exactly how much memory to use within a given neural network or how this memory should be presented.

There are two types of memory: implicit and explicit. Implicit memory is stored by the connections within the neural network itself, whereas explicit memory is presented to the network as part of its input. As the issue of memory occupies a central position in the performance of the network, our design team experimented with various neural network memory options for optimum effect. For example, neural network models using Hebbian learning algorithm were tested as Hebbian learning adjusts the network's weights such that its output reflects its familiarity with input. The more probable an input, the larger the output will become (on average).

Time delay neural networks and recurrent neural networks have implicit memory; that is, they can automatically store the data in architectural memory. Memory in recurrent neural nets can be increased/decreased by increasing/decreasing the number of recurrent nodes. In the time delay neural networks, memory is stored in the hidden layers and can be increased/decreased by reducing/increasing size of the hidden nodes. Both 14 and 7 elements of implicit TDNN memory seemed to work optimally for the Medicorp problem.

Data can be presented explicitly. Presenting memory explicitly is especially useful for neural network architectures that are incapable of implicit memory (e.g. the MLP architecture) For instance, one may present the neural network with the past six weeks of sales data and ask the network to forecast the seventh week of sales demand. The size of this memory can be varied considerably depending on the size of the history window. For Medicorp, we utilized history windows as large as 14 weeks, and as small as 0 weeks. The optimal configuration for the MLP was to present seven weeks of historical data.

In a way, the data scarcity problem is directly related to the complexity of the neural net. "*However, when the input space is of high dimension, the number of connections, and thus of free parameters, may become so large that it is impossible to accurately train the network with the available number of training examples: typically, the network will eventually come to correctly learn the training set and perform poorly on the test set*." [2] Therefore, in order to minimize the data scarcity problem, the performance of smaller (less data dependent) nets was analyzed in detail. Those networks that worked especially well to forecast for the Medicorp inventory problem tended to be the smaller nets of less than 20 hidden nodes.

The scarcity of data problem was a major hurdle in building the neural nets. This scarcity of data problem is not new; it has been encountered in many other neural network problems. Because the data scarcity problem is encountered frequently, various solutions, each having their own advantages and disadvantages, have been postulated. In the inventory problem, we combined a number of approaches to tackle the scarcity problem.

One easy way to address the scarcity problem is to utilize one additional year of historical data from the Medicorp database. This led to better neural network solutions (as measured by mean squared error), significantly reduced the risk of over-training the network, and allowed for better problem generalization. However, this solution also involved increased disk space. One year's sales demand data can be as large as 100-200 megabytes. Required format transformations on the data can increase the size to 200-300 additional megabytes. In all, two-year's worth of sales demand data can easily run up to a gigabyte in disk space. Furthermore, the amount of training time for a network goes up linearly as a factor of how much data is presented to it (i.e., O(n) time). Therefore, a network using only one year's worth of data (1995) that previously took 3 hours of time to train, can take up to 6 hours of time to train using two years worth of data (1994 and 1995). In the real world, where time and memory are limited, increasing data is also a limited option.

A number of acute type drugs such as particular types of antibiotics sell infrequently. In fact, some of them may sell only twice or thrice a year at a particular store. This lack of sales data is a major problem in training neural networks. To solve the data scarcity problem, other methods of transformation of data, reuse, and aggregation of data had to be employed. Among the various data transformation schemes discussed in literature the one that is most commonly used for its effectiveness and overall implementation is based on changing future data sets with some known fraction of past data sets. If $X[i]'$ represents the i^{th} changed data set, $X[i]$ represents the i^{th} initial data set, $X[i-1]$ represents the initial $(i-1)^{th}$ initial data set and μ is some numerical factor, then the new time series can be computed as $X[i]' = X[i] + \mu * X[i-1]$, $X[0]' = X[0]$. The modified time series thus has data elements that keep some fraction of past elements. This scheme is particularly useful in the present data scarcity problem for acute type of drugs. By modifying the actual time series with the proposed scheme, the memory of non-zero drug sales is retained for a longer period of time. It is easier to train the neural networks with the modified time series [7].

Another solution to the scarcity problem is to recycle old data. That is, allow the neural network to learn the same training data set many times. In the Medicorp project, we allowed the neural networks to cycle through the data approximately 3000 times. The greatest advantage of recycling is that it allows for more accurate neural network solutions without having to increase the amount of data fed to the neural network (thereby saving space). However, the recycling method has two disadvantages: risk of over-training and overhead time.

Yet another way to deal with the scarcity of data problem is to aggregate different categories of data. In the inventory problem, scarcity of data was tackled by aggregating sales demand within nearby regions. In this method, prediction of sales demand is not done for individual stores, but rather for all stores that combine to form a 'market'. For instance, accurate prediction of sales demand per store is nearly impossible for some drugs. Therefore, we created a 'Boston market' or an aggregation of stores in the Boston area. This market idea had two advantages: it allowed for considerably more accurate prediction of sales (due to the law of large numbers) and it mitigated the scarcity of data problem by adding sales demand data from many different stores.

6 Statistically Evaluating the Best Neural Network Type

In a repertoire of hundreds of neural networks, it is virtually impossible to differentiate and identify which types and classes of neural networks are better than others for predicting consumer drug demand. To objectively identify those types of networks that are best, three different statistical indicators were used. These coefficients are *Pearson Correlation Coefficient (P. Correlation), Normalized Mean Square Error (NMSE),* and *Absolute Error (AE).* Each of the coefficients, in some way, represents a specific method of measuring how well a simulation performed. No one coefficient can tell how well one simulation fared against others; instead all of these three numbers should generally be considered together.

The *Pearson Correlation Coefficient* shows how well trends, i.e., bumps and valleys were picked up. The *Pearson Correlation* is a number ranging between -1 and 1. If our simulation predicts bumps and valleys perfectly, then the corresponding *Pearson Correlation* would be 1.

The *Normalized Mean Square Error (NMSE)* is a method to compare the mean of a series against the predicted values. If the NMSE is greater than 1, then the predictions are doing worse than the series mean. If the NMSE is less than 1, then the forecasts are doing better than the series mean. The NMSE is a widely used measure in academic journals to evaluate how well a Neural Network has performed.

$$NMSE = \frac{\sum_{n} \frac{(x_{predict} - x_{actual})}{(x_{actual} - \bar{x})}}{n}$$ x = sales data points, n = number of sales data points

The *Absolute Error (AE)* is another way to compare the forecasts with the actual values. It indicates, as a percentage value, the average difference between the predicted and actual value. For instance, an AE of 0.40 means that the neural network will provide predictions which, on the average, are within plus or minus 40% of the actual values. Unfortunately, the AE tells us nothing of how well the computer predicts trends; for that we must use *Pearson Correlation* factor.

$$AE = \frac{\sum_{n} \left| x_{output} - x_{predict} \right|}{n}$$ x = sales data points, n = number of sales data points

Using the above statistical measures, we established that the Multi-Layer Perceptrons architecture provided the best results in the case of Acute Type drugs.

After modeling consumer demand for the acute class of drugs, we conducted analysis on the data for the maintenance class of drugs. In the latter case, we found that the best neural network configuration was the TDNN (7, 14). However the TDNN (7, 14) performed only marginally better than the MLP(1) network (MLP with 14 input nodes, 1 hidden node and 1 output node); therefore, we concluded that the MLP(1) neural network was adequate in predicting sales demand for both acute and maintenance classes of drugs.

7 Interpreting Results

How well did the neural networks perform in predicting customer demand? Also, how effective are the neural networks for minimizing drug inventory levels?

Medicorp is governed by two competing principles: minimize drug inventories and enhance customer satisfaction. As a matter of policy, Medicorp wants to ensure that at least 95% of its incoming customers are happy. That is, Medicorp wishes to fill at least 95% of the prescriptions that it receives. This provides us with a statistical guideline of "customer satisfaction".

One way to compare various inventory models is to stimulate the models to see how they would perform in the real world. For the flat sales model, we used the 1995 data to develop parameters for the model and used the 1996 data to test the predictive performance of the model. For the MLP type neural network, we used the 1994 data to build network parameters and tested them against the 1996 data. The reason that we used two different years to build parameters for the different models (1994 and 1995) was purely due to convenience. The neural network was more easily built with the 1994 data available in neural network pattern form; the flat sales parameter statistics were more easily built with the 1995 data available in spreadsheet form. This difference of years should not bias any of the test statistics toward any particular model because the statistics were "year insensitive"; that is, they did not depend on the year of the sample data. Furthermore, there was no significant statistical difference between the two sets of sample data.

To compare various inventory models, we must have common unbiased statistics to measure how well each model fares. First, as a measure of customer dissatisfaction, the notion of "undershoots" is introduced. The number of undershoots tells how many times a customer would be turned away for lack of drug stock had Medicorp implemented that particular inventory model. The fewer "undershoots", the better a given model fares.

Second and more importantly, the notion of "days-of-supply" is introduced. The "days-of-supply" parameter represents the number of days the drug supply in the inventory is expected to last. The "days-of-supply" indicator is similar to the raw amount of the drug in one way: the greater the 'days-of-supply' the greater the amount of the drug in the inventory. However, "days-of-supply" differs from the raw amount of drug in two important ways. First, the "days-of-supply" parameter allows for common comparisons between different types of drugs. Different drug amounts are measured in different ways: liquid drugs are measured in milliliters (ml); solid drugs are measured in milligrams (mg); and capsules are measured in strength and number. If one talked in terms of raw amount, one would have to take into account different units of measure, namely, ml, mg, number of capsules, and drug strength. "Days-of-supply" allows us to talk about different amounts in terms of one unit: days.

Another important difference between "days-of-supply" and raw amount is that it allows us to intelligently compare models. For instance, if one were to use raw amount, instead of "days-of-supply", one would not know how to judge a given inventory model. If a given model indicated that one should store large amounts of a particular drug X, it could be because X is a popular drug, therefore frequently ordered by customers or because "we have a bad inventory model". If instead, one used the "days-of-supply" indicator, such a difference would be clear. The popularity/unpopularity of a drug would be factored in with the "days-of-supply" indicator.

These two factors, "undershoots" and "days-of-supply", are important in judging inventory models. Each one is a statistically unbiased measure of competing goals: to keep customers happy and to minimize drug stock.

Medicorp has traditionally stored three weeks of supply of each pharmaceutical drug regardless of the drug type. In comparison with two other competing models, this system seems simplistic; it neither insures customer happiness and loyalty nor does it minimize inventory stockpiles. The only visible benefit to this system is its ease of management. The "three week rule" is easy to remember and administer.

The flat sales model, a simple statistical model and more complex model than Medicorp's original policy, maintains customer satisfaction while attempting to minimize drug inventory. The flat sales model, using 1994 sales data as input, and 1995-96 sales data as output, assumes that the customer demand is a normal distribution from January 1995-September 1996. Assuming that last year's sales for a particular drug are very much like this year's sales, the flat sales model reveals the amount of drug (in days of supply) needed in the inventory to attain 95% satisfaction level.

The flat sale value is computed like this: We assume that 1994 year's sales distribution forms a normal curve. Then, the sales normal curve (see figure 3) is transformed into a standard normal curve (using basic statistical equations), and the demand quantity value at z=1.65 is computed. This flat sales value is the amount of inventory that would have been needed to keep 95% of the customers in 1994 happy. Because we assume that 1995 - 1996 sales curve will look similar to sales picture for 1994, we also assume that this computed quantity is also what is needed to keep 95% of the customers in 1995 – 1996 happy.

As one can see, the flat sales model relies on two major assumptions: namely, 1995 – 1996 sales will mimic 1994 sales (or more generally, that the next year's sales will be much like this year's sales), and that sales curves are always normal.

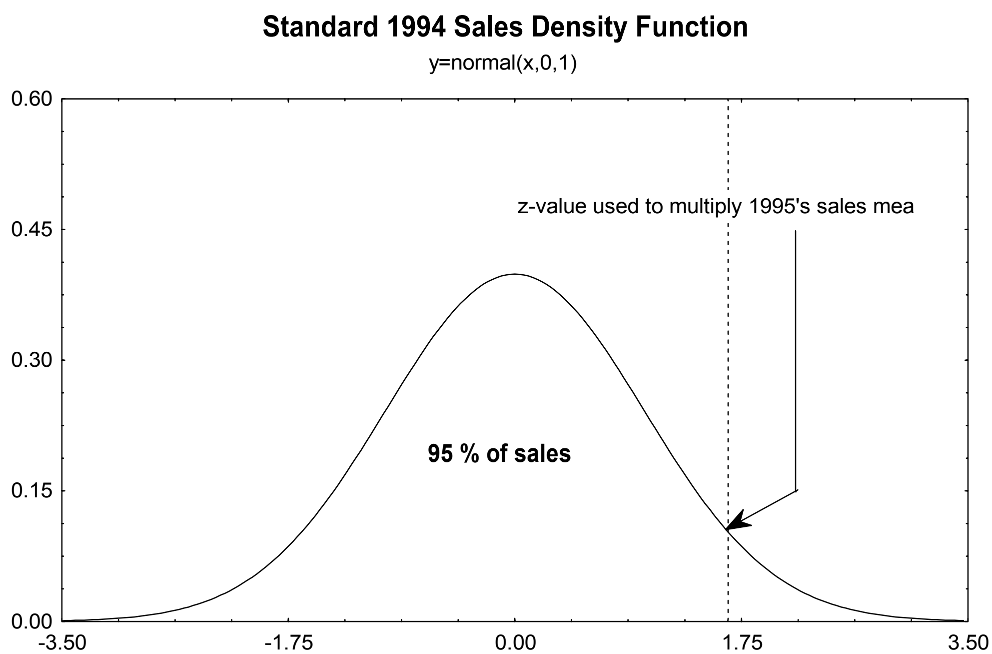


Figure 3: Sales Density Function

For most established drugs, the normality assumption for this flat sales model is reasonable. Established drugs have, for the most part, saturated the market. Their sales pattern mimics the distribution of "white noise" (it resembles a bell curve). Evidence for this assumption also comes from the Shapiro-Wilks' W test. The Shapiro-Wilks' W test, put forward by Shapiro, Wilk, & Chen in 1968, is the preferred test of normality because of its good power properties as compared to a wide range of alternative tests, namely the Kolmogorov-Smirnov test and the Lilliefors test. As one can see in Figure 4, the W value is close to one, which indicates that the distribution is close to normal. Most fast-moving drugs tested under the Kolmogorov-Smirnov test had values close to one, indicating that they had distributions close to normal. While the normality assumption and the equivalence assumption (that next year's sales will be much like this year's sales) hold for established drugs, they fail for both newly introduced and discontinued drugs. Newly introduced drugs have an upward trend in sales since their introduction to the market. Discontinued drugs have a downward sales trend since production ended. In both these cases, the normality assumption may hold using detrending techniques such as differencing. Differencing is often used to eliminate non-stationarity realizations of data.

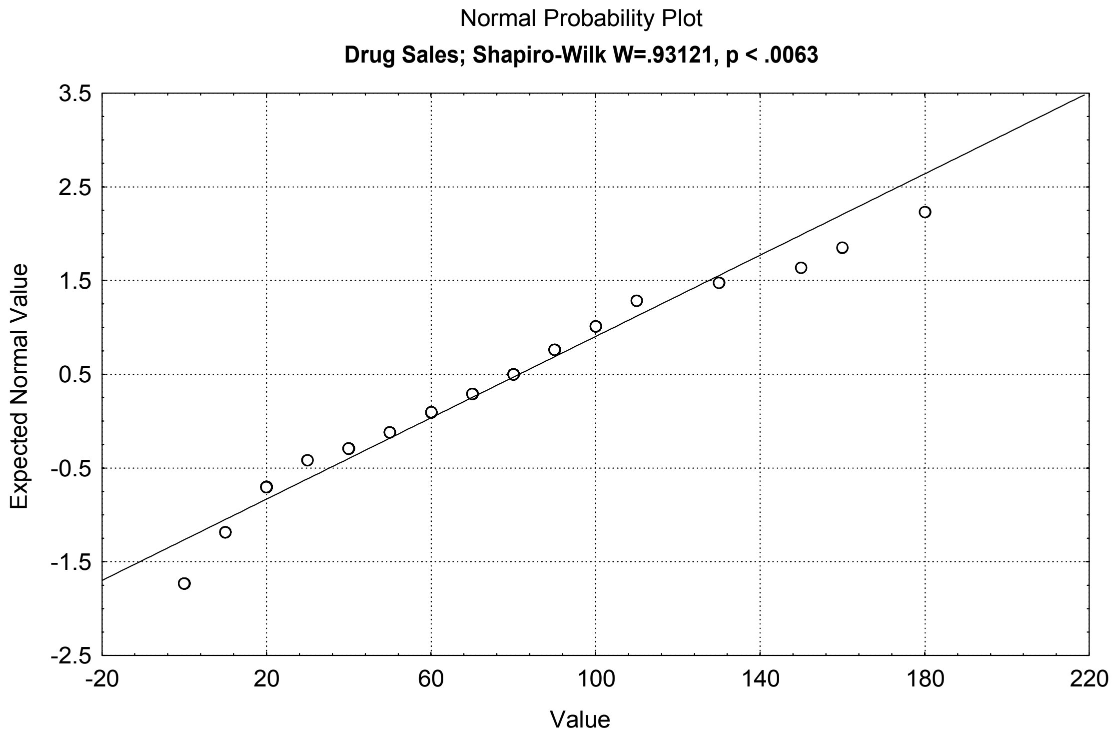


Figure 4: Shapiro - Wilk's Test on a Sample Fast Moving Drug Sales Data

Differencing allows one to eliminate polynomial trends as well as seasonality [6]. While detrending tools may remedy the normality assumption, the equivalence assumption may not hold. Therefore, one disadvantage to the flat model is its inability to deal effectively with and predict inventory stock for newly introduced drugs, or for discontinued drugs.

For slow-moving or established drugs, the MLP model is considerably better than the Flat model (for example, in Table 1, file numbers #78, #82, #1235). While maintaining a 95% probability of customer satisfaction (that is, Medicorp is able to fill the prescription), the MLP model reduces days-of-supply for established drugs in the inventory by 66%. Since established drugs constitute the majority of the drugs in the total drug inventory, a reduction in the days-of-supply offers a major benefit. The neural network model seems to work best in terms of undershoots and days-of-supply. On the average, the neural network undershoots only three times (keeping the 95% customer satisfaction policy of Medicorp).

8 Conclusions

The deployment of neural network-based inventory management systems will encourage organizations to incorporate agile business practices and enabling technologies to produce arbitrary quantities of customizable, reconfigurable, and upgradeable products, supported over the entire product life cycle. A neural network-based inventory management system can enable organizations to establish effective linkages with their partner organizations in the supply chain process.

			Flat Sales		MLP Single w/ 94-95 data	
File #	Average	Standard Deviation	Undershoots	Days of Supply	Undershoots	Days of Supply
78	68	99	1	84	1	25
79	134	145	2	28	2	26
80	1224	729	5	18	10	20
81	9	37	0	12	0	7
82	138	170	0	80	2	24
360	582	133	3	19	5	14
441	487	233	0	32	3	13
446	398	152	1	26	5	14
1118	520	182	1	21	4	15
1119	381	136	0	21	4	13
1120	381	136	0	21	4	11
1121	381	136	0	21	2	22
1122	158	79	1	21	7	15
1234	137	133	1	60	1	19
1235	17	51	1	186	1	41
1236	1318	928	3	23	3	16
1237	115	113	2	37	1	38
1238	53	107	3	31	1	23
1255	11831	7136	2	21	8	19

Table 1: Comparison of results for acute type drug #ABCDEF

The neural network based inventory management system described in this paper emphasizes the following:

1. Comprehensive decision support capability, enabling the end-user to interactively explore a number of tradeoffs using data mining techniques,
2. Concurrent development and dynamic revision of integrated process planning and inventory management solutions,
3. Use of a common representation for exchanging drug inventory information, and
4. Coordination with outside information sources such as drug manufacturers, warehouses, suppliers and end customers.

Inventory control is a nascent application of neural networks. After studying the constraints that characterize the medical arena, an autonomous inventory management system has been created based on an ultra-sparse single layer neural network. By deploying this neural network based model, the inventory at Medicorp consisting of over a billion dollars worth of drugs can be reduced by 50 % to about one-half billion dollars while maintaining an equivalent customer satisfaction level (95% of prescriptions are filled).

Acknowledgements

We thank Aparna Agrawal for her help in building training data for the neural network. We also thank Gerardo J. Lemus Rodriguez and M. V. Nagendra Prasad for their helpful ideas and discussions.

References

1. Noboru Murata et al., Artificial Neural Networks, vol. 1, pp 9-14, 1991.
2. Francois Soulie, Artificial Neural Networks, vol. 1, pp 605-615, 1991.
3. Gallman P. Botton, A framework for the cooperation of learning algorithms. NIPS' 90, Denver, Morgan Kauffman Publishers, 1991.
4. S. Becker, "Improving the convergence of back propagation learning with second-order methods", Proceedings of the 1988 Connectionist Summer School, pp 29-37, Morgan Kaufmann Publishers, 1988
5. D.O. Hebb, The Organization of Behavior, Wiley, 1949.
6. J.D. Hamilton, Time Series Analysis, Princeton University Press, 1994.
7. M.C. Mozer, "Neural Net Architectures for Temporal Sequence Processing", Predicting the future and understanding the past (Eds. A. Weigend and N. Gershenfeld), Addison-Wesley, 1993.
8. D.R. Hush and G.B. Horne, "Progress in supervised neural networks: What's new since Lippmann?," IEEE Signal Processing Magazine, pp 8-38, 1993.
9. A. Waibel, "Modular Construction of Time-Delay Neural Networks for Speech Recognition", Neural Computation Vol: 1, pp 39-46, 1989.
10. *The National Information Infrastructure: Agenda for Action*, Information Infrastructure Task Force, September 15, 1993.
11. R. Nagel and R. Dove, *21st Century Manufacturing Enterprise Strategy: An Industry Led View of Agile Manufacturing*, Iococca Institute, Lehigh University, Volumes I and II, 1991.
12. P. Adriaans and D Zantinge, Data Mining, Quantum Books, 1996.
13. U. Fayyad, G. Piatetsky-Shapiro, P. Smyth and U. Uthurusamy, Advances in Knowledge Discovery and Data Mining, AAAI Press/MIT Press, 1996.
14. L. Goldman, R.L. Nagel and K Preiss, Agile Competitors and Virtual Organizations - Strategies for Enriching the Customer, Van Nostrand Reinhold, 1995.