Open Journal of Optimization, 2014, 3, 13-18
Published Online June 2014 in SciRes. http://www.scirp.org/journal/ojop
http://dx.doi.org/10.4236/ojop.2014.32002

Scientific Research

The Neural Network That Can Find the Maximum Income of Refinery

Bahman Mashood[1], Greg Milbank[2]

[1]1250 La Playa Street. 304. San Francisco. CA 94109. USA
[2]Praxis Group. 1618 Northfield Road. Nanaimo. BC. V9S 3A9. Canada
Email: b_mashood@hotmail.com, Gregm@Praxistech.com

Received 22 March 2014; revised 2 May 2014; accepted 4 June 2014

Abstract

In this article we are going to introduce the neural network approach to approximate the solution for optimization problems. Using this approach we are able to approximate the optimum values for the large class of functions in particular giving the prices of different products that are resulted from refining the crude petroleum into different substances. We are going to design a neural network that can provide us with a decomposition of the given crude petroleum into resulted products in such a way that is most beneficial for the refinery.

Keywords

Neural Network, Optimization

1. Introduction

Many problems in the industry involved optimization of certain complicated function of several variables. Furthermore there are usually set of constrains to be satisfied. The complexity of the function and the given constrains make it almost impossible to use deterministic methods to solve the given optimization problem. Most often we have to approximate the solutions. The approximating methods are usually very diverse and particular for each case. Recent advances in theory of neural network are providing us with completely new approch. This approach is more comprehensive and can be applied to a wide range of problems at the same time. In the preliminary section we are going to introduce the neural network methods that are based on the works of D. Hopfield, Cohen and Grossberg. One can see these results at (Section-4) [1] and (Section-14) [2]. We are going to use the above methods to find the maximum of the refinery under certain assumptions. Our calculations are based on the system of neural networks which is combined of four different neural networks which will be utilized. It will provide us with the desired results that will be included in final section. The results in this article

How to cite this paper: Mashood, B. and Milbank, G. (2014) The Neural Network That Can Find the Maximum Income of Refinery. *Open Journal of Optimization*, **3**, 13-18. http://dx.doi.org/10.4236/ojop.2014.32002

are based on our common work with Greg Milbank of praxis group. Many of our products used neural network of some sort. Our experiences show that by choosing appropriate initial data and weights we are able to approximate the stability points very fast and efficiently. In Section-3 we introduce the extension of Cohen and Grossberg theorem to larger class of differential equations. The appearance of new generation of super computers will give neural network much more vital role in the industry, machine intelligent and robotics. The references [1]-[4] can help the readers to get comprehensive ideas about neural networks, linear programming and matrices.

1.1. On the Structure and Application of Neural Networks

Neural networks are based on associative memory. We give a content to neural network and we get an address or identification back. Most of the classic neural networks have input nodes and output nodes. In other words every neural networks is associated with two integers m and n. Where the inputs are vectors in R^n and outputs are vectors in R^m. neural networks can also consist of deterministic process like linear programming. They can consist of complicated combination of other neural networks. There are two kind of neural networks. Neural networks with learning abilities and neural networks without learning abilities. The simplest neural networks with learning abilities are perceptrons. A given perceptron with input vectors in R^n and output vectors in R^m, is associated with treshhold vector $(\theta_1,\theta_2,\cdots,\theta_m)$ and $m\times n$ matrix $(w_{i,j})$. The matrix W is called matrix of synaptical values. It plays an important role as we will see. The relation between output vector $O=(o_1,o_2,\cdots,o_m)$ and input vector vector $S=(s_1,s_2,\cdots,s_n)$ is given by $o_i=g\left(\sum_{k=1}^{k=n}w_{i,k}s_k-\theta_i\right)$, with g a logistic function usually given as $g(x)=\tanh(\beta x)$ with $1>\beta>0$ This neural network is trained using enough number of corresponding patterns until synaptical values stabilized. Then the perceptron is able to identify the unknown patterns in term of the patterns that have been used to train the neural network. For more details about this subject see for example (Section-5) [1]. The neural network called back propagation is an extended version of simple perceptron. It has similar structure as simple perceptron. But it has one or more layers of neurons called hidden layers. It has very powerful ability to recognize unknown patterns and has more learning capacities. The only problem with this neural network is that the synaptical values do not always converge. There are more advanced versions of back propagation neural network called recurrent neural network and temporal neural network. They have more diverse architect and can perform time series, games, forecasting and travelling salesman problem. For more information on this topic see (Section-6) [1]. Neural networks without learning mechanism are often used for optimizations. The results of D. Hopfield, Cohen and Grossberg, see (Section-14) [2] and (Section-4) [1], on special kind of differential equations provide us with neural networks that can solve optimization problems. The input and out put to this neural networks are vectors in R^m for some integer m. The input vector will be chosen randomly . The action of neural network on some vector $X_1\in R^m$ consist of inductive applications of some function $f;R^m\to R^m$ which provide us with infinite sequence $X_1,X_2,\cdots,X_n,\cdots$. where $X_n=f(X_{n-1})=f^{n-1}(X_1)$. And output (if exist) will be the limit of of the above sequence of vectors. These neural networks are resulted from digitizing the corresponding differential equation and as it is has been proven that the limiting point of the above sequence of vector coincide with the limiting point of the trajectory passing by X_1. Recent advances in theory of neural networks provide us with robots and comprehensive approach that can be applied to wide range of problems. At this end we can indicate some of the main differences between neural network and conventional algorithm. The back propagation neural networks, given the input will provide us the out put in no time. But the conventional algorithm has to do the same job over and over again. On the other hand in reality the algorithms driving the neural networks are quite massy and are never bug free. This means that the system can crash once given a new data. Hence the conventional methods will usually produce more precise outputs because they repeat the same process on the new data. Another defect of the neural networks is the fact that they are based on gradient descend method, but this method is slow at the time and often converge to the wrong vector. Recently other method called Kalman filter (see (Section-15.9) [2]) which is more reliable and faster been suggested to replace the gradient descend method.

1.2. On the Nature of Crude oil and Its Decomposition

Crude oil is naturally occurring brown to black and is flammable liquid. It is principally found in oil reservoirs. Regardless of their origin all crude oils are mainly constituted of hydrocarbons mixed with variable amounts of

sulfur nitrogen and oxygen compounds. The ratio of the different constituents in crude oil how ever vary appreciably from one reservoir to another. Crude oil are refined to separate the mixture into simpler fractions that can be used as a fuel, lubricates, or as intermediate stuck to petrochemical industries. The hydrocarbons in crude oil are mostly paraffines naphthenes and various aromatic hydrocarbons. The relative percentage of the hydrocarbons that appear in crude oil vary from oil to oil . In the average it is consistent of 30/100 paraffines, 40/100 naphthalene, 15/100 aromatic and 6/100 asphalt. The most marketable components of of petroleum are, natural gas, gasoline, benzine, diesel fuel, light heating oils, heavy heating oils and tars. The hydrocarbon components are separated from each other by various refinery process. In the process called fractional dissolution petroleum is heated and into a tower. The vapor of different components condense on collector at different heights in the tower. The separated fractions are then drown from the collectors and further processed into various petroleum products. As the light fractions specially gasoline are in high demand so called cracking procession have been developed in which heat and certain catalyst are used to break up the large molecules of heavy hydrocarbons into smaller molecules of lighter hydrocarbons. Some of the heavier fractions find eventual use as lubricating oil, paraffins, and medical substances. We can summarize the above decomposition methods in the following:

1) Fractional desolation: In this method which is used at the first stage, we use different levels of heat and pressure to desolate different products .

2) Chemical processing: In this method the given products are processed using chemical processing as in the the following. i) Each product can break down into smaller hydrocarbons. ii) Couple of smaller hydrocarbons will be combined to produce heavier hydrocarbons. Now given a market price for a Gallon of crude oil and expenses involved in producing a Gallon of each of the products we are going to introduce a method that can calculate the most beneficial decomposition of a given crude oil into resulted products. At this point we have to mention that the actual process of refining the crude oil might be much more complicated than a simplified version we use here. Our methods are based on some simple but sensible assumption about the process refinery. The a system of neural networks which is a combination of four neural networks will provide us with desired results. These neural networks are also able to extract some vital technical information about the process of refinery just by considering the given basic data.

1.3. Preliminary Model of the Problem in Finding the Maximum Income of Refinery

In this Section we are going to set some assumption regarding to the process of refining the crude oil and its decomposition. These assumption as we mentioned before are some how simplistic and will lead to the system of linear programming that will provide us with optimal solution. Since in reality the equations and constrains can be more complicated we introduce the algorithm based on theory of neural network. This algorithm can estimate the optimum for the cases where the functions and constrains are not linear. Suppose we are given a function $P(u_1,u_2,\cdots,u_n)$ of n variables $u_1,u_2,\cdots,u_n$ to be optimized. Furthermore for $i=1,2,\cdots,n$, let $(g_i(u_i))$, to be the set of C^2 function. Assume we want to optimize the function P, given we have to satisfy the following constrains $\phi=\sum_{i,j}c_{i,j}g_i(u_i)g_j(u_j)+\sum_k g_k(u_k)+c=0$, where $(c_{i,j})$ $s,(d_k)'s$ and c are constants. Following the arguments at (Section-4) [1] and (Section-14.9) [2] we define energy function $E=P+\gamma\phi$. Let us assume that P can be expressed as $P=\sum_i P_i(u_i), i=1,2,\cdots,n$, where each P_i is a C^1 function. Thus we have $E=\sum_i P_i(u_i)+\gamma\phi$. Suppose $c_{i,j}=c_{j,i}$. and $(g_i(u_i))'\geq 0$ Now for $j=1,2,\cdots,n$ consider the following set of n differential equations. $\mathrm{d}u_j/\mathrm{d}t=b_j(u_j)+\sum_i c_{j,i}\varphi_i(u_i)$. Using the results of (Section 14.9) [2], for each integer $j=1,2,\cdots,n$ we get $\varphi_j(u_j)=(2\gamma)^{0.5}g_j(u_j)$ and $P_j(u_j)+\gamma d_j g_j(u_j)+\gamma c/n=\oint_0^{u_j} b_j\varphi_j'(\lambda)\mathrm{d}\lambda$. Therefore we have, $b_j(u_j)=-\left(P_j'(u_j)+\gamma g_j'(u_j)d_j\right)/\varphi'(u_j)$. Finally the results of (Section-14-9) [2], implies that as a function of time E, is a decreasing function which guaranties that the above neural network will converges to optimum of E as the vector $U=(u_1,u_2,\cdots,u_n)$ converge to the vector $U^*=(u_j^*), j=1,2,\cdots,n$, as time goes to infinity. Let us by NLP the set of all optimization systems that we can find their optimum using the above process.suppose Given a polynomial Q in $u_1,u_2,\cdots,u_n$. Then using routine arguments in theory of functions of several variables we know that R^n can be divided into the union of finite disjoint open sets. On each of this open sets the value of the function has the same sign. Next we define the function f acting on R^n with f, equal to the absolute value of Q. Thus f becomes a positive function on the domain R^n of Q. Now given a system P of differential equation

$(du_i)/dt = P_i$, with P_i polynomial,suppose we can write that $(du_i)/dt = (u_i - f_i(u_j, j \neq i))\, g_i(u_j)$. Where all the functions in the i'th expression are polynomials of less degree than P_i. Now let us define a simple energy function $G = \sum_j (f_j(u_k, k \neq j))^2 - \sum_j (u_j)^2$. By small variations around the boundary of G we can assume that G is C^1 function. Finally let us define the energy function G as in the following. There exist a sequence of disjoint open sets (V_i) and a sequence of numbers (υ_i) with υ_i takes one of the values 1 or −1. Then by making an appropriate choice for the above sets and integers E can be defined on each of the sets V_i to be equal to $\upsilon_i G$, such that the value of E, will be strictly positive except at the critical point on which it vanishes. Furthermore $dE/dt < 0.$ This implies that E is a Lyapunov function for the system P and the convergence to a critical point is asymptotically stable. Unfortunately the above arguments the existence of the critical points for the system.

Theorem Suppose for each i, P_i is a polynomial in term of its variables $(u_i)_i$, satisfying the above conditions. Then P is in NLP.

Let us pick up small compact subset $O \subset R^n$ containing the the equilibrium point u_0. Furthermore suppose the set of functions P_i's are analytic functions. Then for any each ϵ small enough there exists an energy function $E(\epsilon)$ where $dE/dt < 0$, as long as E acts on O minus the set of all elements that are ϵ close to u_0. This implies that if the set of functions involving the system P are analytic and if we can guess the region in which u_0 is located then practically we can assume that P belong to NLP. This in fact need some extra initial work and assumptions about the polynomials involved in the above process of approximating the functions P_i in the above differential equations. The problem with the above methods is how to pick the initial conditions in order not to end up in the local minima or local maxima. This will take some dedication. The experience shows that the best alternative is to choose the initial vector U_{in} randomly and choose γ to be of the same size as the average of u_j's. Equally this can be done using generalized Hebb rule as given by the Formula (2.9) [1]. Now we introduce the set of neural networks that will approximate the solution to the refinery problem.

To begin with we assume the weight of the given crude oil is one gallon. Let us assume that $p_1, p_2, \cdots, p_n$, be the set of all hydrocarbons that can be extracted from crude oil. Let $q_1, q_2, \cdots, q_n$, be the corresponding percentage of them in one gallon of the given crude oil. During the refining process the product p_i, will contribute $q_{i,1}$ gallon of its weight to produce other substances . Conversely it receives $q_{i,2}$ gallons as a result of chemical process between other products. The final amount of product p_i, at the end of refining process Q_i, will be given in the following

$$Q_i = q_i - q_{i,1} + q_{i,2} \tag{1}$$

it makes sense to assume that for a given products p_i and p_j, there exist coefficients $c_{i,j}$ and $e_{i,j}$, such that the amount of contribution of product p_i to product p_j is equal to $c_{i,j}$ and the cost of this transformation is equal is equal to $w_{i,j} = c_{i,j} \cdot e_{i,j}$.

$$q_{i,2} = \Sigma_k q_{k,l} c_{k,i} \tag{2}$$

$$q_{i,1} = \Sigma_k q_{k,2} c_{i,k} \tag{3}$$

$$1 = \Sigma_j c_{i,j} \tag{4}$$

$$1 = \Sigma_i c_{i,j} \tag{5}$$

Suppose the price of one gallon of product p_i is Pr_i. This implies that the total profit P of the refinery is equal to the sum of the market price of resulted products minus the total expenses, *i.e.*,

$$P = \Sigma_i Q_i Pr_i - \Sigma_{i,j} q_{i,1} w_{i,j} = \Sigma_i (q_i - q_{i,1} + q_{i,2}) - \Sigma_{i,j} q_{i,1} w_{i,j} \tag{6}$$

And we wish to maximize P. Furthermore we have the following inequalities,

$$q_{i,1} \geq 0 \tag{7}$$

$$q_{i,2} \geq 0 \tag{8}$$

$$q_i \geq q_{i,1} \tag{9}$$

Assuming the values $q_i, c_{i,j}, e_{i,j}, i, j = 1, 2, \cdots, n$ are known, the above system is the system of linear programming in term of $2n$ variables $q_{i,1}, q_{i,2}, i = 1, 2, \cdots, n$ and its solution will provide us with the maximal benefit. On the hand we might seek a brand of crude oil that can maximize the profit P. In this case we use the fact that the maximum amount of product p_i that can be inside the gallon of crude oil is given to be less or equal than a given number M_i. In this case we have to add another set of n variables to the above system of linear programming and the following constrains:

$$\Sigma_i q_i = 1 \tag{10}$$

$$0 \le q_i \le M_i \tag{11}$$

And the solution of this new system of linear progamming will identify the brand of crude oil that will bring maximum benefit. The problem is to find the coefficients $c_{i,j}$ and $w_{i,j}$. In the proceeding Sections we will find ways to get away with this problem and also simplify the above system of linear programming in order to bring down the number of constrains.

2. Connection to Perron Frobenius Theory

Let $A = \left(a_{i,j}\right)$ be $n \times n$ irreducible matrix with positive entries. The perron-frobenius theory states that the greatest eigenvalue r of A, is positive number. Further more it is a single root of the characteristic polynomial of A and has the largest absolute value among all other roots. Therefore the corresponding right eigenspace associated to r is one dimensional. The same is true for the left eigenspace.Let V be the right eigenvector for A, (Respectively Let W be the left eigenvector for A), such that $W^t V = 1$. Then it is well known that as k tends to infinity, $\lim\left(A^k / r^k\right) = VW^t$. Now considering Equations (4) and (5) we can see that the matrix $C = \left(c_{i,j}\right)$ is stochastic matrix. Considering C as an operator then using the fact that all the entries of C are all positive, then the adjoint operator C^* can be defined as $C^* = \left(c^*_{i,j}\right) = \left(c_{j,i}\right)$ Furthermore Equations (2) and (3) imply that the following equalities:

$$q_2 = C^* C q_2 \tag{12}$$

$$Q_1 = q_1 C C^* \tag{13}$$

hold for each fixed $i \ge n$. But $D_1 = CC^*$ and $D_2 = C^* C$ are self adjoint matrices which implies that for $i \epsilon \langle 1, 2 \rangle$, q_i is an eigenvector for the matrix D_i corresponding to the eigenvalue equal to one. One the other hand the equalities (4) and (5), and the fact that the entries of D_i are all positive and less or equal than one, implies that the Perron Frobenius eigenvalue of D_i is equal to one. Hence q_i is a Perron Frobenius eigenvector for D_i. Now summing the Equation (3) over the index i then (12) and (13) imply the following equation

$$\Sigma q_{i,1} = \Sigma q_{i,2} \tag{14}$$

Let us set the following notations

$$\alpha_1 = \left(\Sigma_i q^2_{i,1}\right)^{1/2} \tag{15}$$

$$\alpha_2 = \left(\Sigma_i q^2_{i,2}\right)^{1/2} \tag{16}$$

Then $V_1 = q_1 / \alpha_1$ and $V_2 = q_1 / \alpha_1$ are standard Perron Frobenius eigenvectors for D_1 and D_2 respectively. Now set $V_1 = \left(v_{i,1}\right) = q_1 / \alpha_1$ and $V_2 = \left(v_{i,2}\right) = q_2 / \alpha_2$. Then replacing q_1 in (6) by $\alpha_1 V_1$, we get,

$$P = \Sigma_i Q_i P r_i - \Sigma_{i,j} v_{i,1} w_{i,j} \tag{17}$$

Let us set the following notation,

$$\omega = \Sigma_{i,j} v_{i,1} w_{i,j} \tag{18}$$

Then the Equation (17) can be written as in the following,

$$P = \Sigma_i Q_i P r_i - \left(\alpha_1 \omega\right) \tag{19}$$

Let us define,

$$det(k,j) = v_{k,1}v_{j,2} - v_{k,2}v_{j,1} \tag{20}$$

Now by (1) and the above arguments we have, $Q_i = q_i - \alpha_1 v_{i,1} + \alpha_2 v_{i,2}$. Consider two fixed indices, $0 \le k, j \le n$, such that $det(k,j) \neq 0$. Then the following equations,

$$Q_k = q_k + \alpha_2 v_{k,2} - \alpha_1 v_{k,1}$$

$$Q_j = q_j + \alpha_2 v_{j,2} - \alpha_1 v_{j,1}$$

will provide us with values of α_2 and α_1.

Let us define, $\sigma_{k,j} = \omega\left(det_{k,j}\right)^{-1}$, $\beta_{k,j} = \sigma_{k,j} v_{k,2}$, and $\gamma_{k,j} = \sigma_{k,j} v_{j,2}$

$$P = \Sigma_i Pr_i Q_i - \beta_{k,j} Q_j - \gamma_{k,j} Q_k - \gamma_{k,j} q_k - \beta_{k,j} q_j \tag{21}$$

In order to maximize P given by the Equation (21), we need to know the two values $\gamma_{k,j}$ and $\beta_{k,j}$. This can be done by training a simple perceptron which we call neural network(A), with non-linear separability function, input data, $Q_i, q_i, Pr_i, i = 1,2,\cdots$, output data P And weights that can be taken to be the values $\gamma_{k,j}, \beta_{k,j}$. As we can see at (Section-5) [3], once we train the above perceptron with large enough set of contemporary data the corresponding weights will converge and give us the above values.Let us define the following vectors. $Q = (Q_i), i = 1,2,\cdots,n$, $q = (q_i), i = 1,2,\cdots,n$, $Pr = (pr_i), i = 1,2,\cdots n$.

3. Final Conclusions

At this Section we formulate the final form of our neural network. Our neural network consists of four parts. At Part-I, we fixed two integers $0 \le k, j \le n$, such that $det(k,j) \neq 0$. Next using Neural network(A), which is based on simple perceptron we will calculate the values $\beta_{k,j}$ and $\gamma_{k,j}$ in term of large enough number of input vectors Q, Pr, q and output vector P that would be used to train neural network(A). In Part-II, we use neural network(B) that is based on linear programming (20), calculating the variables $(Q_i), i = 1,2,\cdots,n$ to maximize the profit P. The only constrains that are imposed on this neural network are the facts that the above variables are all positive and the following equation, $\Sigma_i Q_i = \Sigma_i q_i = 1$. At Part-III, neural network(C) will find the kind of crude oil that is the most beneficial for the refinery. Assuming that maximal percentage of the product q_i, inside the crude oil to be less or equal than the known number M_i. It means that at this stage we have to add another n positive variables q_i, satisfying the following set of constrains, $0 \le q_i \le M_i$, to the linear programming (20). This new linear programming will provide us the values for the two vectors $Q = (Q_i), i = 2,3,\cdots,n$ and $q = (q_i), i = 1,2,\cdots,n$, and maximal value for P in term of Q and q. Finally we can train neural network(D) that is based on back propagation. This neural net work is trained, using enough number of data which takes q and Pr as an input with P and Q as an output. After the training is completed neural network(D) is able to give us P and Q as soon as we plug in the vectors q and Pr.

References

[1] Hetz, J., Krough, A., Hetz, R.P.J., Krough, A. and Palmer, R. (1991) Introduction to the Theory of Neural Computation. Copy Right C, Addison Wesley Company.

[2] Haykin, S. (1999) Neural Networks A Comprehensive Foundation. 2nd Edition, Prentice Hall. Inc.

[3] Gass, I. (1958) Linear Programming. McGraw Hill, New York.

[4] Minc, H. (1988) Nonnegative Matrices. John Wiley and Sons. New York.

Scientific Research Publishing (SCIRP) is one of the largest Open Access journal publishers. It is currently publishing more than 200 open access, online, peer-reviewed journals covering a wide range of academic disciplines. SCIRP serves the worldwide academic communities and contributes to the progress and application of science with its publication.

Other selected journals from SCIRP are listed as below. Submit your manuscript to us via either submit@scirp.org or Online Submission Portal.