Application of Neural Network to Medicine

Bahman Mashood[1], Greg Milbank[2]

[1]San Francisco, CA, USA
[2]Praxis Technical Group Inc., Nanaimo, BC, Canada
Email: b_mashood@hotmail.com, Gregm@Praxistech.com

Received 24 October 2015; accepted 8 November 2015; published 13 November 2015

Copyright © 2015 by authors and OALib.
This work is licensed under the Creative Commons Attribution International License (CC BY).
http://creativecommons.org/licenses/by/4.0/
Open Access

Abstract

In this article we are going to introduce certain applications of the theory of neural network to medicine. In particular suppose for a given disease *D*, belonging to the category of immune deficiency diseases or cancers, there are numbers of drugs each with partial remedial effect on *D*. We are going to formulate a cocktail consisting of mixture of the above drugs with optimal remedial effect on *D*.

Keywords

Disease, Cocktail, Optimization, Neural Network, Back Propagation, Trails, Data

Subject Areas: **Drugs & Devices**

1. Introduction

Many problems in the industry involved optimization of certain complicated function of several variables. Furthermore there is usually a 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 a completely new approach. 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 generalized version of the above methods to find the optimum points for some given problems. The results in this article are based on our common work with Greg Millbank of praxis group. Many of our products used neural network of some sorts. Our experiences show that by choosing appropriate initial data and weights we are able to approximate the stability points very fast and efficiently. In [3], we introduce the extension of Cohen and Grossberg theorem to larger class of dynamic systems. The appearance of new generation of super computers will give neural network a much more vital role in the industry, intelligent machine and robotics.

How to cite this paper: Mashood, B. and Milbank, G. (2015) Application of Neural Network to Medicine. *Open Access Library Journal*, **2**: e2067. http://dx.doi.org/10.4236/oalib.1102067

2. 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 threshold 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 $S = (s_1, s_2, \cdots, s_n)$ 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 category of dynamical systems provide us with neural networks that can solve optimization problems. The input and output to these 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 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 including pattern recognition and simulations that can be applied to varieties of cancers and immune deficiency diseases. 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 output 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. At this end we have to mention that many exiting results on applications of neural networks to optimization can be found in [4].

3. On Providing a Cocktail Consisting of Certain Drugs Having Optimum Effect

Suppose there is a certain disease *D*. Next assume that there are number of drugs $n_1, n_2, \cdots, n_k$ available to treat *D*. Let the vector $V_1 \in R^n$ represent the health state of the body. Assume the vector $V_2 \in R^m$ represent the vital limitation of the body. Next let $L_1 \in R^n$, respectively $(L_2 \in R^m)$, represent the initial health state of the body, respectively (the initial vital state of the body). Suppose that the daily use of the amount α_j milligram of the product n_j will finally drive the body to the health state which is represented by the vector, $V_1(L_1, L_2, \alpha_j) \in R^n$ respectively (will drive the vital state of the body to the state which is represented by the vector $V_2(L_1, L_2, \alpha_j) \in R^m$).

We are searching for the optimal solution where the cocktail consisting of certain daily amount α_j, $j = 1, 2, \cdots, k$ from each product n_j, will provide us with the final health state $V_1(L_1, L_2, \alpha_1, \alpha_2, \cdots, \alpha_k)$ and final vital state $V_2(L_1, L_2, \alpha_1, \alpha_2, \cdots, \alpha_k)$ such that they satisfy, the following properties,

$$\text{to minimize } \left(V_1\left(L_1, L_2, \alpha_1, \alpha_2, \cdots, \alpha_k\right) - V_1\right)^2,$$

$$\text{and to maximize } \left(V_2\left(L_1, L_2, \alpha_1, \alpha_2, \cdots, \alpha_k\right) - V_2\right)^2$$

$$\text{keeping the constrain } V_2\left(L_1, L_2, \alpha_1, \alpha_2, \cdots, \alpha_k\right) < V_2 .$$

Beside that we usually have to consider certain limitation of daily use of each one of the products which are expressed in the following inequalities,

$$\alpha_j < d_j,\ j = 1, 2, \cdots, k .$$

The above conditions are equivalent in optimization the function *F*, given in the following

$$F = \left(V_1\left(L_1, L_2, \alpha_1, \alpha_2, \cdots, \alpha_k - V_1\right)\right)^2 - \lambda\left(V_2\left(L_1, L_2, \alpha_1, \alpha_2, \cdots, \alpha_k\right) - V_2\right)^2$$

Together with the set of constrains

$$0 \le \alpha_j \le d_j,\ j = 1, 2, \cdots, k \text{ and } V_2\left(L_1, L_2, \alpha_1, \alpha_2, \cdots, \alpha_k\right) < V_2,$$

where λ is an arbitrary positive number chosen in an appropriate way, usually at the same magnitude as to the set $d_j, j = 1, 2, \cdots, k$.

Finally the above system of optimization corresponds to the following energy function,

$$E\left(L_1, L_2\right) = F + \gamma\left(V_2\left(L_1, L_2, \alpha_1, \alpha_2, \cdots, \alpha_k\right) - V_2\right) + \sum_{j=1}^{j=n} \lambda_j\left(\alpha_j - d_j\right) + \sum_{j=1}^{j=n}\left(-\varphi_j \alpha_j\right)$$

where $\gamma, \lambda, \lambda_1, \lambda_2, \cdots, \lambda_k, \varphi_1, \varphi_2, \cdots, \varphi_k$, are fixed positive numbers that are chosen in an appropriate way.

Also we can choose simpler approach by formulating the energy function *E* in the following

$$E\left(L_1, L_2\right) = F + \lambda\left(V_2\left(L_1, L_2, \alpha_1, \alpha_2, \cdots, \alpha_k\right) - V_2\right)$$

where we keep the set of constrains, $0 \le \alpha_j \le d_j, j = 1, 2, \cdots, n$ and consider λ as an independent variable to be added to the set of variables $\alpha_j, j = 1, 2, \cdots, k$, and set it to be equal to α_{k+1}.

The problem with this approach is that now the boundary points can be candidate for optimum point too.

Provided we have the values,

$$V_j\left(L_1, L_2, \alpha_1, \alpha_2, \cdots, \alpha_k\right), j = 1, 2, \text{ for each vector } \left(\alpha_1, \alpha_2, \cdots, \alpha_k\right) \in \left(R^+\right)^k,$$

then using the arguments in [3], we are able to calculates immediately the vector $\left(\gamma_1, \gamma_2, \cdots, \gamma_k\right) \in \left(R^+\right)^k$ that will give us the optimum, where R^+ represent the set of all positive real. For example If the values of $V_j\left(L_1, L_2, \alpha_1, \alpha_2, \cdots, \alpha_k\right), j = 1, 2$, can be assumed to be equal a linear sum of $V_j\left(L_1, L_2, \alpha_l\right), l = 1, 2, \cdots, n$, $j = 1, 2$, then we are done, otherwise if the assumption of linearity fails we have to use a back propagation neural network N_1, as we describe at the Section-1. In this case first we have to train the neural network N_1 using sufficiently many experiences on randomly chosen patients with randomly chosen parameters

$$L_1, L_2, \alpha_j, j = 1, 2, \cdots, n.$$

This process depending on the number of available patients and duration of the trails might takes up to two years. At this point we have to mention that though it seems very complicated but in the future the simulation on neural network might replace the human trails which will contribute immensely to the field of medical research.

Next we feed the vectors $V_j\left(L_1, L_2, \alpha_1, \alpha_2, \cdots, \alpha_k\right), j = 1, 2$, corresponding to the randomly chosen patient of the above to back propagation neural network N_1, for training. Once the training is complete, for a given vector $\left(L_1, L_2, \beta_1, \beta_2, \cdots, \beta_n\right)$ the neural network N_1, will provide us with the vectors $VN_j\left(L_1, L_2, \beta_1, \beta_2, \cdots, \beta_k\right)$, $j = 1, 2$.

Finally for a given small $\epsilon > 0$, suppose the set of vectors $(\beta_1, \beta_2, \cdots, \beta_n) \in (R^+)^n$ are chosen such that they will cover up to the above $\epsilon > 0$ the interval

$$J = [0, d_1] \otimes [0, d_2] \otimes \cdots \otimes [0, d_n] \subset (R^+)^n .$$

Next let $(\gamma_1, \gamma_2, \cdots, \gamma_k) \in J$ be the vector to optimize the energy function $E(L_1, L_2)$, hence to give us the optimum cocktail for a patient with a initial condition (L_1, L_2).

Now when we feed the new patient with the optimum cocktail, to the end of the trail we measure the actual values of $V_j(\gamma_1, \gamma_2, \cdots, \gamma_k), j = 1, 2$ and comparing them with the neural network N_1 output $VN_j(\gamma_1, \gamma, \cdots, \gamma_k), j = 1, 2$, the differences will be used to train further the back propagation neural network N_1 and to get new synaptical values. Hence after many trails the neural network will be able to find the exact optimum values for the cocktail that is appropriate for the patient with initial condition (L_1, L_2).

4. Conclusion

There is a set of diseases which cannot be cured using the existing set of drugs. But some of the above drugs can improve the state of health of an infected person. We suggest that it is possible to make a cocktail consisting of the above drugs which will provide us with the most effective treatment of the above person. We suggest that this method can be set on trails and be examined.

References

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

[2] Haykin, S. (1999) Neural Networks: A Comprehensive Foundation. 3rd Edition, Prentice Hall Inc., Upper Saddle River.

[3] Mashood, B. and Millbank, G. (2015) Advances in Theory of Neural Network and Its Application. *Journal of Global Prospect of Artificial Intelligent*, Preprint.

[4] Takefuji, Y. Neural and Parallel Processing. *The Kluwer International Series in Engineering and Computer Science*, SEeS0164.