

Method of Forecasting an Oil Spreading on Water by Means of Neural Networks

Ghasemi Habashi YA¹, Gasimov BM², Efendiyeva HC¹ and Mutallimov MM^{1*}

¹Institute of Applied Mathematics of Baku State University, Z. Khalilov Street, 23, AZ1148 Baku City, Azerbaijan

²Azerbaijan State University of Economics, Baku, Istiqlaliyyat str., 6, AZ1001, Baku City, Azerbaijan

Abstract

This paper is devoted to calculating the oil spot spreading in the sea by means of neural networks. Here the problem is found out oil spot spreading in the sea after some real times and we can determine the process with the neural networks.

Keywords: Oil; Sea; Neural; Networks

Introduction

In a wide variety of disciplines it is of great practical importance to measure, to describe and compare the shapes of objects. In general terms, the shape of an object, data set, or image can be defined as the total of all information that is invariant under translation, rotation and isotropic rescaling. The field of shape analysis involves hence methods for the study of the shape of objects where location, rotation and scale can be removed. The two or more dimensional objects are summarized according to key points called landmarks. This approach provides an objective methodology for classification whereas even today in many applications the decision for classifying according to the appearance seems at most intuitive.

Statistical shape analysis is concerned with methodology for analyzing shapes in the presence of randomness. It is a mathematical procedure to get the information of two or more dimensional objects with a possible correction of size and position of the object. So objects with different size and/or position can be compared with each other and classified. To get the shape of an object without information about position and size, centralization and standardization procedures are used in some metric space.

Interest in shape analysis began in 1977. Kendall published [1] a note in which he introduced a new representation of shapes as elements of complex projective spaces. Mardia on the other hand investigated the distribution of the shapes of triangles generated by certain point processes, and in particular considered whether towns in a plain are spread regularly with equal distances between close towns.

The full details of this elegant theory which contains interesting areas of research for both probabilists and statisticians where published by Kendall and Bookstein [1]. The details of the theory and further developments can be found in the textbooks [1,2].

Neural Network

Neural networks have been developed originally in order to understand the cognitive processes. Nowadays there are a lot of applications of neural networks as a mathematical method in quite different disciplines.

The term “neural networks” points to the model of a nerve cell, the neuron, and the cognitive processes carried and driven by the network of interacting neurons. A neuron perceives chemical and physical excitement from the environment by its dendrites. The neuron is

processing this incoming data and sending the information to other neurons via axon and synapses.

McCulloch and Pitts implemented the biological processes of a nerve cell for the first time in the mathematical way. Nerve cells have to access and process incoming data in order to evaluate target information. Therefore the corresponding neural networks are called supervised neural networks. An unsupervised neural network has no target and is similar to a cluster algorithm.

The data consists of n variables x_1, \dots, x_n on binary scale. For data processing, the i 'th variable x_i is weighted with w_i . Normalized with $|w_i| \leq 1$ the multiplication of x_i with w_i determines the relevance of x_i for a target. The value y w_i reflects the correlation between the input variable and the target, the sign indicating the direction of the influence of the input variable on the target. Weighting the input variables for a target variables is similar to discriminant analysis [3].

The critical quantity for the neuron is the weighted sum of input variables

$$q = \sum_{i=1}^n w_i x_i = w_1 x_1 + \dots + w_n x_n \quad (1)$$

For the target y with binary scale, a threshold S is needed. Crossing the threshold yields 1 and falling below the threshold yields 0. Hence the activation function F can be written as

$$F(q) = \begin{cases} 1, & \text{if } x > s \\ 0, & \text{if } x \leq s \end{cases} \quad (2)$$

In comparison to discriminant analysis, for neural networks the threshold S has to be assigned, depending on properties of the target; it cannot be derived from the data in a straightforward manner. Neural networks usually include no assumption about the data. Rather they are a numerical method [4].

***Corresponding author:** Mutallimov MM, Institute of Applied Mathematics of Baku State University, Z. Khalilov Street, 23, AZ1148 Baku City, Azerbaijan, Tel: 994125391595; E-mail: mutallim@mail.ru

Received July 14, 2014; **Accepted** October 28, 2014; **Published** November 07, 2014

Citation: Ghasemi Habashi YA, Gasimov BM, Efendiyeva HC, Mutallimov MM (2014) Method of Forecasting an Oil Spreading on Water by Means of Neural Networks. J Appl Computat Math 3: 189. doi:10.4172/2168-9679.1000189

Copyright: © 2014 Ghasemi Habashi YA, et al. This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

With the input of the activation function, we obtain $y=F(q)$ as

$$y = 1, \text{ if } \sum_{i=1}^n w_i x_i > s_n$$

$$y = 0, \text{ if } \sum_{i=1}^n w_i x_i \leq s_n$$

To simulate all logical more than one layer is necessary.

Multi-layer Perceptrons

In general a given target may be reached only up to a certain error. Given a certain measure $E(\hat{y}, y)$ for the distance between the given target state y and the state \hat{y} computed by the neural network, learning of the neural network corresponds to minimization of $E(\hat{y}, y)$. The following training algorithm is inspired by Rumelhart, Hinton and Williams. The total error measure over all states of a given layer is defined as

$$E_{total}(\hat{y}, y) = \frac{1}{2} \sum_{k=1}^N (\hat{y}_k - y_k)^2 \quad (3)$$

Different kind of error measurements suited for the application can be applied. The upper error measurement will be used below to reset the weight in each layer of the neural network.

For simplicity, we consider now a 2-layer perceptron network, which also will be sufficient bellow for our purpose of calibrating the stochastic process.

The processed state \hat{y} of the neural network is computed by the following steps. First the critical parameter for the first layer is computed from n weighted input values as $\sum_{i=1}^n w_i x_i$ we consider a hidden output layer with m neurons. For n let g_j be the activation function of the j -th neuron of the hidden layer with an activation value of h_j given as

$$h_j = g_j \left(\sum_{i=1}^n w_i x_i \right) \quad (4)$$

Usually for all neurons of a given layer common activation function $g=\{g_1, \dots, g_m\}$, e.g. a sigmoid function, is used. Alternatively for simulating cyclical processes trigonometric function can be applied. This would be the case, if we assume that the same input value in combination with the time point has to be interpreted contrary. Next, the output of the previous (hidden) layer becomes the input of the next layer, and the activation proceeds analogously to the previous layer.

Let f be the activation function of the pre-final (here the second) output layer. Then the pre-final critical value is

$$q = f \left(\sum_{j=1}^m u_j h_j \right) \quad (5)$$

Finally, the pre-final critical value q is interpreted by a final activation function F yielding

$$\hat{y} = F(q) \quad (6)$$

As a final state value computed from the neural network with the given weight of the input variables from input and hidden layers.

Now the neural network performs a training step by modifying the weight of all input layers. The learning mechanism the weights are determined by the target distance measure (3) he weights of both layers are changed according to the steepest descent, i.e.

$$\nabla_{w_i} E = \frac{\partial E}{\partial w_i} \quad (7)$$

$$\nabla_{u_i} E = \frac{\partial E}{\partial u_i} \quad (8)$$

With a learning rate α which should be adapted to the data, the weights are changed as follows:

$$w_i^{new} = w_i^{old} - \alpha \nabla_{w_i} E \quad (9)$$

$$u_i^{new} = u_i^{old} - \alpha \nabla_{u_i} E \quad (10)$$

Method of Solution

The necessary number of iterations depends on the requirements posed by the data, the user, and the discipline [5].

Instead of the error function, we are using the variance. We try to find an optimal variance for differentiating our groups.

Spreading of the oil spot in the sea and enlarge it, is harmful for sea animals and natural Geographic, to this reason and according to research it's very necessary to find out the size of spreading and can intercept or arrest it (Figures 1 and 2) [6].

The weather condition, gale energy, sun refulgence, water density

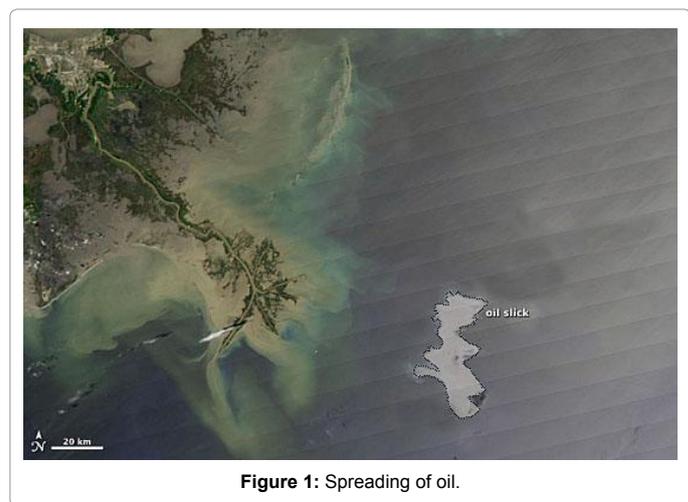
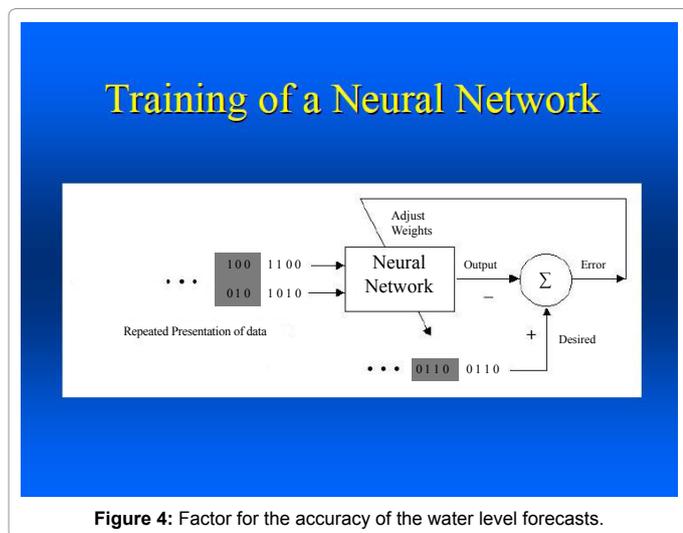
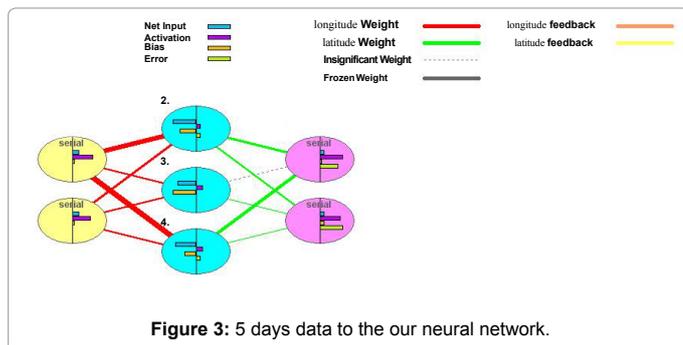


Figure 1: Spreading of oil.



Figure 2: Oil slick.



and water clarity is very important parameters in oil spreading, but we want to solve the problem by neural networks without these problems.

We have so many methods to estimating the future of some things or process in different science, but also we know that the results of each method is different with another method and all of the results not really correct, results are closer to target.

Same we show in Figures 1 and 2, in our method first we determined 200 different points in outside circumference of oil spot, that these points are very close together, then get the all longitudes and latitudes of these points, and register these. We register new longitudes and latitudes of those points at each 6 hours on 5 sequential days, after 5 days we have new 20 longitudes and 20 latitudes for each point, then we input continual all 5 days data to the our neural network (Figure 3).

Results

Neural network features

1. non linear modeling capability
2. generic modeling capability
3. robustness to noisy data
4. ability for dynamic learning
5. requires availability of high density of data

Neural network modeling shows excellent promises for local forecasting of water levels, computationally and financially inexpensive method.

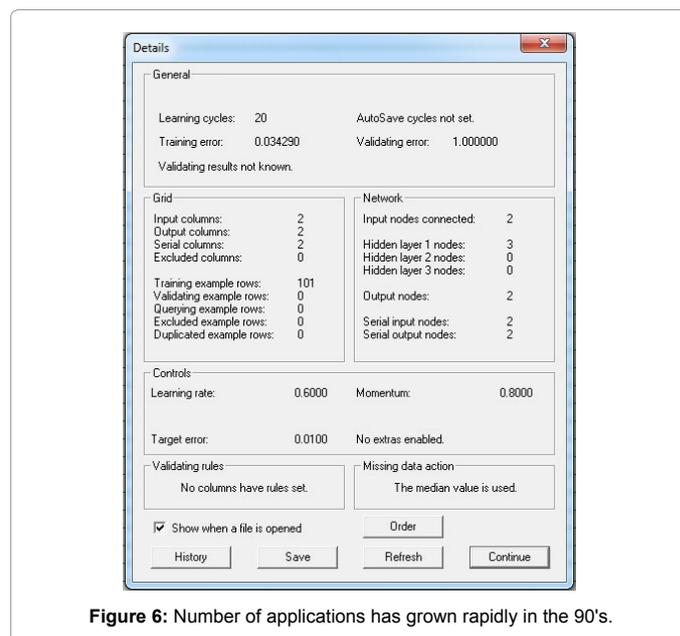
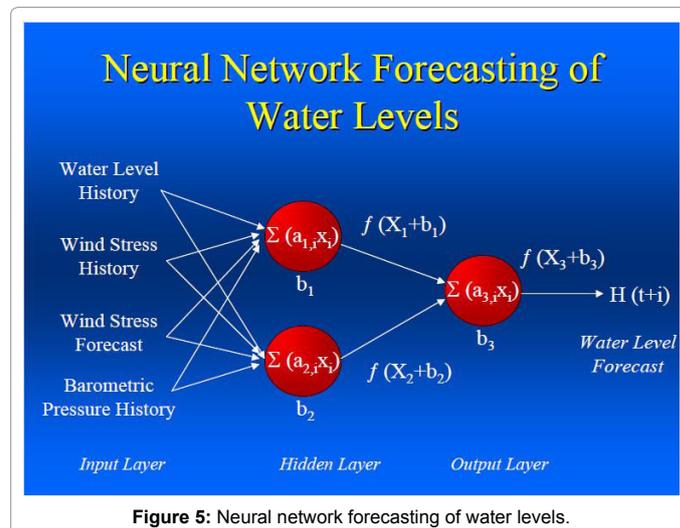
The quality of the wind forecasts will likely be the limiting factor for the accuracy of the water level forecasts (Figure 4).

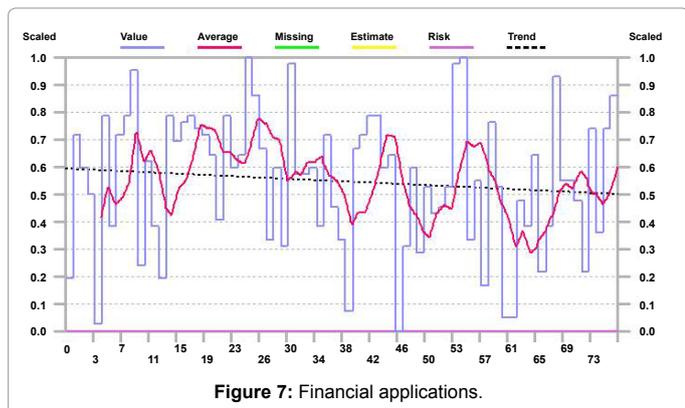
First use historical time series of previous water levels, winds, barometric pressure as input then train neural network to associate changes in inputs and future water level changes after that make water level forecasts using a static neural network model (Figure 5).

Neural network modeling started in the 60's then key innovation in the late 80's: backpropagation learning algorithms after that number of applications has grown rapidly in the 90's especially financial applications and then growing number of publications presenting environmental (Figures 6 and 7).

Conclusion

In our learning algorithm we use previous data to estimate next data (like Fibonacci method). Now a day we begin to input all of data (5 day's data) to network and according to our learning algorithm, this network give us the new longitudes and latitudes of those points on 6th day (output) [7].





The points longitudes and latitudes on 6th day, show us the estimating result of oil spot spreading (enlargement size) in the sea, that we can find it by means of neural network without using any human or other way.

But the best result of this network is, find out the estimating result of enlargement size on 9th day without calculating results of 6th, 7th and 8th days.

Note that using the results of [8,9] we can apply this technique to other practical problems.

References

1. Kendall DG (1977) The diffusion of shape. *Adv Appl Probab* 9: 428-430.
2. Dryden IL, Mardia KV (1998) *Statistical shape Analysis*. J. Wiley Sons.
3. Bookstein FL (1986) Size and shape spaces for landmark data in two dimensions (with discussion). *Statistical Science* 1: 181-242.
4. Coppes MJ, Campbell CE, Williams BRG (1995) *Wilms Tumor: Clinical and Molecular Characterization*. Austin Texas USA: RG Landes Company.
5. Small CG (1996) *The Statistical Theory of Shape*. Springer-Verlag, New York.
6. Bishop CM (1995) *Neural networks for pattern recognition*, Clarendon Press, Oxford.
7. Giebel S (2007) *Statistical Analysis of the shape of renal tumors in childhood*. Diploma thesis, University Kassel.
8. Aliev FA, Mutallimov MM, Ismailov NA, Radzhabov MF (2012) Algorithms for constructing optimal controllers for gaslift operation. *Automation and Remote Control* 73: 1279-1289.
9. Majidzadeh K, Mutallimov MM, Niftiyev AA (2012) The problem of optimizing the torsional rigidity of a prismatic body about a cross section. *Journal of Applied Mathematics and Mechanics* 76: 482-485.