A Soil Moisture Map Generated from Satellite Data by Using Domains of Attraction in Neural Networks

Ohkubo, Akito
Department of Informatics Kyushu University

Mohamed, Marghny H.
Department of Informatics Kyushu University

Niijima, Koichi
Department of Informatics Kyushu University

出版情報：DOI Technical Report. 152, 1998-07-22. Department of Informatics, Kyushu University
バージョン：
権利関係：

DOI-TR-152

DOI Technical Report

A Soil Moisture Map Generated from Satellite Data by Using Domains of Attraction in Neural Networks

by

Akito Ohkubo, Marghny Mohamed, Koichi Niijima

July 22, 1998

Department of Informatics
Kyushu University
Fukuoka 812-81, Japan

Email: a-ohkubo@i.kyushu-u.ac.jp Phone: +81-92-092-921-9942

A Soil Moisture Map Generated from Satellite Data by Using Domains of Attraction in Neural Networks

Akito Ohkubo, Marghny H.Mohamed and Koichi Niijima
Department of Informatics
Kyushu University
6-1, Kasuga-Koen, Kasuga 816-8580
Japan
a-ohkubo@i.kyushu-u.ac.jp

Abstract

A learning method using domains of attraction in three-layered neural networks is proposed. The method is a combination of output error minimization learning with maximization learning of domains of attraction in one-layered perceptrons. To simplify the structure of the network, a successive learning technique is employed for hidden units. Domains of attraction in the network is derived by restricting output conditions at the hidden layer for training input data.

A three-layered neural network is determined by this method using a training set which consists of satellite observation data and soil moisture data surveyed in some places. The estimation of soil moisture at all places corresponding to satellite data is carried out based on the derived domains of attraction in the network.

Keyword:learning method, domain of attraction, satellite data, soil moisture data

1 INTRODUCTION

The back propagation method (BPM) has been used well for learning multi-layered neural networks. Until now, many variants of BPM have been developed to solve application problems [4]. However, the neural network learnt by BPM does not have estimation ability for non-training data.

In this paper, we propose a learning method using domains of attraction in three-layered neural networks. Our approach is based on two optimization learning methods. One is related to a minimization of output errors for a training set such as BPM. The minimization learning of output errors is done by adding hidden units successively to simplify the structure of the network. The other concerns a maximization of domains of attraction derived by imposing firing conditions on hidden layer outputs for training input data. It has been shown in the paper [1] that any element belonging to the domain of attraction has almost the same output in one-layered perceptron. Such domains of attraction enable us to classify unknown input data.

A three-layered neural network is constructed by our learning method to obtain a soil moisture map. A training set has satellite observation data as input data and soil moisture data surveyed in some places as supervised values. We have many satellite observation data, but only a little soil moisture data. Therefore, a training set is usually small. By our learning method, we can obtain domains of attraction, each of which contains vectors of satellite observation data corresponding to each surveyed soil moisture datum. By virtue of the domains, we can estimate unknown soil moisture at the places where soil samples are not gathered.

2 LEARNING METHOD

2.1 Three Layered Neural Networks

We consider a three-layered neural network:

$$y = g\left(\sum_{i=1}^{h} w_i f\left(\sum_{k=1}^{n} v_{ik} x_k - \theta_i\right)\right), \tag{1}$$

where x_k is an input, v_{ik} denotes a connection weight between the input and output layers, θ_i indicates a threshold, w_i denotes a weight connecting the hidden and output layers, and y is an output. The functions $f(t)$ and $g(t)$ are sigmoid functions given by

$$f(t) = \frac{1 - e^{-t}}{1 + e^{-t}}, \qquad g(t) = \frac{1}{1 + e^{-t}}.$$

This network is shown in Figure 1.

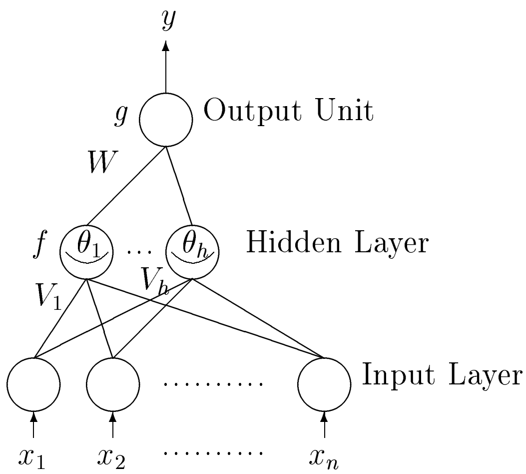

Figure 1: Three-layered neural network

We put $x = (x_1, x_2, \cdots, x_n)$ and $V_i = (v_{i1}, v_{i2}, \cdots, v_{in})$, and define $\varphi_i(x)$ by

$$\varphi_i(x) = f(V_i{\cdot}x - \theta_i),$$

where $\cdot$ denotes the inner product symbol. Using $W = (w_1, w_2, \cdots, w_h)$ and $\varphi(x) = (\varphi_1(x), \varphi_2(x), \cdots, \varphi_h(x))$, we can write (1) as

$$y = g(W \cdot \varphi(x)).$$

2.2 Minimization of Output Errors

Let (x^ν, y^ν), $\nu = 1, 2, \cdots, m$, be training data. Although the output error for x^ν may be expressed as $y^\nu - g(W{\cdot}\varphi(x^\nu))$, we now define the output error by

$$c^\nu = g^{-1}(y^\nu) - W \cdot \varphi(x^\nu).$$

We assume that V_i, θ_i and W have already been learnt. Adding one more unit to the hidden layer, we define the new connection weight vector and the output function at the hidden layer by $\widetilde{W} = (W, w)$ and $\widetilde{\varphi}(x) = (\varphi(x), f(v{\cdot}x - \theta))$, respectively, where v and w denote new weights, and θ a threshold.

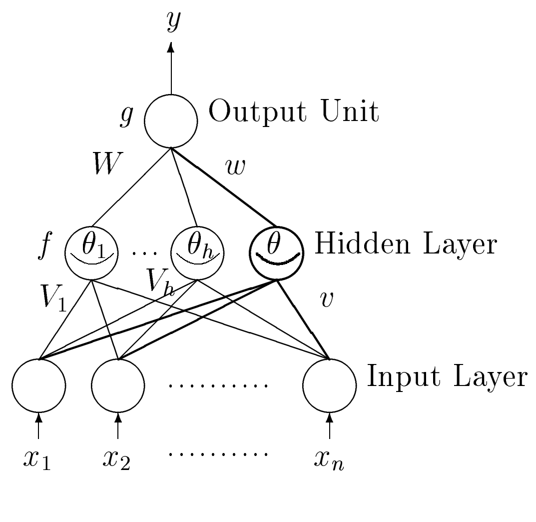

Figure 2: Three-layered neural network after adding one unit in the hidden layer in Figure 1

Then, the new output error can be written as

$$\widetilde{c}^\nu = g^{-1}(y^\nu) - \widetilde{W} \cdot \widetilde{\varphi}(x^\nu).$$

We have already known in [2] that if we choose

$$w = \frac{\sum_{\nu=1}^{m} f(v \cdot x^{\nu} - \theta) c^{\nu}}{\sum_{\nu=1}^{m} f^2(v \cdot x^{\nu} - \theta)},$$

the following relation holds:

$$\sum_{\nu=1}^{m} (\tilde{c}^{\nu})^2 = \sum_{\nu=1}^{m} (c^{\nu})^2 - I(v, \theta),$$

where

$$I(v, \theta) = \frac{(\sum_{\nu=1}^{m} f(v \cdot x^{\nu} - \theta) c^{\nu})^2}{\sum_{\nu=1}^{m} f^2(v \cdot x^{\nu} - \theta)}.$$

It is desirable for $I(v, \theta)$ to be large. There are many methods for determining v and θ so as to maximize $I(v, \theta)$. In the next section, we propose a method for determining such parameters with the help of domains of attraction in the network.

2.3 Domains of Attraction

We assume that $\varphi_i(x^{\nu}) \leq -1 + \varepsilon$ or $\varphi_i(x^{\nu}) \geq 1 - \varepsilon$ holds for V_i, θ_i and W already determined, where $\varepsilon > 0$ is sufficiently small. We define two index sets $I_{\nu,-}$ and $I_{\nu,+}$ as follows:

$$I_{\nu,-} = \{i \mid \varphi_i(x^{\nu}) \leq -1 + \varepsilon\},$$
$$I_{\nu,+} = \{i \mid \varphi_i(x^{\nu}) \geq 1 - \varepsilon\},$$

where $I_{\nu,-} \cup I_{\nu,+} = \{1, 2, \cdots, h\}$.
By the results in [1], we can define the domain

$$D_{\rho}(x^{\nu}) = \{x \mid V_i \cdot (x - x^{\nu}) \leq \rho \left| V_i \cdot x^{\nu} - \theta_i \right|, i \in I_{\nu,-} \\ V_i \cdot (x - x^{\nu}) \geq -\rho \left| V_i \cdot x^{\nu} - \theta_i \right|, i \in I_{\nu,+}\}, \quad (2)$$

where $0 < \rho < 1$, and it follows that for any x in $D_{\rho}(x^{\nu})$, the output $\varphi(x)$ at the hidden layer is almost the same as $\varphi(x^{\nu})$.

For new weight v and threshold θ, we assume that

$$f(v \cdot x^{\nu} - \theta) \leq -1 + \varepsilon \quad (3)$$

or

$$f(v \cdot x^{\nu} - \theta) \geq 1 - \varepsilon \quad (4)$$

is satisfied. For latter convenience, we rewrite (3) and (4) as

$$v \cdot x^{\nu} - \theta \leq -ln \frac{1 - \varepsilon}{\varepsilon} \quad (5)$$

or

$$v{\cdot}x^\nu - \theta \;\; \geq \;\; ln\frac{1-\varepsilon}{\varepsilon}. \tag{6}$$

By this assumption, it follows that for any x in

$$D_\rho^-(x^\nu) = D_\rho(x^\nu) \cap \{x \mid v \cdot (x - x^\nu) \leq \rho\,|v \cdot x^\nu - \theta|\}$$

or

$$D_\rho^+(x^\nu) = D_\rho(x^\nu) \cap \{x \mid v \cdot (x - x^\nu) \geq -\rho\,|v \cdot x^\nu - \theta|\},$$

the output $\tilde{\varphi}(x)$ is almost equal to $\tilde{\varphi}(x^\nu)$ (see [1]).

2.4 Cost Function

It is desirable for $D_\rho^-(x^\nu)$ and $D_\rho^+(x^\nu)$ to be as large as possible. We may concentrate on $D_\rho^-(x^\nu)$ because the situation is the same for $D_\rho^+(x^\nu)$. Notice here that $D_\rho^-(x^\nu)$ can be decomposed as $D_\rho^-(x^\nu) = Cone^-(x^\nu) \cup Str^-(x^\nu)$, where

$$\begin{aligned} Cone^-(x^\nu) = \{x \mid\, & V_i{\cdot}(x - x^\nu) \leq 0,\; i{\in}I_{\nu,-}, \\ & V_i{\cdot}(x - x^\nu) \geq 0,\; i{\in}I_{\nu,+},\; v{\cdot}(x - x^\nu) \leq 0\} \end{aligned}$$

and

$$\begin{aligned} Str^-(x^\nu) = \{x \mid\, & 0 < V_i \cdot (x - x^\nu) \leq \rho\,|V_i \cdot x^\nu - \theta_i|\,,\; i \in I_{\nu,-}, \\ & 0 > V_i \cdot (x - x^\nu) \geq -\rho\,|V_i \cdot x^\nu - \theta_i|\,,\; i \in I_{\nu,+}, \\ & 0 < v \cdot (x - x^\nu) \leq \rho\,|v \cdot x^\nu - \theta|\}. \end{aligned} \tag{7}$$

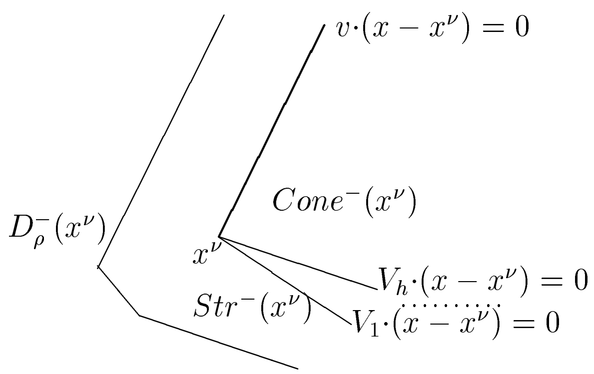

Figure 3: $D_\rho^-(x^\nu)$, $Cone^-(x^\nu)$ and $Str^-(x^\nu)$

To enlarge the domain $D_\rho^-(x^\nu)$, it suffices to maximize the angles of $Cone^-(x^\nu)$ and to make large the width of $Str^-(x^\nu)$. It was shown in [1] that for the angle γ at which the hyperplanes $V_j\cdot(x-x^\nu)=0$ and $v\cdot(x-x^\nu)=0$ cross, $cos\gamma$ is expressed by $V_j\cdot v/\,\|V_j\|\,\|v\|$ and that the width of $Str^-(x^\nu)$ is given by $|v\cdot x^\nu-\theta|\,/\,\|v\|$. Therefore, we minimize the cost function:

$$J(v,\theta)=\frac{\|v\|^2}{\sum_{\nu=1}^m(v\cdot x^\nu-\theta)^2}+C_1\sum_{j=1}^h\frac{V_j\cdot v}{\|V_j\|\,\|v\|}$$
$$+C_2\sum_{\nu=1}^m\left(ln\frac{2-\varepsilon}{\varepsilon}-v\cdot x^\nu+\theta\right)_+^2\left(ln\frac{2-\varepsilon}{\varepsilon}+v\cdot x^\nu-\theta\right)_+^2, \qquad (8)$$

where C_1 and C_2 denote penalty constants, $z_+^2=z^2$ if $z\geq 0$ and $z_+^2=0$ if $z<0$, and we have transformed the conditions (5) and (6).

In actual computation, however, we minimize the following functional to avoid numerical instability:

$$J(U,\beta,\eta)=\frac{1}{\sum_{\nu=1}^m(U\cdot x^\nu-\eta)^2}+C_1\sum_{j=1}^h U_j\cdot U+C_3(\|U\|^2-1)^2$$
$$+C_2\sum_{\nu=1}^m(\alpha-\beta\,(U\cdot x^\nu-\eta))_+^2\,(\alpha+\beta(U\cdot x^\nu-\eta))_+^2\,, \qquad (9)$$

where C_3 denotes a penalty constant and we have assumed $U_j=V_j/\|V_j\|$, $\alpha=ln((2-\varepsilon)/\varepsilon)$, $U=v/\|v\|$, $\beta=\|v\|$ and $\eta=\theta/\|v\|$.

We minimize (9) by using, for example, the gradient method to obtain U,η and β, and compute the connection weight $v=\beta U$ and the threshold $\theta=\beta\eta$.
We must also minimize $-I(v,\theta)\equiv-I(U,\beta,\eta)$ appeared in Section 2.2. Thus, we minimize finally the following cost function:

$$K(U,\beta,\eta)=J(U,\beta,\eta)-CI(U,\beta,\eta)$$

with a penalty constant C.

3 SOIL MOISTURE MAP

3.1 Satellite Data and Surveyed Soil Moisture Values

We applied our learning method to make a soil moisture map based on the data observed by artificial satellites [3]. The object area is about 20 km all sides of Chikushi plain in the north of Kyushu island of Japan. We focused the area enclosed with the square in Figure 4.

First, we show how to make the input vector x=(x_1,x_2,x_3,x_4). The first three components x_1,x_2 and x_3 are chosen as

$$x_k=20\,log_{10}(I_k)+CF_k, \quad k=1,2,3.$$

Here, I_1,I_2 and I_3 denote the œ[2 0 values observed by the artificial satellites ERS-2, JERS-1 and JERS-1, respectively, and CF_k are given in advance.

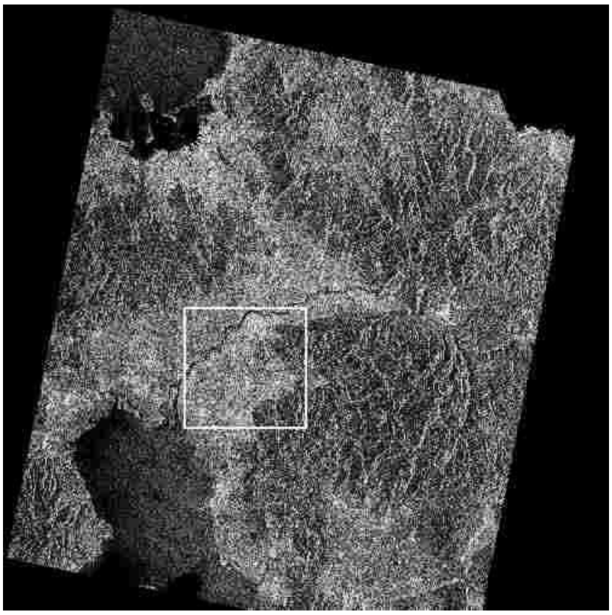

Figure 4: The object area

Also, x_4 is calculated by

$$x_4 = \frac{NIR - VIS}{NIR + VIS},$$

where NIR and VIS reveal the near-infrared and visible red values, respectively. NIR and VIS are obtained from SPOT-2 and LANDSAT-5.
Next, we give the training data $(x^\nu, y^\nu), \nu = 1, 2, \cdots, 18$. The supervised values y^ν indicate soil moisture which were surveyed in some places. The training input data x^ν denote the satellite data x corresponding to y^ν.

3.2 Learning and A Soil Moisture Map

From the construction of data in the previous section, the number n of input nodes is 4 and the number of the training data is 18. In simulations, we chose $\varepsilon = 10^{-10}$, $C_1 = 9.5$, $C_2 = 10.0$ and $C_3 = 1.0$, and changed h from 1 to 15. The output values at the hidden layer were as in Table 1 in which we wrote "+1" if $f(v \cdot x^\nu - \theta) \geq 1 - \varepsilon$, and "-1" if $f(v \cdot x^\nu - \theta) \leq -1 + \varepsilon$. From Table 1, we see that the 18 training input data were grouped in 7 classes. Of course, we can obtain the domains of attraction. Table 2 says that when the number of hidden units increases, the output errors are decreasing while the number of unclassified pixels increase. According to the statistical regression analysis, the correlation coefficient between the supervised values y^ν and their estimates is 0.455 in this problem. Since this value is close to 0.434 in Table 2, we adopt a neural network with 5 hidden units. Figure 5 shows the soil moisture map made based on the domains of attraction in this network. In Figure 5, we masked the mountain area because the soil moisture was surveyed only in the plain. The

Table 1: The output values at the hidden layer for the training data

$\nu \setminus n$	1	2	3	4	5
1	+1	-1	+1	-1	-1
2	+1	-1	+1	+1	+1
3	+1	-1	+1	-1	+1
4	-1	-1	-1	+1	+1
5	-1	-1	-1	-1	+1
6	-1	-1	-1	-1	+1
7	+1	-1	+1	-1	-1
8	-1	-1	-1	-1	-1
9	-1	-1	-1	+1	+1
10	+1	-1	-1	-1	-1
11	+1	-1	+1	-1	-1
12	+1	-1	+1	+1	+1
13	+1	-1	+1	-1	-1
14	+1	-1	+1	+1	+1
15	+1	-1	+1	-1	+1
16	+1	-1	+1	+1	+1
17	+1	-1	+1	-1	-1
18	-1	-1	-1	-1	+1

obtained domains of attraction covered 86% in the plain area where soil moisture can be estimated.

Table 2: The trade-off between the output error and the number of hidden units ($\rho = 0.9$)

	Number of Hidden Units		
	5	10	15
Output Error	0.991	0.875	0.753
Coverage Rate(%)	86.02	44.46	27.64
Correlation Coefficient	0.434	0.530	0.647

4 CONCLUSIONS

We proposed a learning method for three-layered neural networks based on the domains of attraction. We constructed a neural network using our method to make a soil moisture map. As a result, we could obtain the domains of attraction for classifying the satellite data which enable us to estimate soil moisture values. Although the increase of the hidden units can make small output errors for training data, the

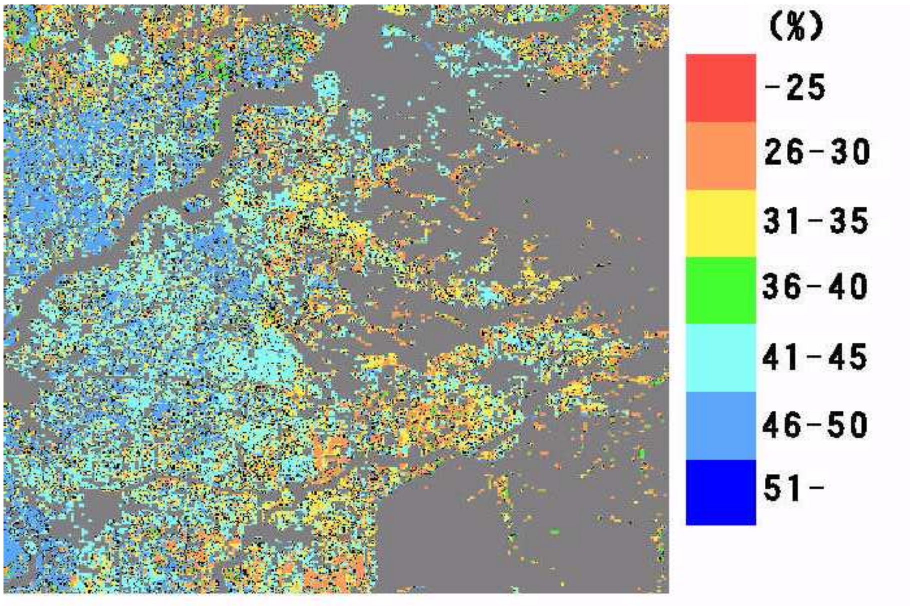

Figure 5: A soil moisture map

size of domains of attraction becomes small, which is a trade-off problem.

References

[1] Niijima, K., "Learning of Associative Memory Networks Based upon Cone-Like Domains of Attraction (Journal)," *Neural Networks*, **Vol.10**, 9, pp.1649–1658 (1997).

[2] Niijima, K. et al, "Minimization Learning of Neural Networks by Adding Hidden Units (Journal)," *Research Reports on Information Science and Electrical Engineering of Kyushu University*, **Vol.2**, 2, pp.173–178 (1997).

[3] Ohkubo, A., Takagi, J., Kuroyanagi, N. and Hatae, N., "The Estimation of Soil Moisture by Multiple Artificial Satellite Data and the Synchronized Survey with Satellite Flying Time (Proceeding)," Proceedings of the 23rd Japanese Conference on Remote Sensing, pp.221–222 (1997).

[4] Rumelhart,D.E. and McClelland,J.L., *Parallel Distributed Processing – Book –*, Cambridge, Mass., MIT Press, 1986.