CLASSIFICATION OF SOIL TYPE USING BACKPROPAGATION NEURAL NETWORK

Wanna Htun, Sandar Htay
Computer University, Hpa-an
wannahtunn2003@gmail.com

Abstract

A number of classification systems have been developed depending on the intended purpose of the system. Soil classification has proved to be very useful for soil engineers. Classification of soil type is an ever challenging area of investigation for scientists. This system is used to classify soil according to their general behavior under given physical conditions. The system is present soil classification using Backpropagation Neural Network. The result is very encouraging and it is found that the feature based soil type can make prediction with degree of accuracy. Classification of soil type can also build an awareness and knowledge of each irrigated field.

1. Introduction

A classification of soil type is one of the most important attributes in agriculture and irrigates fields. Agriculture sectors as well as many agricultural industries and irrigated fields are largely depended on soil type. This system is capable of providing only such information which is usually generalized over a larger geographical area. This system is capable of soil classification for a particular station using the data collected locally.

In this paper, a model for classification is proposed using Artificial Neural Networks. The Artificial Neural Networks use many simplifications over actual biological neurons that help us to use the computational principles employed in the massively parallel machine. The neural networks adaptively change their synaptic weights through the process of learning. The knowledge that an Artificial Neural Networks gains about a problem domain is encoded in the weights assigned to the connections of the Artificial Neural Networks. Classification of soil type using Backpropagation Neural Network is accurate more than general formula.

The neural networks trained to a satisfactory level with proper features and architectures perform better in prediction, pattern classification, feature extraction etc. The generalization of the network, which means producing appropriate outputs for those input samples not encountered during training process of the network, is best described by training data size and number of synaptic weights.

2. Background Theory

2.1 Neural Network

Neural network is a set of connected input/ output units where each connection has a weight associated with it. During the learning phase, the network learns by adjusting the weights so as to be able to predict the correct class label of the input sample. Neural Network learning is also referred to as connectionist learning due to the connections between units.

Neural Network involves long training times and more suitable for application where this is feasible. Neural Network have been criticized for their poor interpretability, since it is difficult for humans to interpret the symbolic meaning behind the learn weights [6]. Basic structure of neural network is shown in figure 1. In this system, 2 input nodes, 13 hidden nodes, and 1 output node and five hidden layers are used. In hidden layer, one hidden layer is used 13 nodes, two hidden layer is used 7 nodes for first layer and 6 nodes for second layer, three hidden layer is used 4 nodes for first and second layer and 5 nodes for third layer, four hidden layer is used 3 nodes for first, second and third layer and 4 nodes for fourth layer, five hidden layer is used 2 nodes for first and second layer,3 nodes for third, fourth and fifth layer.

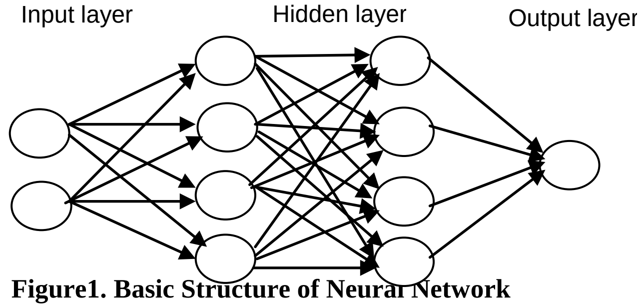

Figure1. Basic Structure of Neural Network

2.2 Network Topology

Before training can begin, the user must decide on the network topology by specifying the number of units in the input layer, the number of hidden layers (if more than one), the number of units in each hidden layer, and the number of units in the output layer. Normalization the input values for each attribute measured in the training sample will help speed up the learning phase [2].

One output unit may be used to present two classes (where the value 1 represented one class, and the value 0 to presents the other). If there are more than two classes, then one output unit per class is used. There are no clear rules as to the best number of hidden layer units [5].

Network design is a trial-and-error process and may affect the accuracy of the resulting trained network. The initial values of the weights may also affect the resulting accuracy. Once a network has been trained and its accuracy is not considered acceptable, it is common to repeat the training process with a different network topology or a different set of initial weights.

3. Backpropagation

Backpropagation is a neural network learning algorithm. The backpropagation algorithm performs learning on a multilayer feed-forward neural network. The inputs are corresponds to the attributes measured for each training sample. The weighted outputs of these units are fed simultaneously to a second layer known as hidden layer.

The hidden layer's weighted output can be input to another hidden layer, and so on. The number of hidden layers is arbitrary, although in practice, usually only on is used. The weighted output of the last hidden layer is input to the units making up the output layer, which emits the network's prediction for given sample. The units in the hidden layers and output layer are sometimes referred to as neurodes, due to their symbolic basis, or as output units [6,7].

3.1. Backpropagation Algorithm

The total net input for each neuron in the hidden layer is calculated by the following formula.

$$net_j = w_0 + \sum_{i=1}^{n} x_i w_{ij} \qquad (3.1)$$

Eg: net_j, w_{ij}, x_i

Where i is the number of input neurons.
j is the number of hidden neurons.
net_j is the total net input to hidden layer.
w_{ij} is the weight from input layer to hidden layer.
x_i is the value from the input layer.

The output value of each neuron in the hidden layer is calculate by means of sigmoid function.

$$o_j = \frac{1}{1+\exp(-net_j)} \qquad (3.2)$$

Where o_j is the output of a neuron in the hidden layer.
net_j is the total net input to hidden layer.

Each output unit calculate its error

$$\delta_j = (tj - o_j)o_j(1-o_j) \qquad (3.3)$$

Where δ_j is the each output unit of error.

Last hidden layer calculate error for each unit

$$\delta_k = o_j(1-o_j)\sum_k \delta_k w_{kj} \qquad (3.4)$$

Where δ_k is the last hidden layer of error.

Each unit of all layers update weights

$$\Delta w_{ij}(n+1) = \eta(\delta_j o_j) + \alpha \Delta w_{ij}(n) \qquad (3.5)$$

Where Δw_{ij} is the new weight for all layers [7].

4. Case Study of the System

4.1. System Design

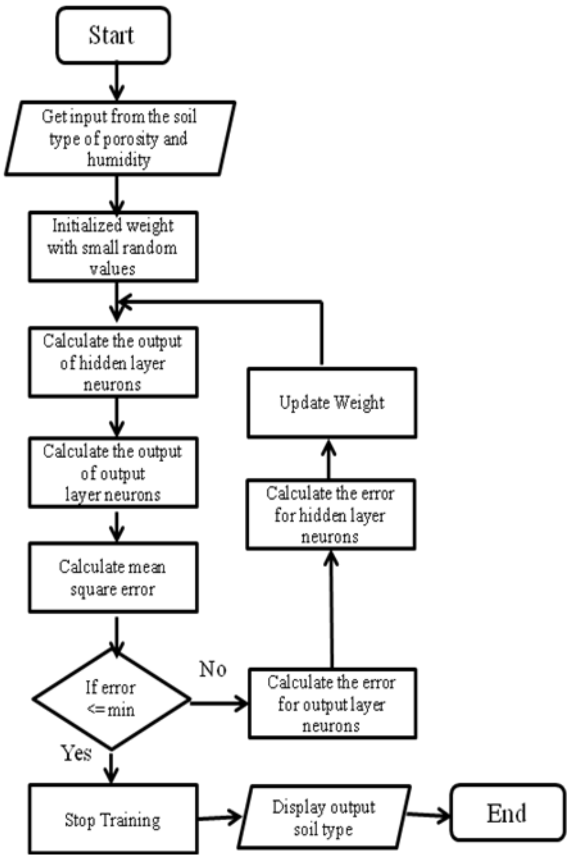

Figure 2. System Design

In this system, user can input porosity and humidity (water content) unknown class and then weight with small random values will be initialized. Calculate the output of hidden layers and output layers neurons and then calculate mean square error if error is less than 0.00001, system will classify soil type upon their void ratio. Unless the error is less than 0.00001, this system will calculate the error of output and hidden layer neurons and then will update weights. System will calculate the output of hidden layers and output layer neurons, and is less than 0.00001, the system will stop training and will classify soil type.

4.2. Implementation of the System

In this system, inputs are porosity and water content (humidity %) of soil and output will display soil type. Geotechnical (void ratio) is calculated by soil engineers and Neural (void ratio) calculated by the scientist engineers for neural network. This system is trained by five hidden layers. Hidden layer analysis will display void ratio, epoch and times(s). So the user can choose hidden layers.

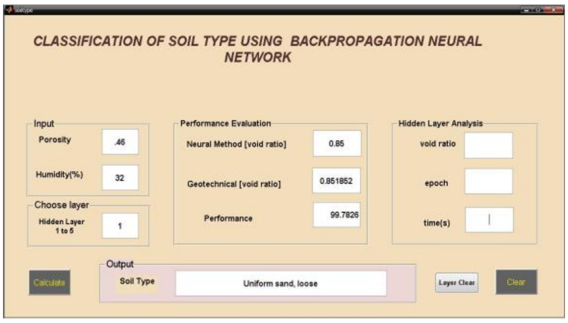

Figure 3. Classification of Soil by Hidden Layer One

In this figure, porosity of soil is 0.46 and water content of soil (humidity %) is 32% and hidden layers chosen as layer one, and calculate neural result (void ratio) and geotechnical result (void ratio). The performance evaluation is calculated of neural and geotechnical results (As the final result), output display "uniform sand, loose".

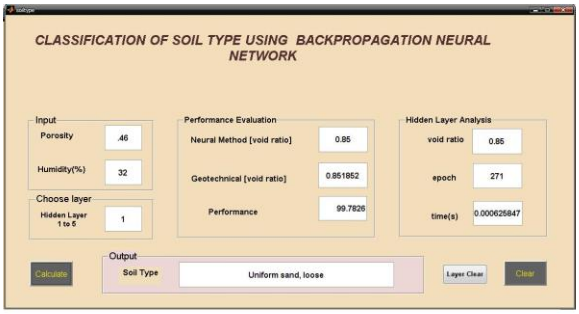

Figure 4. Hidden Layers Analysis

In this figure, hidden layers analysis will display void ratio of soil types, epoch and times of this system for neural network.

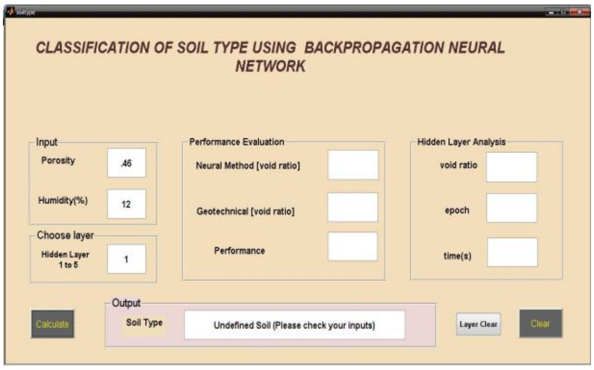

Figure 5. Wrong Inputs in Natural State

In this figure, the inputs are not match in natural state. So this system cannot classify soil type**.**

Table1. Sample for Training Dataset

No	Soil Type	Porosity	Humidity	Output	Unit Weight	
		(n)	(e)(%)	(void ratio)	γ_d	γ_{sat}
1	Uniform sand, loose	0.46	32	0.85	90	118
2	Uniform sand, dense	0.34	19	0.51	109	130
3	Mixed-grained sand, loose	0.40	25	0.67	99	124
4	Mixed-grained sand, dense	0.30	16	0.43	85	116
5	Windblown silt (loose)	0.50	21	0.99	116	135
6	Glacial till, very mixed-grained	0.20	9	0.25	132	145
7	Soft glacial clay	0.55	45	1.2	76	110
8	Stiff glacial clay	0.37	22	0.6	106	129
9	Soft slightly organic clay	0.66	70	1.9	58	98
10	Soft very organic clay	0.75	110	3.0	43	89
11	Soft montmorillonitic clay (calcium bentonitc)	0.84	194	5.2	27	80

This table is porosity, water content (humidity), void ratio and unit weight of typical soils in natural state[4]. Many training data and testing data can be used by using neural network for soil type's classification. This system is used eleven training for one soil type. So, it has totally 121 training dataset. Sample for training dataset is shown in table 1.This system is used 50 testing dataset.

4.2.1 Geotechnical Engineering

Accuracy rate is accurate when we compare with neural network classification and general formula for classification of soil type as shown in below calculation.

e= void ratio , n= porosity ,w= humidity (%)

$$e\ (geo) = \frac{nw+n-1}{(1-n)(n(w+1)-1}-1 \qquad (4.1)$$

For example
1.Uniform sand ,loose
n= 0.46 , w=32%

$$e\ (geo) = \frac{(0.46*32)+0.46-1}{(1-0.46)(0.46(32+1)-1} - 1$$

=0.851852
target output is 0.85

4.2.2. Classification of Soil Type Using Backpropagation Neural Network

Example: in one hidden layer
input are porosity (n) and humidity (w)
output is void ratio of soil type(e)
for example
1.Uniform sand ,loose
n=0.46
w=32%
e(bp)= 0.85
target output is 0.8

4.3. Performance Evaluation

a= target output (accuracy)
e(geo)= geo formula (void ratio)
e(bp) = Bp network (void ratio)

$$a = \frac{e(bp)}{e(geo)} * 100 \qquad (4.2)$$

e(geo)=0.851851851 , e(bp) =0.85

$$a = \frac{0.850998}{0.851851851} * 100$$

a=99%

5. Conclusion

This system is capable of soil classification. Soil classification is useful to track the water cycle in the Earth system, to determine the times of plant sprouting and growth, soil classification is useful to determine how much soil is available for crop, when to start planning, what kind of soil to apply, and where to irrigate. Accuracy of the system is shown in table 2.

Table2. Accuracy of the System

Accuracy for one soil type						
No	*Soil type*	Hidden layers	*Target output*	*Gen: formula*	*BP network*	*Accuracy*
			(void ratio) e	*(void ratio) e(geo)*	*(void ratio) e(bp)*	*a*
1	*Uniform sand, loose*	*1*	*0.85*	*0.851852*	*0.85*	*98.78%*
2		*2*	*0.85*	*0.851852*	*0.85*	*98.78%*
3		*3*	*0.85*	*0.851852*	*0.85*	*98.78%*
4		*4*	*0.85*	*0.851852*	*0.85*	*98.78%*
5		*5*	*0.85*	*0.851852*	*0.85*	*98.78%*

5.1. Limitation of the System

This system provides an integrated development for soil classification system. The inputs of this system will be in natural state because in this system the measured soil is in standard condition.

5.2. Further Extension of the System

Furthermore, this system can be used for water storage, nutrient uptake, water for plant use atmospheric humidity, weathering and to prevent flooding. An easy-to-use interface, this system gives the user the possibility to design particular application not requiring knowledge of Neural Network background theory.

REFERENCES

[1] Braja M.Dos, Fourth Edition: Principles of Geotechnical Engineering
[2] Bose, N.K., and P. Liang, 1998: Neural Network Fundamentals with Graphs, Algorithms and Applications. Tata McGraw-Hill, pp 478.
[3] Haykin, S., 1999: Neural Networks- A Comprehensive Foundation. Addison Wesley Longman.
[4] I.S Dunn, L.R Anderson, F.W Kiefer, 1980: Fundamental of Geotechnological Analysis
[5] Mathur, S., Shukla, A.K., and Pant, R.P., Sept. 22-23, 2001, International Conference on Optimization Techniques and its Applications in Engineering and Technology, A Comparative Study of Neural Network and gression Models for Estimating Stock Prices.
[6] Paras, Sanjay Mathur, Avinash Kumar, and Mahesh Chandra, A Feature Based Neural Network Model.
[7] Robet Callen: The Essence of Neural Networks. Prentice Hall Europe.
http://www.solent.ac.uk/syseng/faculty/html/staff/rcallan/.