Artificial neural networks in predicting forest cover types from cartographic variables

Zhanhui Lin

Abstract. The study evaluated four wilderness areas in the Roosevelt National Forest, located in the Front Range of northern Colorado. We use a series of multi-layer perceptron (deep neural networks model) predicted forest cover type. Compared with the first assignment, I increased the depth of the neural network, adopted a new activation function(Relu), added Softmax function in the output layer and adopted batch in the training, which greatly improved the training speed of the network. The final accuracy rate on test data increased by 14% compared to the previous one. Also got closer to the performance in the paper published.

Keywords: MLP, deep neuron network, batch, Relu, forest cover type, prediction

1 Introduction

Accurate natural resource inventory information is vital to any private, state, or federal land management agency. Forest cover type is one of the most basic characteristics recorded in such inventories. Generally, cover type data is either directly recorded by field personnel or estimated from remotely sensed data. Both of these techniques may be prohibitively time consuming and:or costly in some situations. Furthermore, an agency may find it useful to have inventory information for adjoining lands that are not directly under its control, where it is often economically or legally impossible to collect inventory data. Predictive models provide an alternative method for obtaining such data.

Artificial neural networks are ‘‘computing devices that use design principles similar to the design of the information-processing system of the human brain’’ [1]. Several recent textbooks describe the mechanics of ANNs [2,3,4,5,6]. Recent publications involving artificial neural networks being applied to natural resources topics include: modeling complex biophysical interactions for resource planning applications [7]; generating terrain textures from a digital elevation model and remotely sensed data [8]; modeling individual tree survival probabilities [9]; and Harvey and Dean [10], who used geographic information systems (GIS) in developing computer-aided visualization of proposed road networks. Recent comparisons in which ANNs performed favorably against conventional statistical approaches include Reibnegger et al. [11], Patuwo et al. [12], Yoon et al. [13], Marzban and Stumpf[14], Paruelo and Tomasel [15], Pattie and Haas[16], and Marzban et al. [17].

However, artificial neural networks do not always outperform traditional predictive models. For example, Jan [18] found a traditional maximum-likelihood classifier outperformed artificial neural network models when classifying remotely sensed crop data.

This study examined the ability of an ANN(multi-layer Perceptron) model to predict forest cover type classes in forested areas that have experienced relatively little direct human management activities in the recent past. The predictions produced by the MLP model were evaluated based on how well they corresponded with observed cover types (absolute accuracy).

2 Data Set and Model Design

2.1 Data Set

We download from http://archive.ics.uci.edu/ml/datasets/Covertype.

The seven forest cover type classes used in this study were lodgepole pine (Pinus contorta), spruce/fir (Picea engelmannii and Abies lasiocarpa), ponderosa pine (Pinus ponderosa), Douglas-fir (Pseudotsuga menziesii), aspen (Pop-ulus tremuloides), cottonwood/willow (Populus angustifolia, Populus deltoides, Salix bebbiana, Salix amygdaloides), and krummholz.

Both soil type information and wilderness area designation were obtained from the USFS. These qualitative variables were treated as multiple binary values. This resulted in a series of variables for each raster cell, where a value of '0' would represent an 'absence' and a value of '1' would represent a 'presence' of a specific wilderness area or soil type. A total of four wilderness areas and 40 soil type classes were used in this study, producing four wilderness area designator variables, forty soil type designator variables, and ten continuous variables for a total of 54 possible independent variables available for each model.

Headings. For this study, three mutually exclusive and distinct data sets were created to train, validate, and test the predictive models. A training data set was used to develop classifiers for both the artificial neural network and the discriminant analysis predictive models. Number of observations for each data set are given in Table 1.

The first extracted set contained 348607 randomly selected observations (60% of total observations) and became the training data set.

The second data set extracted from the remaining data contained 116202 randomly selected observation (20% of total observations) and became the validation data set.

All variables in the three data sets used by the artificial neural network model were linearly scaled to lie in the range between zero and one.

Table 1. Number of observations for each data set

Train data set size	validation data set size	Test data set size
348607	116202	116203

2.1 Model Design

loss function(MSE):

$$E(w) = \frac{1}{2N} \sum_{n=1}^{N} \sum_{i=1}^{k} (d_i(n) - y_i(n))^2 \quad (1)$$

where E(w) is the mean square error term, w are the synaptic weights to be estimated, N is the number of observation (input) vectors presented to the network, n is a single observation vector, k is the number of output nodes, i is a single output node, d_i (n) is the observed response and y_i (n) is the predicted response for observation n and output node i. The N observation vectors constitute a training data set, which is used specifically to 'teach' the network to recognize the relationships between the independent and dependent variables (e.g. to develop a classifier). This classifier will consequently be used to predict class membership for other vectors of input variables not included in the training data set. Theoretically, the back-propagation algorithm ultimately finds a set of weights w that minimizes E(w).

All artificial neural network models in this study had fully connected input, hidden, and output layers (i.e. each node in layer m was connected to all nodes in layer m+1). The generalized delta rule with gradient descent (commonly used with the backpropagation learning algorithm) was utilized in each network's learning process. The activation function for each network's input layer was linear [f(x)= x], while hidden utilized Relu activation functions, and output layers utilized softmax functions. Adam was used as the optimization algorithm. Initial synaptic weights were randomly selected between negative one and positive one, based on a random seed and no input noise. All input variables were linearly scaled to lie in the range between zero and one.

Training patterns were presented to the network in a random order, with an update of the validation data set MSE at an interval of every ten epochs through the training data set. Training was halted after either

(1) a minimum of 100 training epochs had been completed

(2) a validation MSE of 0.05 was reached, or

(3) it was subjectively determined that the validation MSE would not significantly decrease with further training epochs.

Two hidden layers were used in all of the artificial neural networks developed in this study. The numbers of nodes in these two hidden layers were systematically changed across 8 possible values while holding constant the learning rate (LR). The

Relu activation function was adopted and batch method was used in the training process.

This procedure identified the number of hidden nodes which produced the minimum error (MSE) of the validation data set under the original LR. The number of hidden nodes was then held constant and the LR parameter were sequentially changed to find the best combination of network architecture and training parameter values for this data set. Fig 1 shows the 2 hidden layers neuron netwotk. Table 2 lists the various architectural and training parameter values investigated in this study.

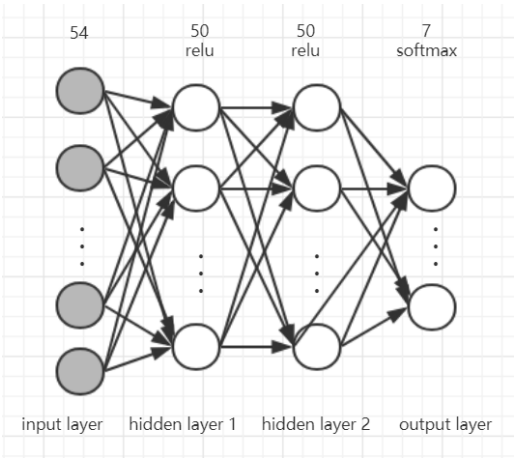

Fig. 1. The construction of the artificial neuron network.

Table 2. Artificial neural network architectural and training parameter values

Step 1: Select the optimal number of hidden nodes parameter from 8 possible values while holding constant LR=0.01								
Hidden1 nodes	10	20	30	50	100	150	200	300
Hidden2 nodes	10	20	30	50	100	150	200	300
Step 2: Hold the optimal number of hidden nodes parameter value (selected from step 1) constant, and determine the optimal learning rate (LR)								
Learning rate	0.0005	0.001	0.005	0.01	0.05	0.1	-	

3 Results and Discussion

The MSE values across all 8(the published paper is 14. Because in our experiment, 8 were selected and we achieved the best at around (50,50) epoch) different numbers

of hidden nodes from the 54 variable ANN models are shown in Fig. 1. Each of these networks held the LR at 0.01, respectively. This figure shows that roughly 50 hidden 1 nodes and 50 hidden 2 nodes (the published paper is 120 of only one hidden layer) were necessary to minimize the MSE in these networks.

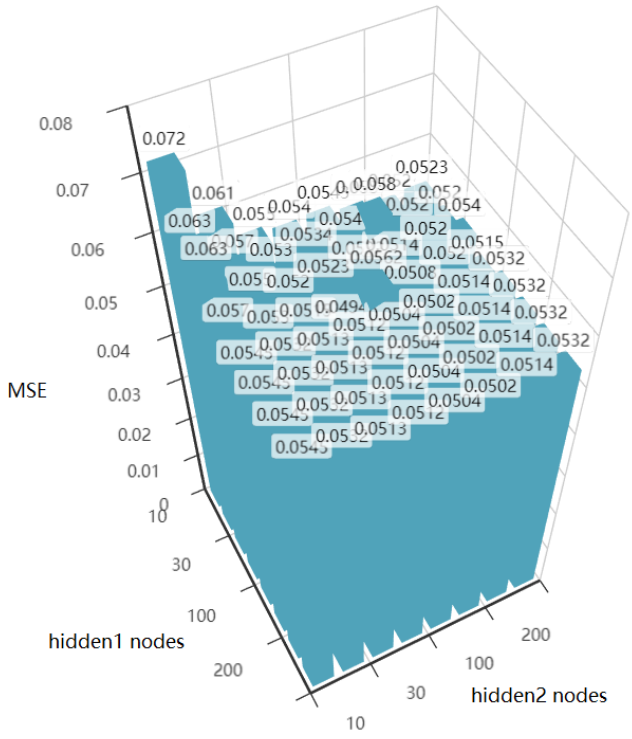

Fig. 2. MSE for the different numbers of hidden nodes for the 54 variable ANN models (optimal value for this model is 50 hidden 1 nodes and 50 hidden 2 nodes).

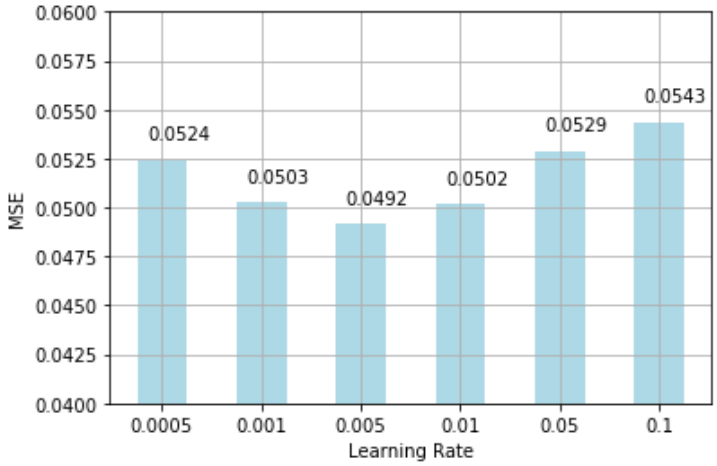

Fig. 3. MSE of the validation data set for the 54 variable ANN models with 50 hidden 1 nodes and 50 hidden 2 nodes (optimal values for this model are a learning rate of 0.005). MSE values are shown by learning rates (symbols) and are equal along concentric lines of the plot.

Once the 'best' number of hidden nodes was identified, the learning rates and momentum rates were sequentially changed to determine optimal values for these parameters. Fig. 3 displays a plot of the resulting MSE values for 6 different candidate networks, all with 50 hidden 1 nodes and 50 hidden 2 nodes. And we found the network model with the lowest MSE was produced with an LR of 0.005.

As shown in Table 3, the MLP predictions of forest cover type produced an overall classification accuracy of 78.5221%. (the best model, LR=0.005, hidden 1 nodes=50, hidden 1 nodes=50)

In comparison, Table 4 presents the XGBoost algorithm results obtained from the test data set. The overall classification accuracy for the XGBoost algorithm model was 87.0029%.

Table 3. MLP matrix for the test data set

		predict label						
MLP	observed	33140	8629	0	0	84	11	1070
		8457	46570	319	0	1279	293	21
		21	771	6221	257	71	1505	8
		3	2	95	253	0	36	0
		41	123	0	0	459	0	0
		36	725	443	26	40	1604	0
		520	75	0	0	0	0	2998

Table 4. XGBoost algorithm matrix for the test data set

		predict label						
Xgb model	observed	35472	6519	2	0	24	6	195
		5458	50897	211	0	117	192	20
		2	256	6484	46	1	286	0
		0	0	61	455	0	20	0
		17	667	43	0	1190	16	0
		8	193	356	22	0	2870	0
		350	14	0	0	1	0	3732

4 Conclusion and Future Work

Compared with the first assignment, the new deep neuron network and new activation function greatly improved the performance of the network. And the batch method in the training method make the training speed much faster than Assignment 1. The final accuracy rate on test data increased by 14% compared to the previous one. From the experimental results, However, XGboost algorithm in the paper is still better than MLP, to improve the model through:

1. Try other neuron network structures.
2. Try other loss function, such as cross entropy.

$$cross\_entropy = \sum_{x} p(x) log(\frac{1}{q(x)})$$

3. Add dropout to deal with overfitting.
4. Add more training data.

References

1. Bharath R, Drosen J. Neural network computing[C]// 1994.
2. Hertz J, Krogh A, Palmer R G. Introduction to the theory of neural computation. [J]. American Journal of Physics, 1991, 62(1):27-29.
3. Haykin S, Network N. A comprehensive foundation[J]. Neural Networks, 2004.
4. Masters T. Signal and image processing with neural networks: a C++ sourcebook[M]// Signal and image processing with neural networks: J. Wiley, 1994:5724-5724.
5. Bishop, ChristopherM. Neural networks for pattern recognition[M]. Oxford University Press, 1995.
6. Ripley B D. Pattern Recognition and Neural Networks[J]. Technometrics, 1999, 39(2):233-234.

7. Gimblett R H, Ball G L. Neural network architectures for monitoring and stimulating changes in forest resource management[J]. Ai Applications, 1995, 9(2):103-123.
8. Alvarez S. Generation of terrain textures using neural networks[J]. 1995.
9. Guan B T, Gertner G Z. Modeling individual tree survival probability with a random optimization procedure: an artificial neural network approach. [J]. Ai Applications, 1995, 9(2):39-52.
10. Blackard J A, Dean D J. Comparative accuracies of artificial neural networks and discriminant analysis in predicting forest cover types from cartographic variables[J]. Computers and electronics in agriculture, 1999, 24(3): 131-151.
11. Reibnegger G, Weiss G, Werner-Felmayer G, et al. Neural networks as a tool for utilizing laboratory information: comparison with linear discriminant analysis and with classification and regression trees[J]. Proceedings of the National Academy of Sciences, 1991, 88(24): 11426-11430.
12. Paruelo J M, Tomasel F. Prediction of functional characteristics of ecosystems: a comparison of artificial neural networks and regression models[J]. Ecological Modelling, 1997, 98(2-3): 173-186.
13. Yoon Y, Swales Jr G, Margavio T M. A comparison of discriminant analysis versus artificial neural networks[J]. Journal of the Operational Research Society, 1993, 44(1): 51-60.
14. Marzban C, Stumpf G J. A neural network for tornado prediction based on Doppler radar-derived attributes[J]. Journal of applied meteorology, 1996, 35(5): 617-626.
15. Paruelo J M, Tomasel F. Prediction of functional characteristics of ecosystems: a comparison of artificial neural networks and regression models[J]. Ecological Modelling, 1997, 98(2-3): 173-186.
16. Pattie D C, Haas G. Forecasting wilderness recreation use: neural network vs. regression[J]. AI applications, 1996, 10(1): 67-74.
17. Marzban C, Paik H, Stumpf G J. Neural networks vs. gaussian discriminant analysis[J]. AI applications, 1997, 11(1): 49-58.
18. Jan J F. Classification of remote sensing data using adaptive machine learning techniques[D]. Colorado State University, 1997.