

An initial study of deep learning for mangrove classification

S Faza¹, E B Nababan^{2*}, S Efendi², M Basyuni³, R F Rahmat²

¹Student in Faculty of Computer Science and Information Technology, Universitas Sumatera Utara, Medan, Indonesia

²Faculty of Computer Science and Information Technology, Universitas Sumatera Utara, Medan, Indonesia

²Faculty of Forestry, Universitas Sumatera Utara, Medan, Indonesia

*ernabrn@usu.ac.id

Abstract.

Deep Learning is a new breakthrough in the area of neural network. One of its methodology is Deep Neural Network. Usually, deep neural network is a method that can solve problems such as classification or prediction. Here in this study, we collect data for mangrove classification, and we see an opportunity to classify the type of mangrove based on its features using deep neural network. Research on mangrove plants related to classification is more widely used in mangrove dispersal spread through spectral and hyperspectral satellite images than using real value data such as morphological data, thus providing blank space for researchers to use mangrove morphological data. This initial study is to build a deep neural network architecture and analyze it in terms of mangrove sprout plant classification. The result from our architectural methodology of this research is reaching the lowest training error of 0.1345 and the highest testing accuracy value of 98%.

1. Introduction

Deep Learning is a new advancement in the area of neural network [1]. Deep Learning is a method in machine learning that has more "deep" architecture than other methods of learning to solve a problem in classification or prediction [2]. Classification is the determination or identification of data to enter in a particular class. Data classification uses two phases, the first phase of the training phase and the testing phase.

Research that has utilized deep learning in classification problem, one specific research from Krizhevsky shows that the classification of images using Convolutional Neural Network (CNN) yields the smallest error rate compared with six sparse coding model and Fisher Vectors (FVs) model which is 15.3% [3], CNN also has been applied before in classifying Korean Letter and resulting proper classification [4]. Also, Hassan & Mahmood classified the sentence using the Long Short-Term Memory (LSTM) Recurrent Neural Network (LSTM), which resulted in the smallest error rate compared to some models in the study of 11.32% [5].

Mangrove plant is one of protected ecological habitat nowadays; they need particular attention it has implication in the coastal area to prevent erosion [6]. Research on mangrove-associated with classification is more widely used in mangrove dispersal deployment through spectral and hyperspectral satellite images [6] [7] [8] than using real value data such as morphological data, thus providing blank spaces for researchers to use mangrove morphological data.



Other machine learning method has been proposed before, such as DANGLE [9] and evolving connectionist [10] will give a good result but not so well in term of performance if compared to deep neural network. Because of this, in this study, we will use other deep learning methods, namely Deep Neural Network (DNN) method for mangrove morphological data classification.

2. Methodology

The method used in this research consists of several stages, they are pre-processing data phase by performing data transformation, and the determination of attributes. Then the next step is to classify using Deep Neural Network (DNN) method.

2.1. Data pre-processing

The pre-processing stage of data is a data selection stage that aims to get data that is suitable to use. The data used in this study is real value data from the morphology of mangrove sprout plants that are aged 3-6 months that numbered 1200 datasets.

2.1.1. Types of attributes

Several attributes can be used in the classification of mangrove species in this study. The attributes used in the data of mangrove sprout plant are as follows: salinity (a), height (b), wet root weight (c), wet weight of stem (d), wet weight of leaf (e), root dry weight (f), dry weight of stem (g), leaf dry weight (h), and number of leaves (i).

2.1.2. Data Transformation

Transformation of data is done to get the overall value on a scale of 0 to 1 so that the analysis process is easier to process, with the following formula:

$$\text{new_value} = \frac{x - \text{min_value}}{\text{max_value} - \text{min_value}} (\text{range_max} - \text{range_min}) + \text{range_min} \quad (1)$$

Where range min = 0, and range max = 1.

Here we described original data before the transformation occurred:

Table 1. Original data samples

No	(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)	(i)
1	3	1	0.4	0.1	1	0.25	0.05	0.1	4
2	1.5	3.6	0.8	1.2	1.1	0.4	0.15	0.7	6
3	0.5	5.2	0.6	0.1	1.7	0.35	0.05	0.5	6
4	0.5	1.2	0.6	0.1	1.7	0.35	0.05	0.5	3
5	0	0.5	0.1	0.1	0.5	0.01	0.05	0.25	2
6	1.5	3.2	0.8	1.2	1.1	0.4	0.15	0.7	4
7	2	2.6	0.3	0.1	1.3	0.25	0.05	0.8	4
8	2	3	0.3	0.1	1.3	0.25	0.05	0.8	6
:	:	:	:	:	:	:	:	:	:
1200	3	2	0.4	0.1	1	0.25	0.05	0.1	4

In Table 1, we can see each attribute has a different scale; therefore we need to transform it using equation 1. The result of the data transformation is described in table 2.

Tabel 2. Transformed data samples

No	(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)	(i)
1	1	0.1	0.4	0	0.4	0.62	0	0	1
2	0.5	0.7	1	1	0.5	1	1	0.86	1
3	0.2	1	0.7	0	1	0.87	0	0.57	1
4	0.2	0.2	0.7	0	1	0.87	0	0.57	0
5	0	0	0	0	0	0	0	0.21	0
6	0.5	0.6	1	1	0.5	1	1	0.86	1
7	0.7	0.5	0.3	0	0.7	0.62	0	1	1
8	0.7	0.5	0.3	0	0.7	0.62	0	1	1
:	:	:	:	:	:	:	:	:	:
1200	1	0.3	0.4	0	0.4	0.62	0	0	1

In table 2, each attribute is transformed into a 0-1 scale of value, and this process will smooth then the next process to be performed.

2.2. Data Processing

Data that has been normalized and transformed will be divided into two parts, training dataset, and testing dataset. The test and training phase is divided into two stages; the first stage is unsupervised training with Deep Believe Network (DBN) which has Restricted Boltzmann Machine (RBM) stacked in every layer and do the fine tuning (supervised learning) [11]. The second stage is performing back propagation to the fine-tuned DBN; those two processes will create what we call DNN. DNN will be tested using testing dataset.

2.2.1. Training deep neural network

The training phase of this research is divided into two stages, the first stage of unsupervised training with DBN which has RBM stack. The next stage will be transformed into DNN by applying a back propagation (supervised learning) algorithm. In this study, the DNN architecture used consists of 9 neuron inputs, 2 hidden layers with 10 neurons in each hidden and 3 neuron outputs. The activation function used is the sigmoid activation function and the softmax activation function at the output layer. Parameters used such as weight and bias will be randomly initialized, and the learning rate used is 0.1.

The first phase training of DBN is training RBM using Gibbs sampling with four stages for binary type data [12], receiving real value data on a visible layer and different visible units can use Gaussian Unit or Gaussian Bernoulli RBM [13].

1. Assumed that the visible layer in RBM as v with the layer marker as h .
2. Positive Phase, update all hidden units.

$$P(H_j = 1|V) = f\left(B_j + \sum_{i=1}^m W_{ij} \frac{v_i}{\sigma_i^2}\right) \quad (2)$$

Where P is the probability, H is the hidden unit, V is the visible unit, f is activation function, B is bias, W is weight, and σ standard deviation.

3. Negative Phase. Conducting visible unit reconstruction by using equation 3.

$$P(V_i = V|H) = N(V|C_j + \sum_{i=1}^m W_{ij} H_i, \sigma_i^2) \quad (3)$$

Where N function is Gaussian probability density with Mean and Standard deviation.

4. Update the weight of each edge.

$$Updt(W_{ij}) = W_{ij} + L * (Positive(E_{ij}) - Negative(E_{ij})) \quad (4)$$

5. Repeat the stages until the termination criteria are met.

After the first phase of the training is completed, the results of the parameters obtained, such as weight and bias, will be used in the second training phase, with the addition of discriminant layer at the top layer of RBM as the output layer, then change the connection from two directions to the direction. In this research apply back propagation algorithm.

2.3. Performance Evaluation

The result of deep neural network training will be evaluated using Cross Entropy Error. Calculation of cross-entropy error can be seen in the following formula.

$$E = -\sum_i^{n_{class}} t_i \log(y_i) \quad (5)$$

Where t is the target vector and y is the output vector. In addition to evaluating the performance of deep neural network training by calculating the actual error, the results of deep neural network testing can be calculated the percentage of its accuracy also to see how well the classification. Calculation of the percentage of accuracy can be seen in the following formula.

$$accuracy (\%) = \left[\frac{\text{the number of data on the target}}{\text{number of data}} \right] * 100\% \quad (6)$$

3. Results and Discussion

In this chapter will discuss the results obtained such as an error in the training process, and accuracy in the testing process using the Deep Neural Network (DNN) method.

3.1. Training Performance Results

Training conducted using 1100 dataset with epoch maximum 300. It is shown in Table 3 below.

Table 3. Results of Training Performance

Epoch	Cross-Entropy Error
50	0.7378
100	0.5318
150	0.3948
200	0.3185
250	0.2309
300	0.1345

From the Table 1, it can be seen that more epoch will give less error, as in the 50th epoch error 0.7378, the epoch to 100 error is reduced to 0.5318, and so on until the epoch maximum (epoch to-300) error obtained for 0.1345. Changes in errors from the 50th epoch to the 300th epoch can be seen in graphs, graphics results of deep neural network training error can be seen in Figure 1.

From the figure 1, can be seen the number of errors decreased as the epoch more and more. The result suggests the Deep Neural Network method can perform the introduction of data during training, and the more training made, the smaller error.

3.2. Testing Performance Results

Testing conducted using 100 different datasets with epoch maximum 300 also. Training result can be seen in Table 4.

Table 4.Results of Testing Performance

Epoch	Accuracy
50	82%
100	88%
150	92%
200	97%
250	97%
300	98%

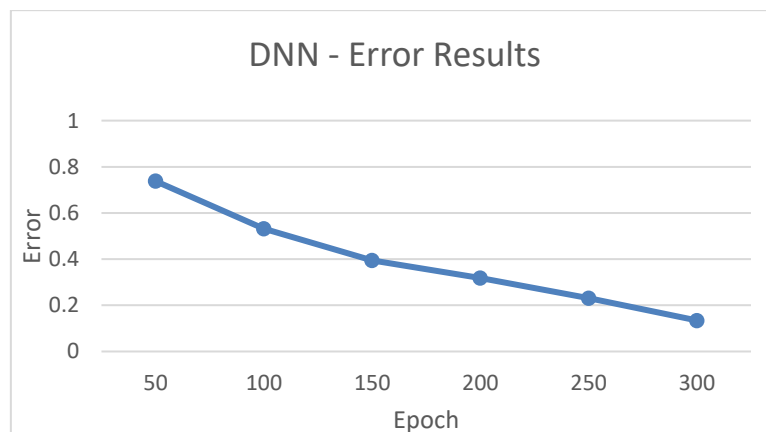


Figure 1. DNN Error Result

From the table 2, the results of the deep neural network can classify the data with an accuracy of 82% in the 50th epoch, the accuracy rate at the epoch to 100 to 88%, and increasing to the 300th epoch maximum to 98%. The above accuracy results can be seen in the graph in Figure 2.

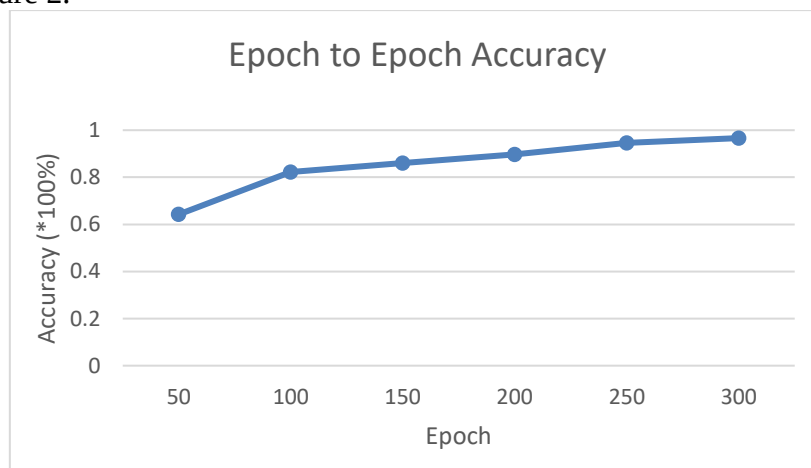


Figure 2. DNN Accuracy Result

From the figure 2 above, it can be seen that the accuracy level is getting up to 100% when the epoch is more, that is, the 300th epoch maximum accuracy is 98%. This result shows and proving that the deep neural network method can classify mangrove data well if more epoch is done.

4. Conclusion

From the previous chapters that addressed the error during training and the accuracy of the test results, the deep neural network method was able to classify the morphological data of the mangrove sprouts well. The results show that Deep Neural Network method gets the value of the lowest training error of 0.1345 and the highest accuracy is at 98% which is obtained in the 300th epoch.

5. References

- [1] Heaton, J. 2015. *Artificial Intelligence for Humans Volume 3: Deep Learning and Neural Networks*. Heaton Research, Inc, : Chesterfield.
- [2] Bengio, Y. 2009. *Learning Deep Architectures For AI*. Now Publisher Inc. :USA
- [3] Krizhevsky, A., Sutskever, I. & Hinton, G.E. 2012. Imagenet classification with deep convolutional. *Proceedings of the 25th International Conference on Neural Information Processing Systems*, Vol 1, pp. 1097-1105.
- [4] Purnamawati, S., Rachmawati,D., Lumanauw, G., Rahmat, R.F.,&Taquyuddin, R. 2018. Korean letter handwritten recognition using deep convolutional neural network on android platform.*Journal of Physics: Conference Series*978(1),012112
- [5] Hassan, A. & Mahmood, A., 2017. Deep learning for sentence classification. IEEE. *Long Island Systems, Applications and Technology Conference (LISAT)*.
- [6] Geng, J., Fan, J., Su, X., Ma, X. & Wang, H. 2015. Spectral-spatial information extraction and classification of mangrove species using joint sparse representation. IEEE. *4th International Conference on Computer Science and Network Technology (ICCSNT)*.
- [7] Chakravortty, S. & Shah, E. 2013. Application of non-linear spectral unmixing on hyperspectral data for species level classification of mangroves. IEEE. *International conference on Communication and Signal Processing*.
- [8] Chakravortty, S. & Sinha, D. 2014. Performance of pure pixel extraction algorithms on hyperspectral data for species level classification of mangroves. IEEE. *4th International Conference of Emerging Applications of Information Technology*.
- [9] Rahmat. R.F., Pasha, M.F., Syukur, M., & Budiarto, R. 2015. A gene-regulated nested neural network.*International Arab Journal of Information Technology*12 pp. 532–539.
- [10] Nababan, E.B., Iqbal, M., & Rahmat, R.F. 2016. Breast cancer identification on digital mammogram using evolving connectionist systems.*International Conference on Informatics and Computing (ICIC)*pp. 132–136.
- [11] Bai, Y., Li, C., Sun, Z., & Chen, H. 2017. Deep neural network for manufacturing quality prediction. *Prognostics and System Health Management Conference (PHM-Harbin)*.
- [12] Hinton, G. 2010. *A Practical Guide to Training Restricted Boltzmann Machines*, Technical Report, University of Toronto
- [13] Yamashita, T., Tanaka, M, Yoshida, E. & Fujiyoshi, H. 2014. To be Bernoulli or to be Gaussian, for a restricted Boltzmann machine. IEEE. *22nd International Conference on Pattern Recognition*