Modified weight initialization in the self-organizing map using Nguyen-Widrow initialization algorithm

M N Linan, B Gerardo, R Medina

Graduate Programs, Technological Institute of the Philippines, Quezon City, Philippines
College of ICT, West Visayas State University, Lapaz, Iloilo City, Philippines

E-mail: maureenettie@gmail.com

Abstract. Self-organizing map (SOM) is a neural network trained using unsupervised learning for clustering. The clustering result and learning speed of SOM, however, is dependent on the initial weights which are randomly initialized with a low close to zero value from the range of vectors within the given input space data negatively affecting training speed and clustering result. The study used the Nguyen-Widrow initialization algorithm to initialize the weights of SOM and speed up the training process. The cluster error rate and the number of iterations to converge to final clustering is recorded and compared with the traditional SOM to determine the performance of the modified SOM. The result reveals that the modified SOM algorithm produces better cluster results and improved training speed as compared to traditional SOM. Hence, Nguyen-Widrow algorithm for initialization of weights in the SOM yields better cluster result and improved training speed of the algorithm in terms of the number of iterations.

1. Introduction

Clustering is an essential analytical method in data mining wherein objects within a data set are grouped into clusters of objects with high similarity. Among clustering algorithms, the self-organizing map (SOM), an artificial neural network model, has found wide application in industry, finance, natural sciences, and linguistics. This stems from the ability of SOM to represent multidimensional data in much lower dimensional space – usually in one or two dimensions [1].

Clustering performance of SOM, however, is greatly dependent on the initial weight initialization. Weights in SOM are randomly initialize with a low close to zero value from the range of input vectors within the given space. With this, the quality of clustering results together with learning speed is greatly affected [2, 3]. Various studies have been conducted to address weight initialization. The study of Haripriya, et al. [4] works on the different weight initialization strategies. The initialization strategies are applied to datasets before dimensionality reduction and after dimensionality reduction as well. Results show that even after dimensionality reduction, the performance of the initialization techniques applied is dependent on the linear or nonlinear properties of the datasets. In the work of Akinduko, et al. [5], datasets were separated into two classes: the quasilinear and the essentially nonlinear datasets. Using the principal component initialization (PCI) for weights initialization, results reveal that PCI performs well on quasilinear datasets but not on nonlinear datasets. Another study used the Frequency Sensitive Competitive Learning (FSCL) algorithm [3] to pre-process the weights in order to improve the results in terms of better neuron utilization and less quantization and topographic error. The algorithm achieves a better performance on a distributed dataset, however, the algorithm did not achieve a significant result on unevenly distributed datasets. Dogan, et al. [6] showed that SOM++ has good
performance in stability and outperforms SOM, K-means and K-means+SOM training time. This approach has to perform the K-means++ method prior to determining the initial weight values and the starting points. The self-organizing map is then applied to determine the final clustering result. Furthermore, the Valova, et al. [2] considers placing the neurons in the Hilbert curve to prevent tangling of networks as SOM has the tendency to converge similarly to self-similar curves. Results reveal Hilbert curve initialization is the best tool for the initial placement of neurons. The neighborhood size, as well as the mode of initialization, proved to be the most important when it comes to quality coverage with evenly distributed neurons. However, optimization of the neighborhood size and the learning rate decay must be addressed.

To address the weights initialization issue, the Nguyen-Widrow algorithm was used in this study. Nguyen-Widrow initialization algorithm is a method for initialization of the weights of neural networks to speed up the training process. Initial weight is set up so that each hidden node is assigned to approximate a portion of the range desired function at the start of the training. In this approach, the weights connecting the output units to the hidden units are initialized with small random values over the interval (-0.5, 0.5) [8,9]. A study on image compression using a multilayer feed-forward artificial neural network based on Nguyen-widow weight initialization algorithm was proposed by Mishra, et al. [10]. All weights in the network are adjusted in an identical manner using Nguyen-Widrow, and thus prevent the error function from being reduced. Using Nguyen-Widrow in artificial neural network improved the execution training time and peak-signal-to-noise (PSNR). Furthermore, results obtained in the study conducted by Pavelka & Procházka [11], provided a noteworthy evidence on the superiority of Nguyen-Widrow’s initialization algorithm against random initialization techniques. The Nguyen-Widrow method generates initial weights and bias values for a layer so that active regions of the layer neurons will be distributed evenly over input space. This speeds up the training process by setting the initial weights of the first layer so that each node is assigned its own interval at the start of the training.

Hence, this paper introduces the Nguyen-Widrow initialization algorithm approach in the SOM. The Nguyen-Widrow weight initialization algorithm is used to initialize the weights in the SOM. Performance results in terms of error rate and number of iterations to converge to final clustering are then compared.

2. Materials and Methods
2.1. Self-organizing Map (SOM)
SOM algorithm is an unsupervised learning algorithm and it is widely used for clustering large sets of data. SOM has only two layers: the input layer and the output layer. The input layer is one-dimensional, while the output layer consists of radial units typically organized in one or two-dimensions. In the input layer, each data item was associated with an n-length vector of elements. The map, as shown in Figure 1, is an array of nodes called neurons. Each node is a vector of N weights [3].

![Figure 1. SOM architecture](image-url)
The first step in SOM training is to randomly initialize the weight vectors $w_i$ of the $m \times n$ neurons. Input vector $x(t)$ is then randomly selected. Next is the competition among the neurons to find the winner using Euclidean function (1):

$$c = \min_{1 \leq i \leq mn} \{ ||w_i(t) - x(t)|| \}$$  \hspace{1cm} (1)

where $x(t)$ and $w_i(t)$ are the input and weight vector of neuron $i$ at iteration $t$ respectively. This winner is called best matching unit (BMU) of the input data pattern. Then, in the cooperation step, spatial neighbors of the winning neuron are selected. Finally, the process of weight adjustments using equation (2):

$$w_i(t + 1) = w_i(t) + h_{c,i}(t)[x(t) - w_i(t)]$$  \hspace{1cm} (2)

where $h_{c,i}(t)$ is a Gaussian neighborhood function:

$$h_{c,i}(t) = \alpha(t).\exp\left( -\frac{||r_c-r_i||^2}{2\sigma^2(t)} \right)$$  \hspace{1cm} (3)

where $r$ is the coordinate position of the neuron on the map, $\alpha(t)$ is the learning rate and $\sigma(t)$ is the width of the neighborhood radius. Both $\alpha(t)$ and $\sigma(t)$ are decreased monotonically using the following equations:

$$\alpha(t) = \alpha(0) \left( \frac{\alpha(T)}{\alpha(0)} \right)^{t/T}$$  \hspace{1cm} (4)

$$\sigma(t) = \sigma(0) \left( \frac{\sigma(T)}{\sigma(0)} \right)^{t/T}$$  \hspace{1cm} (5)

where $T$ is the training length. For all the input data, the same process is repeated from the random selection of input to weight adjustments [12].

Clustering of data using the SOM algorithm is achieved mainly by two steps: 1. training the data with the initialized parameters, and 2. clustering of data. SOM training results depend on the initialization of weight values for neurons in the SOM map. The final values of these weights are then used for clustering [3].

2.2. SOM with Nguyen-Widrow initialization algorithm

In this paper, the Nguyen-Widrow initialization algorithm was introduced to initialize the weights. Input vectors were randomly selected and winning neuron, also called as the Best Matching Unit (BMU) was determined. The winning neuron was determined using Euclidean distance. Weights are then updated. The process is repeated until all the vectors trained. The SOM with Nguyen-Widrow initialization algorithm is presented in table 1.

To achieve better clustering result and to avoid the negative effects produced by noise and outliers, the data sets were pre-processed using data cleaning normalization schemes. The data were normalized using the equation (8):

$$X = \frac{x_i - \min(x_r)}{\max(x_r) - \min(x_r)}$$  \hspace{1cm} (8)

where $X$ is the raw data, $\min(X)$ the smallest value in $X$, and $\max(X)$ the largest value in $X$. 


Table 1. SOM with Nguyen-Widrow initialization algorithm.

| ALGORITHM: SOM algorithm with Nguyen-Widrow initialization algorithm |
|---------------------------------------------------------------------|
| 1. Initialization of weights using the Nguyen-Widrow initialization algorithm [9] |
| a. All weights \( w \) of hidden layers with random values over the interval (-0.5 to 0.5) were initialized |
| b. For each hidden layers, beta value \( \beta \) using equation (6) was calculated |
| \[ \beta = 0.7 \ast H(1/n) \] (6) |
| c. For each synapse |
| i. For each weight |
| Weight, \( w_i \) was adjusted by dividing it with the norm of weight for neuron and multiplying beta value using the equation (7) |
| \[ w_i = \beta \ast \frac{w_i \text{(random)}}{\|w_i \text{(random)}\|} \] (7) |
| 2. An input vector \( x(t) \) was selected randomly |
| 3. “Best Matching Unit (BMU)” \( c \), using the Euclidian distance formula (1) was determined. |
| 4. The weight vector of the neurons using equation (2) was updated. |
| 5. Steps 2 to 5 were repeated for the input data. |

With modified SOM, the weights generated with Nguyen-Widrow initialization was used as the initial weights for training data. Clustering is then performed using the final weights as the initial weights.

In the study, data was trained using 1-dimensional neurons at a learning rate of 0.5. Two datasets comprised of an iris dataset and a wine dataset from UCI Machine Learning Repository [13] were used to test the algorithm. The iris dataset consists of 150 samples belonging to one of three clusters, namely: iris setosa, iris versicolor, or iris virginica. Each class has 50 samples with 4 attributes [14]. On the other hand, the wine dataset contains 178 samples with 13 attributes belonging to one of three clusters [15]. Both datasets are multivariate with no missing values. For both datasets, 70% was used for training and 30% for clustering.

The result of conventional SOM and modified SOM was then compared. The performance comparison was based on the cluster error rate and the number of iterations. Error rate computation was done using equation 9.

\[ \text{Error rate \%} = \frac{\text{Number of incorrectly clustered data}}{\text{Total number of data}}. \] (9)

Meanwhile, the number of iterations needed to achieve convergence to final clustering was noted and used as the measure for the speed of the algorithm.

3. Results and Discussion

The quality of the clustering result and the learning speed of SOM relies on the initialization of weight [3]. With the iris dataset, the conventional SOM has an error rate of 0.0889 and 2100 iterations while an error rate of 0.0925 and 2136 iterations was observed using the wine dataset. Using the SOM with Nguyen-Widrow initialization, an error rate of 0.0444 was computed for the iris dataset and 0.0556 error rate was observed for the wine dataset.

Meanwhile, iterations of 1950 and 1958 were observed for iris and wine datasets respectively. Based on the result, modified SOM gives better cluster result for it has a lower error rate compared to conventional SOM. Moreover, the modified SOM using Nguyen-Widrow initialization algorithm performs faster since it has lesser number of iterations to converge to final clustering than conventional SOM.
An improvement in the training speed of 7.14% and 8.33% from conventional SOM, in iris and wine datasets respectively, was observed in modified SOM. The comparative result of SOM and modified SOM is summarized in Table 2.

|                  | Iris Data | Wine Data |
|------------------|-----------|-----------|
|                  | Error rate | Iterations | Error rate | Iterations |
| SOM              | 0.0889    | 2100      | 0.0925     | 2136       |
| Modified SOM     | 0.0444    | 1950      | 0.0556     | 1958       |

4. Conclusions
The modified SOM performs better in terms of cluster error rate and the number of iterations to converge to final clustering as compared to the conventional SOM. Using the Nguyen-Widrow algorithm for initialization of weights in the SOM thus yields better cluster result and improved training speed of the algorithm in terms of the number of iterations.

References
[1] Kohonen T 2013 Essentials of the self-organizing map *Neural Networks* 37 52–65 Crossref
[2] Valova I, Georgiev G, Gueorguieva N, Olson J 2013 Initialization issues in self-organizing maps *Procedia Computer Science* 20 52–57 Crossref
[3] Aggarwal V, Ahlawat AK, Pandey BN 2013 A weight initialization approach for training Self Organizing Maps for clustering applications In: *Advance Computing Conference (IACC), 2013 IEEE 3rd International* IEEE pp 1000–1005 Crossref
[4] Haripriya H, Devisree R, Pooja D, Nedungadi P 2015 A Comparative Performance Analysis of Self Organizing Maps on Weight Initializations using different Strategies In: *Advances in Computing and Communications (ICACC), 2015 Fifth International Conference on* IEEE pp 434–438 Crossref
[5] Akinduko AA, Mirkes EM, Gorban AN 2016 SOM: Stochastic initialization versus principal components *Information Sciences* 364–365 213–21 Crossref
[6] Dogan Y, Birant D, Kut A 2013 SOM++: integration of self-organizing map and k-means++ algorithms In: *International Workshop on Machine Learning and Data Mining in Pattern Recognition* Springer pp 246–259 Crossref
[7] Nguyen D, Widrow B 1999 Improving the learning speed of 2-layer neural networks by choosing initial values of the adaptive weights In: *1990 IJCNN International Joint Conference on Neural Networks* 3:21–26 Crossref
[8] Adam SP, Karras DA, Magoulas GD, Vrahatis MN 2014 Solving the linear interval tolerance problem for weight initialization of neural networks *Neural Networks* 54 17–37 Crossref
[9] Mishra K, Mittal NK, Mirja MH 2014 Image Compression Using Multilayer Feed Forward Artificial Neural Network with Nguyen Widrow Weight Initialization Method *International Journal of Emerging Technology and Advanced Engineering* 4(4)
[10] Pavelka A, Procházka A 2004 Algorithms for initialization of neural network weights In: *Proceedings of the 12th Annual Conference, MATLAB* 453–459
[11] Chaudhary V, Bhatia RS, Ahlawat AK 2014 A novel Self-Organizing Map (SOM) learning algorithm with nearest and farthest neurons *Alexandria Engineering Journal* 53(4) 827–831
[12] UCI Machine Learning Repository [Internet]. [cited 2018 Jul 8]. Available from: http://archive.ics.uci.edu/ml/index.php
[13] UCI Machine Learning Repository: Iris Data Set [Internet]. [cited 2018 Jul 20]. Available from: http://archive.ics.uci.edu/ml/datasets/Iris
[14] UCI Machine Learning Repository: Wine Data Set [Internet]. [cited 2018 Jul 20]. Available from: http://archive.ics.uci.edu/ml/datasets/Wine