Learning Vector Quantization Neural Network (LVQNN): Can it be implemented on the Forecasting Electrical Loads?

A Dharma¹, N K Mataram², and I N G Antara³

¹ Department of Electrical and Computer Engineering, Faculty of Engineering Udayana University, Jalan Raya Kampus UNUD, Kampus Bukit Jimbaran, South Kuta, Bali 80361
² Department of Civil Engineering, Faculty of Engineering Udayana University, Denpasar, Bali, Indonesia, Jalan Raya Kampus UNUD, Kampus Bukit Jimbaran, South Kuta, Bali 80361
³ Department of Mechanical Engineering, Faculty of Engineering Udayana University, Denpasar, Bali, Indonesia, Jalan Raya Kampus UNUD, Kampus Bukit Jimbaran, South Kuta, Bali 80361

agd2bali@gmail.com

Abstract. Short-term load forecasting is an important activity in planning the operation of the electric power system to estimate the load conditions of the following days and the results may help in the decision making. The imbalance in electrical power between the supply side and the demand side may lead blackouts on the consumer side. Consequently, the generating unit must be operated to meet its load requirements. Various techniques and methods are used to short-term load forecasting. Learning vector quantization neural network (LVQNN) is a classification algorithm that is superior in classifying digital images. Based on these considerations this research aimed to develop LVQNN to forecast short-term electricity peak loads. The idea was used as a reference on the discovery of architectural data classification processes that resembled forecasting techniques. LVQNN development was carried out by adjusting the sample data to the LVQNN architecture. First-distinct sample data were used to obtain the weight vector, then the remaining data from the distinct data process were divided into training data and testing data. By developing the new concept of LVQNN into forecasting technique, the preliminary research was conducted on 11 commodities to predict price fluctuations that have relatively precise forecasting values when implemented in the population. Based on these results, this algorithm will be developed to forecast the fluctuation of electrical load to provide better results in the upcoming research. It is expected to perform with better accuracy results to indicate that the mean absolute percentage error (MAPE) of the predicted values of loads (in MW) will close to the actual loads.

1. Introduction

Forecasting is one of the most important parts in the management decision-making process of a company or business entity. The use of forecasting techniques in decision-making has relevance to the risks that management may face in the future [1]. These forecasting activities are generally identical to trade or other economic activities, though actual forecasting can be implemented in various areas of life [2]–[10]. In its evolution, trading activities have spawned data as a new commodity that can be traded [11].
This phenomenon is in line with technological evolution that leading to management of big data and cloud computing [12]. This is also aligned with the ever-growing number of internet users [11]. Thus, classification techniques and data forecasting become one of the techniques required in the handling or management of big data in the future. Some of research studies suggest that the Learning Vector Quantization Algorithm (LVQNN) has LVQ has proven its reliability in data classification [10], [13], [14]. However, most of these studies illustrate the effectiveness of LVQNN in classifying digital images. From these considerations, the preliminary research has revealed that the reliability of LVQNN can be used to predict commodity prices. In this research, forecasting commodity prices is influenced by six parameters. These parameters include the price of the commodity itself, the IDR exchange rate against the USD, the amount of exports and imports, the percentage of inflation as well as the ORB price [15], [16]. The selection of these parameters is based on Indonesia's economic indicators [17] with the ORB price as an external indicator. The data of each variable is daily data with a population of about 1,600 data for each commodity category. The development of LVQNN to forecast commodity price fluctuations results in a forecasting system that has a relatively precise forecasting value when implemented in the population. Then based on these result, this algorithm will be continued to forecast the fluctuation of electrical load to provide better results in the upcoming research.

2. Learning Vector Quantization Neural Network

The architecture of LVQNN used in the study as described in Figure 1

![Figure 1. Example of LVQNN Architecture [18]](image)

In Figure 1, there are several notations used to help understanding of the LVQNN architecture, including:

- \( x = x_1, x_2, x_3, x_4, x_5, \ldots, x_n \) is an input vector in which its implementation is filled with training data and testing data.
- \(|x_{11} - w_{11}|, |x_{11} - w_{12}|, |x_{11} - w_{13}|, \ldots, |x_{ij} - w_{mn}|\) is a weight vector, where in this vector we will calculate the Euclidean distance between the input vectors to the weight vector.
- \( w_{ij} = w_{11}, w_{12}, w_{21}, w_{22}, w_{31}, \ldots, w_{mn} \) is the Euclidean distance between the input vectors to the weight vectors.
- \( y_{in} = y_{in1}, y_{in2}, y_{in3}, \ldots, y_{in_n} \) is the smallest selected Euclidean distance value of some Euclidean distance value from the result of input vector comparison to weight vector. This value can also be termed as an "Actual Target"
- \( F = F_1, F_2, F_3 \ldots, F_n \) is an activation function, which in the function the target value is compared with the actual value.
\[ y = y_1, y_2, y_3, ..., y_n \] is the computational output that contains the new weight value, which is the result of the weight update based on the activation function.

The initial stage of LVQNN computing is the adjustment of data to the LVQNN architecture. The data processed by LVQNN is divided into weight data, training data and testing data. After the training data, testing data and weighted data meet these requirements, then training on LVQNN to forecast fluctuations in commodity prices as a preliminary study. The training begins with the initialization variable max epoch loop and also learning rate = 0.1.

As long as the loop value has not reached the specified max epoch, then any train data is compared to weighted data. The process of comparing the training data to the weights data is termed as the process of calculating the value of Euclidean distance. The Euclidean distance value can be searched by the Equation (1) as follows:

\[
D(j) = \sqrt{\sum_{i=1}^{n} (x_i - w_{ij})^2}
\]  

where:
\[ x = x_1, x_2, x_3, x_4, x_5, \ldots, x_n \] are input vectors
\[ w_{ij} = w_{1j}, w_{2j}, w_{3j}, w_{4j}, w_{5j}, \ldots, w_{nj} \] are weight vectors for each input, can also be expressed as a classifier vector.

The next process is to compare the value of the actual class "Predictable Price" parameter (as the actual target) to the value of the "Predicted Price" parameter of the training data (as the target forecasting). If the actual target has the same value as the forecasting target then the update-weighted data approaches the input vector (train data) through the Equation (2).

\[
w_{j}(new) = w_{j}(old) + \alpha \cdot (x - w_{j}(old))
\]  \(2\)

S whereas if the actual target has the same value as the forecast target, then the weight vector (or class vector) is updated near the cluster with the following Equation (3).

\[
w_{j}(new) = w_{j}(old) - \alpha \cdot (x - w_{j}(old))
\]  \(3\)

where:
\[ \alpha = 0.1 \] is the learning rate, whose value is reduced in each epoch
\[ x = x_1, x_2, x_3, x_4, x_5, \ldots, x_n \] are input vectors
\[ w_{ij} = w_{1j}, w_{2j}, w_{3j}, w_{4j}, w_{5j}, \ldots, w_{nj} \] are weight vectors for each input, can also be expressed as a classifier vector.
\[ j = 1, 2, 3, 4, 5, \ldots, n \]

After the weight update process is done, then the value of learning rate is reduced gradually in every iteration epoch change. The process of updating the value of learning rate is done by the Equation (4).

\[
\alpha_{new} = \alpha_{old} \cdot \left(1 - \frac{epoch\ loop}{max\ epoch\ loop}\right)
\]  \(4\)

where:
\[ \alpha = \] is the learning rate, whose value is reduced in each epoch, its initial value is 0.1

After a series of processes are completed, the series of processes is counted as a series of processes in an epoch. The series of processes repeats until the epoch loop value reaches the value of the loop limit that is max epoch loop fulfilled.
3. Preliminary Research on Forecasting Commodities

The commodity data used in the preliminary research were the data of the national cost of basic needs [15]. The data were chosen because they were relatively more complete when compared with other commodities. There were 11 types of commodities processed by LVQNN architecture. The amount of sample data used in this research were 303 rows of data. The sample data were divided into weight (classifier data), training and testing data. The process of dividing the sample data into weight data (classifier data) was conducted by the technique of distinct data to obtain each row of unique values of the sample data. The rest of the sample data were distinct then divided into two with were training and testing data with a ratio of 50:50. The samples of training data used in the research are shown in Table 1.

Table 1. The Example Of Training Or Testing Data For Commodity Unpacking Cooking Oil [15], [16], [17]

| Commodity Price (IDR/Kg) | Rupiah Exchange (IDR/1USD) | Exports Values (Million US$) | Import Values (Million US$) | Inflation Rate | ORB Price (US$) | Forecast Price (IDR/Kg) |
|--------------------------|-----------------------------|------------------------------|-----------------------------|----------------|-----------------|-------------------------|
| 9960                     | 8924                        | 12181.6                      | 9654.1                      | 0.44           | 75.49           | 9960                    |
| 10026                    | 8928                        | 14399.6                      | 12120                       | 0.06           | 81.07           | 9989                    |
| 9953                     | 9802                        | 16133.4                      | 14770.3                     | 0.03           | 102.75          | 9960                    |
| 9960                     | 9802                        | 16133.4                      | 14770.3                     | 0.03           | 102.11          | 9965                    |

While the sample weight data used in the study is as shown in Table 2.

Table 2. The Example Of Classifier Data (Weight Vector) For Commodity Unpacking Cooking Oil [15], [16], [17]

| Commodity Price (IDR/Kg) | Rupiah Exchange (IDR/1USD) | Exports Values (Million US$) | Import Values (Million US$) | Inflation Rate | ORB Price (US$) | Forecast Price (IDR/Kg) |
|--------------------------|-----------------------------|------------------------------|-----------------------------|----------------|-----------------|-------------------------|
| 10009                    | 8924                        | 12181.6                      | 9654.1                      | 0.44           | 74.04           | 9989                    |
| 9989                     | 8924                        | 12181.6                      | 9654.1                      | 0.44           | 75.54           | 9965                    |
| 9965                     | 8924                        | 12181.6                      | 9654.1                      | 0.44           | 75.37           | 9960                    |

The training, testing and weight data have the same architecture, the only difference is in the weighted data structure which has no redundancy value on its "Price Predictions" parameter. This is because the weighting data serve as a classified data, which is as reference data of the purpose of the classification process or forecasting, where in the class set is not allowed the existence of classes of equal value. Computation process is done twice for each commodity. The first computation is conducted by testing LVQNN on data training, testing and pure weight data. While the second computation is performed by testing LVQNN to the training data and testing data combined with weight data to add variant of training data and testing data.
The process of distinct data is done to divide the sample data into input data and classifier data resulting in a relatively large difference of the amount of data. This relatively large disparity is caused by relatively "heterogeneous" price fluctuations, as to produce a relatively large amount of data classifier. However, the distinct data in price forecasting are a mandatory process for obtaining values that can be classed as data. Because in the process of classifying the data, it is not allowed to have classes of the same value.

The process of distinct data is relatively influential on LVQNN computing in forecasting. Since the amount of pure training data and pure testing data become relatively small. This impacts on the percentage of forecasting which also become relatively small. However, this does not mean that LVQNN forecasting results is not favourable or inappropriate. Because when viewed from the composition of pure exercise data and pure testing data against classifier data (or weight data) it was clear that the number and variation of the data could not represent the variant of the classifier data.

However, any pure training data and pure testing data were represented in the classifier data through a distinct data process performed on the sample data. Figure 2 shows one example of the graph of the accuracy value of each commodity category with the training data and pure testing data. Figure 3 shown one example of the graph of the accuracy value of each commodity category with a combination of weight data on training data and testing data.

![Figure 2](image1.png)

**Figure 2.** Percentage Accuracy Value for Unpacking Cooking Oil Commodities with Training Data and Pure Testing Data

![Figure 3](image2.png)

**Figure 3.** Percentage Accuracy Value for Unpacking Cooking Oil with Combination of Weight Data on Training Data and Test Data
Table III shows the percentage of forecasting accuracy of 11 types of commodity during the testing process. The percentage of forecasting is inversely proportional to the deviation value. The smaller the value of the forecasting deviation the higher the percentage of accuracy.

**Table 3. The Percentage of Accuracy on LVQ Testing Process For All Pricing Commodities Fluctuation**

| Commodity Name          | Percentage of Testing Accuracy |
|-------------------------|-------------------------------|
|                         | Pure Training & Testing Data | Added with Classifier Data |
| Unpacking Cooking Oil   | 0                             | 90.46                        |
| Beef                    | 25                            | 94.44                        |
| Ranged Chicken Meat     | 0                             | 96.97                        |
| Ranged Chicken Eggs     | 16.6                          | 97.97                        |
| Wheat Flour             | 9.43                          | 77.20                        |
| Imported Soybean        | 15                            | 83.33                        |
| Local Soybean           | 10.41                         | 80.85                        |
| Medium Rice             | 14.58                         | 60.93                        |
| Granulated Sugar        | 13.51                         | 82.77                        |
| Curly Red Chili         | 40                            | 98.32                        |
| Onion                   | 33.33                         | 96.96                        |
| Unpacking Cooking Oil   | 0                             | 90.46                        |

Improvement from LVQNN computational results for forecasting can be seen from the increasing percentage of success after the training data or testing data is modified by adding the classifier data itself as input. The addition of this classifier data adds to the variant of training data and testing data, resulting in an improvement on forecasting percentage.

**4. Discussion**

The addition of classifier data to train data and testing data on LVQNN forecasting on commodity data (without data sampling process) shows an increase to forecasting percentage value. The decrease in RMSD value on each iteration has the opposite effect on the percentage of increasing forecast accuracy in each iteration. The average forecasting accuracy tested for 11 commodity categories is 87.2955%. This value was obtained from the computational model by adding weight data to training data and testing data to add variants of training data and testing data used in the computing process. However, the data variant was a value that affects the results of forecasting LVQNN. This can be seen from the average value of the percentage of accuracy that only reaches 16.1764% in the computational model with training data and pure testing data. The percentage value of forecasting accuracy decreased dramatically to reach 1.2982% in the computational model by adding only weight data to training data. The addition of weight data carried out only on the training data clearly makes the size and variance of the testing data imbalance to the training data and weight data. So that there is a very significant decrease in forecasting accuracy. This is also confirmed by the average forecasting accuracy values generated by computational models that use training data and pure testing data and are applied to populations without data sampling. The average value of accuracy produced only reaches a value of 8.1136%.

**5. Conclusion**

Based on the previous research, it can be concluded that LVQNN can be used to forecast commodity price fluctuations. The average value of the LVQNN forecasting accuracy percentage of eleven commodities on National Basic Needs was 87.2955%.
Furthermore, if the data predicted by LVQNN is too heterogeneous, then the percentage of forecasting accuracy will be reduced as a result of the small number of data variants. Testing LVQNN computing without providing proper learning may not improve the quality of computing or forecasting accuracy. It is imperative to involve data weighting in computing to add learning variants capable of increasing computational results and percentage accuracy of LVQNN forecasting.

Finally, the results of the preliminary study has shown that with the development of the LVQNN method, it can be used for forecasting with good results. The results of this research are the basis for further development g, which are expected to provide significant results.

6. Reference

[1] Kasmir and Jakfar, Studi Kelayakan Bisnis: Edisi Revisi. Jakarta: Prenada Media, 2015.
[2] Agus Dharma, Imam Robandi, and Hery Purnomo, “Application of Interval Type-2 Fuzzy Logic System in Short Term Load Forecasting on Special Days,” J. Theor. Appl. Inf. Technol., vol. 49, Mar. 2013.
[3] P. Podržaj and A. Čebular, “The Application of LVQ Neural Network for Weld Strength Evaluation of RF-Welded Plastic Materials,” IEEEASME Trans. Mechatron., vol. 21, no. 2, pp. 1063–1071, Apr. 2016.
[4] Y. Xu, X. Zhang, and H. Gai, “Quantum Neural Networks for Face Recognition Classifier,” Procedia Eng., vol. 15, pp. 1319–1323, 2011.
[5] F. An, X. Zhang, L. Chen, and H. J. Mattausch, “A Memory-Based Modular Architecture for SOM and LVQ with Dynamic Configuration,” IEEE Trans. Multi-Scale Comput. Syst., vol. 2, no. 4, pp. 234–241, Oct. 2016.
[6] H. T. Yang, Y. C. Huang, and Y. S. Pai, “A Weather-Based Hybrid Method for 1-Day Ahead Hourly Forecasting of PV Power Output,” IEEE Trans. Sustain. Energy, vol. 5, no. 3, pp. 917–926, Jul. 2014.
[7] L. Wang, Y. Chai, and Y. Liu, “Analysis of specialized production of transaction services based on essential services quantity,” Tsinghua Sci. Technol., vol. 22, no. 5, pp. 529–538, Sep. 2017.
[8] D. R. Reising, M. A. Temple, and J. A. Jackson, “Authorized and Rogue Device Discrimination Using Dimensionally Reduced RF-DNA Fingerprints,” IEEE Trans. Inf. Forensics Secur., vol. 10, no. 6, pp. 1180–1192, Jun. 2015.
[9] A. Demirhan, M. Törü, and İ. Güler, “Segmentation of Tumor and Edema Along With Healthy Tissues of Brain Using Wavelets and Neural Networks,” IEEE J. Biomed. Health Inform., vol. 19, no. 4, pp. 1451–1458, Jul. 2015.
[10] V. B. Semwal and G. C. Nandi, “Toward Developing a Computational Model for Bipedal Push Recovery,” IEEE Sens. J., vol. 15, no. 4, pp. 2021–2022, Apr. 2015.
[11] F. Liang, W. Yu, D. An, Q. Yang, X. Fu, and W. Zhao, “A Survey on Big Data Market: Pricing, Trading and Protection,” IEEE Access, vol. 6, pp. 15132–15154, 2018.
[12] Z. Wang, N. M. Nasrabadi, and T. S. Huang, “Spatial #x2013;Spectral Classification of Hyperspectral Images Using Discriminative Dictionary Designed by Learning Vector Quantization,” IEEE Trans. Geosci. Remote Sens., vol. 52, no. 8, pp. 4808–4822, Aug. 2014.
[13] M. M. Al-Rifaie, A. Aber, and D. J. Hemanth, “Deploying Swarm Intelligence in Medical Imaging Identifying Metastasis, Micro-Calcifications and Brain Image Segmentation,” IET Syst. Biol., vol. 9, no. 6, pp. 234–244, 2015.
[14] G. Wang and J. Unger, “A Strategy to Move Taiwan’s IT Industry From Commodity Hardware Manufacturing to Competitive Cloud Solutions,” IEEE Access, vol. 1, pp. 159–166, 2013.
[15] “Tabel Harga Kebutuhan Pokok Nasional Kementerian Perdagangan Republik Indonesia,” kemendag.go.id. [Online]. Available: http://www.kemendag.go.id/id. [Accessed: 03-May-2018].
[16] “OPEC : OPEC Basket Price.” [Online]. Available: http://www.opec.org/opec_web/en/data_graphs/40.htm. [Accessed: 03-May-2018].
[17] “Indikator Ekonomi Indonesia Kementerian Perdagangan Republik Indonesia,” kemendag.go.id. [Online]. Available: http://www.kemendag.go.id/id. [Accessed: 03-May-2018].

[18] Kusumadewi S., Membangun Jaringan Saraf Tiruan Menggunakan MATLAB dan Excel Link. Yogyakarta: Graha Ilmu, 2004.

Acknowledgments
This research/article’s publication is supported by the United States Agency for International Development (USAID) through the Sustainable Higher Education Research Alliance (SHERA) Program for Universitas Indonesia’s Scientific Modeling, Application, Research and Training for City-centered Innovation and Technology (SMART CITY) Project, Grant #AID-497-A-1600004, Sub Grant #IIE-00000078-UI.1. Appreciation is given as well to Oka Wisnawa for his strong effort on data analyzing on previous research.

This article is presented at the International Conference on Smart City Innovation 2018 that supported by the United States Agency for International Development (USAID) through the Sustainable Higher Education Research Alliance (SHERA) Program for Universitas Indonesia’s Scientific Modeling, Application, Research and Training for City-centered Innovation and Technology (SMART CITY) Project, Grant #AID-497-A-1600004, Sub Grant #IIE-00000078-UI-1.