Fuzzy Systems and Data Mining VIII
A.J. Tallón-Ballesteros (Ed.)

doi:10.3233/FAIA220387

Optimization of Fuzzy System Inference Model on Mini Batch Gradient Descent

Sugiyarto SURONO[a,1], Aris THOBIRIN[a], Zani Anjani Rafsanjani HSM[a], Asih Yuli ASTUTI[a], Berlin Ryan KP[a], and Milla OKTAVIA[a]
[a] *Mathematics Study Program, Ahmad Dahlan University, Yogyakarta, Indonesia*

Abstract. Optimization is one of the factors in machine learning to help model training during backpropagation. This is conducted by adjusting the weights to minimize the loss function and to overcome dimensional problems. Also, the gradient descent method is a simple approach in the backpropagation model to solve minimum problems. The mini-batch gradient descent (MBGD) is one of the methods proven to be powerful for large-scale learning. The addition of several approaches to the MBGD such as AB, BN, and UR can accelerate the convergence process, hence, the algorithm becomes faster and more effective. This added method will perform an optimization process on the results of the data rule that has been processed as its objective function. The processing results showed the MBGD-AB-BN-UR method has a more stable computational time in the three data sets than the other methods. For the model evaluation, this research used RMSE, MAE, and MAPE.

Keywords. Fuzzy System Inference, Machine learning, Optimization

1. Introduction

The development of machine learning in recent years has become a special concern in Artificial Neural Networks (ANN). This ANN is an information processing system designed to imitate the human brain. Rasit [1] found the use of artificial neural networks in various ways of pattern recognition, optimization, simulation, and prediction. One of the most widely used Neural Network techniques is backpropagation. It is one of the algorithms often used in solving complex problems due to its high accuracy. However, there are several weaknesses, including being trapped in the local minimum [2]. One of the backpropagation methods to solve the minimum problem is gradient descent [3]. In recent years, a gradient descent method was developed to improve the performance of deeper neural networks [4].

The speed of convergence depends on the initial parameters, such as the number of hidden notes, inputs, outputs, learning rates, and weights in the network. Therefore, optimization is one of the keys in machine learning to help model training during backpropagation by adjusting the weights to minimize the loss function and to overcome dimensional problems. Conventional optimization methods relate to the selection of variables that optimize the objective function. However, when optimization problems in the real world become more complex, then conventional algorithms are uncertain to solve

[1] Corresponding Author, Sugiyarto SURONO, Mathematics Study Program, Ahmad Dahlan University, Yogyakarta, Indonesia; E-mail: sugiyarto@math.uad.ac.id.

the challenges [5]. In recent years, modern optimization methods have emerged to solve complex problems [6]. These methods include genetic algorithm [7], simulated annealing [8], as well as particle swarm [9], ant colony, neural network-based, and fuzzy optimizations. The idea of optimization in the fuzzy system needs to be modified because the objective and constraint functions are characterized by membership functions in the system [6].

Mini-batch gradient descent (MBGD) is one of the methods proven to be powerful for large-scale learning [10]. The related research includes Khirirat et al [11], which used MBGD to reduce variance in gradient estimation and utilized several matrix optimizations to make the algorithm more optimal. Furthermore, Pengqi and Jianjun [12] used MBGD to train an ANN equalizer automatically and efficiently. Jing Li et al [13] also used MBGD to overcome the high computational costs of large-scale real hyperspectral images.

Batch Normalization (BN) is an approach added to the MBGD method to normalize batches in datasets with the aim to accelerate the processing performance in the aggregation section. In the training process, each scalar function is normalized by making the average zero and adding variance to the smallest dimension [14]. The expansion of the Neural Network can be assumed as a layer of random samples from the distribution of the dataset that has been modified during the training process or iteration [15].

The regularization technique is used to avoid overfitting and increase generalization. This regularization provokes generalization of the algorithm by avoiding coefficients to fit the training sample data [16]. According to Goodfellow [17], regularization is "any modification made to a learning algorithm intended to reduce generalization errors and not training errors". Accordingly, a regularization technique is needed to stabilize the numerical calculations [18].

AdaBound [19] is one of the optimizers used to optimize the learning rate on MBGD. This increases the speed of convergence and its optimization, then it converges to a global minimum at the end of the training [20]. Liu et al [21] proved that with AdaBound iterations, the objective function converges to a finite value and the corresponding gradient converges to a minimum value.

2. Method

2.1. Takagi Sugeno Kang (TSK)

TSK fuzzy inference system is a method that is represented in the form of *if-then*, where the output is not a fuzzy set but a function or constant [22]. The steps to produce output on this fuzzy system are as follows [23].

The first step is the fuzzification process. The input variable is entered in the membership function of the antecedent part of the fuzzy rule to obtain the membership value of each linguistic label. The second step is the ground rules. The basic rules are made based on information created by experts or obtained from numerical data. The rule base is expressed in the form of *if-then*, where *if* is the antecedent and *then* is the consequent. Mathematically, it can be written as follows:

$$IF\ (x_1\ is\ A_1)\ dan\ (x_2\ is\ A_2)\ THEN\ z = k \qquad (1)$$

After establishing the basic rules, the next step is to find an inference engine, which is the process of getting the α–predicate value of each rule $(\mu_1, \mu_2, \dots, \mu_i)$ using the max implication function. Furthermore, each $\alpha - predicate$ value is used to calculate the crisp inference output for each rule $(z_1, z_2, \dots, z_i)$ [24]. The final step is defuzzification, which is defined as the process of converting fuzzy values into firm numbers. The input is a set obtained from the composition of fuzzy rules [25].

$$Y = \frac{\sum_{i=1}^{n} \alpha_i y_i}{\sum_{i=1}^{n} \alpha_i}, \qquad i = 1,2,3,\dots,n \tag{2}$$

Where α_i is the aggregation value in the $i - th\, rule$, y_i is the output in the $i - th$ and the number of rules used.

2.2. Mini-batch gradient descent

The rules obtained in the TSK fuzzy inference system are modified using MBGD, which is a combined update of Batch and Stochastic Gradient Descents. It is proven to cope with large-scale datasets by reducing computational complexity in each iteration while reducing optimization time complexity by using small datasets to update each iteration. These small data are called mini-batches [26][27]. The MBGD algorithm can be written as follows:

$$\theta_{t+1} = \theta_t - \eta \times \nabla_\theta MSE(z_t, \theta_t) \tag{3}$$

where z_t is mini-batch taken randomly.

2.3. Optimization

AdaBound, BN, and UR were used to optimize the TSK-MBGD model. AdaBound is an adaptive optimization method that uses dynamic limits on the learning rate [19]. Furthermore, it limits the learning rate from upper and lower, hence a rate that is too large or too small cannot occur. The function used to determine the upper and lower limits is as follows:

$$\begin{aligned} l(k) &= 0.01 - \frac{0.01}{(1-\beta_2)k + 1} \\ u(k) &= 0.01 - \frac{0.01}{(1-\beta_2)k} \end{aligned} \tag{4}$$

In the initial training $(k = 0)$, the limit is $[0, +\infty)$. During the training $(k \to \infty)$, the limit is close to $[0.01, 0.01]$.

The second optimization method is UR. The goal of UR is to have a similar and minimized average firing level. Also, the UR loss function can be added to the original in the MBGD training. The updates to the MBGD-UR parameters in each rule are as follows [26]:

$$L = l + \eta l_2 + \lambda \sum_{i=1}^{N} \left(\frac{1}{N} (y_i - \hat{y}_i) \right)^2 \tag{5}$$

Where l is the cross entropy lost between class probability estimates, l_2 is a regularization with l_2, N is the number of training examples, α is the learning rate, and λ is the regularization parameter.

In ANN, the input distribution in each training varies, which is caused by changes in the previous parameters. This slows down and complicates learning of the model during training. Therefore, BN is needed as a technique to increase speed, performance, and stabilization. Shifts and scales in the BN technique represent identity transform batches, hence. The addition of shift (γ) and scale (β) to represent the identity transform batch in the normalization process. the transformation process for each layer can be defined as follows [28]:

$$BN(x_i) = \gamma \hat{x}_i + \beta \tag{6}$$

Where $\tilde{x}_B$ is average mini-batch, x_i is i-th input data, m is the size of mini-batch, B is data groups from $\{x_i, \dots, m\}$, α_B^2 is mini-batch variation, $\hat{x}_i$ is z-score normalization, β is shift training parameter, γ is scale training parameter, and ε is the smallest positive constant.

2.4. Model Evaluation

This evaluation determines the success level of the model. Furthermore, the used model is Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE). RMSE represents the square root of the mean square error. The error rate decreases as the RMSE value approaches zero [29].

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N} (y_i - \hat{y}_i)^2} \tag{7}$$

where y_i is the category value in the original data, $\hat{y}_i$ is the defuzzification value, and N is the amount of data.

Definition [30] MAE measures the average absolute difference between N prediction vector $S = \{x_1, x_2, \dots, x_N\}$ and N actual observation $S = \{y_1, y_2, \dots, y_N\}$, associated with L_1 norm $(\|.\|_1)$. The corresponding loss function is defined as:

$$MAE = \frac{1}{N}\sum_{i=1}^{N} \|x_i - y_i\|_1 \tag{8}$$

MAPE is calculated using the absolute error in each period and divided by the actual observed value, the absolute error percentage is then averaged [31][32].

$$MAPE = \frac{\sum_{i=1}^{N} \frac{|a-b|^2}{a}}{N} \times 100\% \tag{9}$$

Where a is real data, b is predictive data, and n is the amount. The standard value prediction criteria for MAPE is $< 10\%$ (excellent), $10\% - 20\%$ (good), $20\% - 50\%$ (reasonable), and $> 50\%$ (bad) [33].

3. Result and discussion

This research used data on Jakarta Air Pollution (ISPU) from April to October 2021. Data were obtained from the Data Open Jakarta. The first dataset used is shown in Table 1.

Table 1. ISPU Data.

No	X_1	X_2	X_3	X_4	X_5	X_6	Y
1	26	42	22	11	13	19	42
2	41	54	27	11	17	19	54
3	25	42	20	8	18	22	42
…	…	…	…	…	…	…	…
1063	49	74	0	10	31	13	74

The data in Table 2 were preprocessed to fill in the missing values using the average. An encoding process was carried out, where unused columns and rows with no value were deleted. However, variables SO_2 and NO_2 only falls into the good category, indicating they were not used.

Table 2. ISPU Fuzzification.

X_1	X_2	X_3	X_4	Y
0.3800	0.6047	0.5714	0.9661	0.3600
0.6800	0.4082	0.0000	0.2203	0.2000
…	…	…	…	…
0.0000	0.3953	0.5714	0.4407	0.7200

After obtaining the fuzzy values, the second step of the TSK system is the formation of rules in the form of fuzzy implications that state the relationship between input and output variables. The basic rules formed from ISPU data are as follows.
[R1] When X_1 is Moderate or X_2 is not Healthy, X_3 X_2 is Not Healthy. When X_4 is Good then Y is Moderate.

$$\vdots \qquad \vdots \qquad \vdots \qquad \vdots \qquad \vdots$$

[R17] When X_1 is Good, X_2 is not Healthy or X_3 and X_4 are Good, then Y is Moderate

The third stage of the system is to perform fuzzy inference of several rules obtained from the collection and correlation between rules. The max (maximum) method is used in performing the TSK fuzzy system inference.

Table 3. ISPU Interference.

Rule	1	2	…	16	17
ISPU	1.0000	0.9831	…	1.0000	0.4915

$\alpha_{predicate}$ from ISPU data on the first rule has a value of 1.0000, while the second rule has 0.9831. Furthermore, each rule is optimized using seven algorithms including

MBGD-AB, MBGD-BN, MBGD-UR, MBGD-AB-BN, MBGD-AB-UR, MBGD-BN-UR, and MBGD-AB-BN-UR. The regression function is obtained as follows.

a) Linear function of MBGD-AB algorithm

$$\begin{aligned}&[\text{R1}]\ Z^* = 0.2553 + 0.0020x_1 + 0.7417x_2 - 0.0080x_3 - 0.0007x_4\\ &\qquad\vdots \qquad\vdots \qquad\vdots \qquad\vdots \qquad\vdots\\ &[\text{R17}]\ Z^* = -0.8267 + 0.4190x_1 + 1.0791x_2 + 0.4043x_3 + 0.6778x_4\end{aligned}$$

b) Linear function of MBGD-BN algorithm

$$\begin{aligned}&[\text{R1}]\ Z^* = 0.2536 - 0.0080x_1 + 0.7482x_2 - 0.0057x_3 + 0.0027x_4\\ &\qquad\vdots \qquad\vdots \qquad\vdots \qquad\vdots \qquad\vdots\\ &[\text{R17}]\ Z^* = -0.8123 + 0.4776x_1 + 0.9616x_2 + 0.4043x_3 + 0.3914x_4\end{aligned}$$

c) Linear function of MBGD-UR algorithm

$$\begin{aligned}&[\text{R1}]\ Z^* = -0.3050 + 0.6790x_1 + 0.8546x_2 + 0.4456x_3 + 0.0182x_4\\ &\qquad\vdots \qquad\vdots \qquad\vdots \qquad\vdots \qquad\vdots\\ &[\text{R17}]\ Z^* = -0.7871 + 0.5480x_1 + 0.8242x_2 + 0.4759x_3 + 0.6482x_4\end{aligned}$$

d) Linear function of MBGD-AB-BN algorithm

$$\begin{aligned}&[\text{R1}]\ Z^* = 0.2554 + 0.0026x_1 + 0.7415x_2 - 0.0080x_3 - 0.0008x_4\\ &\qquad\vdots \qquad\vdots \qquad\vdots \qquad\vdots \qquad\vdots\\ &[\text{R17}]\ Z^* = -0.8303 + 0.4040x_1 + 1.1049x_2 + 0.4130x_3 + 0.6727x_4\end{aligned}$$

e) Linear function of MBGD-AB-UR algorithm

$$\begin{aligned}&[\text{R1}]\ Z^* = -0.3050 + 0.6790x_1 + 0.8546x_2 + 0.4456x_3 + 0.0182x_4\\ &\qquad\vdots \qquad\vdots \qquad\vdots \qquad\vdots \qquad\vdots\\ &[\text{R17}]\ Z^* = -0.7871 + 0.5480x_1 + 0.8242x_2 + 0.4759x_3 + 0.6482x_4\end{aligned}$$

f) Linear function of MBGD-BN-UR algorithm

$$\begin{aligned}&[\text{R1}]\ Z^* = -0.3050 + 0.6790x_1 + 0.8550x_2 + 0.4460x_3 + 0.0182x_4\\ &\qquad\vdots \qquad\vdots \qquad\vdots \qquad\vdots \qquad\vdots\\ &[\text{R17}]\ Z^* = -0.7871 + 0.5480x_1 + 0.8242x_2 + 0.4759x_3 + 0.6482x_4\end{aligned}$$

g) Linear function of MBGD-BN-UR algorithm

$$\begin{aligned}&[\text{R1}]\ Z^* = -0.6070 + 0.6734x_1 + 0.9339x_2 + 0.6826x_3 + 0.3111x_4\\ &\qquad\vdots \qquad\vdots \qquad\vdots \qquad\vdots \qquad\vdots\\ &[\text{R17}]\ Z^* = -0.7871 + 0.5480x_1 + 0.8242x_2 + 0.4759x_3 + 0.6482x_4\end{aligned}$$

The fourth step of the system is defuzzification, which is the process of converting fuzzy numbers into firm. The defuzzification value can be calculated as follows.

Table 4. Defuzzification result of ISPU dataset.

MBGD AB	MBGD BN	MBGD UR	MBGD AB-BN	MBGD AB-UR	MBGD BN-UR	MBGD AB-BN-UR
65.9712	66.7914	74.4366	64.7281	72.8482	74.7720	74.0331
92.0287	94.3085	94.4062	92.7757	92.6824	94.7941	92.7612
…	…	…	…	…	…	…
96.1493	97.6603	84.3119	98.5433	98.5433	100.6128	118.4188

After obtaining the defuzzification results, the model was evaluated using RMSE, MAE, and MAPE. Also, the error calculation used the formula in equation (10)-(12). The results of the error values obtained show in Figure 1.

Figure 1 shows the error comparison of ISPU dataset using RMSE, MAE, and MAPE between 7 methods in percent. Besides the error value, the computational time for each rule was also obtained in the MBGD-AB, MBGD-BN, MBGD-UR, MBGD-AB-BN, MBGD-AB-UR, MBGD-BN-UR, and MBGD-AB-BN-UR methods.

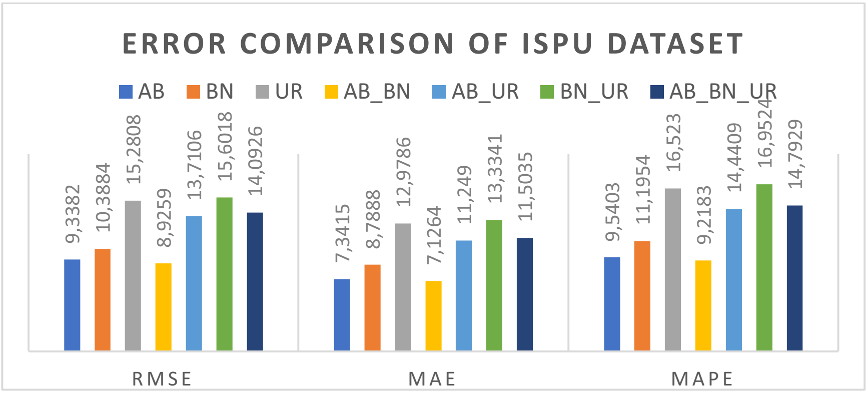

Figure 1. Error comparison of ISPU dataset.

Computational time for ISPU data using the MBGD-AB, MBGD-BN, MBGD-UR, MBGD-AB-BN, MBGD-AB-UR, MBGD-BN-UR, and MBGD-AB-BN-UR methods show in Figure 2. with the average computation time of 0.8304, 1.0300, 0.4777, 0.3897, 0.3016, 0.2967, and 0.3225, respectively.

Figure 2 shows the time comparison between the MBGD-AB, MBGD-BN, MBGD-UR, MBGD-AB-BN, MBGD-AB-UR, MBGD-BN-UR, and MBGD-AB-BN-UR methods on the ISPU dataset. It can be seen that the MBGD-AB-BN-UR has a more stable time than other methods.

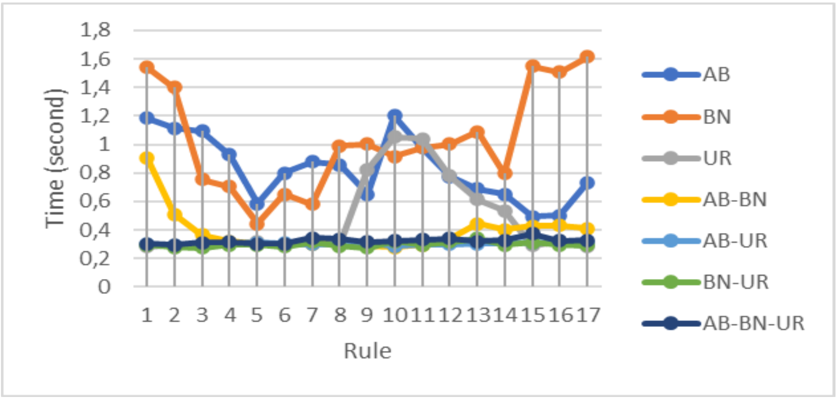

Figure 2. Comparison of time on ISPU dataset.

4. Conclusion

The TSK fuzzy system is a very useful machine learning model for regression and classification problems. Experiments were carried out on the ISPU, and the methods used to process the data include the MBGD-AB, MBGD-BN, MBGD-UR, MBGD-AB-BN, MBGD-AB-UR, MBGD-BN-UR, and MBGD-AB-BN-UR. From the processing results, MBGD-AB-BN-UR has a more stable computation time.

As segmentation data with many dataset partitions into the same group, clustering can be used as another alternative method of predictive analysis on a particular problem.

This is because clustering can find unknown groups in the data. The implemented method only focused on changing the data. Therefore, this research can be developed by adding other methods, namely AGD, Adagrad, Adadelta, RMSSprop, and Adam.

References

[1] Ata R. Artificial neural networks applications in wind energy systems: a review. Renew. Sustain. Energy Rev. 2015; 49: 534–562, doi: 10.1016/j.rser.2015.04.166.
[2] Asriningtias SR, Dachlan HS, Yudaningtyas E. Optimasi training neural network using hybrid adaptive mutation. Eeecis, 2015;9(1): 79–84.
[3] Krestinskaya O, Salama KN, James AP. Learning in memristive neural network architectures using analog backpropagation circuits. IEEE Trans. Circuits Syst. I Regul. Pap., 2019;66(2): 719–732, doi: 10.1109/TCSI.2018.2866510.
[4] Soydaner D. A comparison of optimization algorithms for deep learning. Int. J. Pattern Recognit. Artif. Intell. 2020;34(13), doi: 10.1142/S0218001420520138.
[5] W. Li, G. G. Wang, and A. H. Alavi, "Learning-based elephant herding optimization algorithm for solving numerical optimization problems," Knowledge-Based Syst., vol. 195, p. 105675, 2020, doi: 10.1016/j.knosys.2020.105675.
[6] S. S. Rao, Engineering optimization: Theory and practice, Fourth Edi. Hoboken, New Jersey: John Wiley & Sons, Inc., 2019.
[7] S. Mishra, "Genetic Algorithm: An Efficient Tool for Global Optimization," Adv. Comput. Sci. Technol., vol. 10, no. 8, pp. 2201–2211, 2017, [Online]. Available: http://www.ripublication.com.
[8] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by Simulated Annealing Downloaded from," Sci. 220(4598), vol. 220, no. 4598, pp. 671–680, 1983.
[9] M. O. Okwu and L. K. Tartibu, "Particle Swarm Optimisation," Stud. Comput. Intell., vol. 927, pp. 5–13, 2021, doi: 10.1007/978-3-030-61111-8_2.
[10] H. R. Feyzmahdavian, A. Aytekin, and M. Johansson, "An Asynchronous Mini-Batch Algorithm for Regularized Stochastic Optimization," IEEE Trans. Automat. Contr., vol. 61, no. 12, pp. 3740–3754, 2016, doi: 10.1109/TAC.2016.2525015.
[11] S. Khirirat, H. R. Feyzmahdavian, and M. Johansson, "Mini-batch gradient descent: Faster convergence under data sparsity," 2017 IEEE 56th Annu. Conf. Decis. Control. CDC 2017, vol. 2018-Janua, no. Cdc, pp. 2880–2887, 2018, doi: 10.1109/CDC.2017.8264077.
[12] P. Gou and J. Yu, "A nonlinear ANN equalizer with mini-batch gradient descent in 40Gbaud PAM-8 IM/DD system," Opt. Fiber Technol., vol. 46, no. September, pp. 113–117, 2018, doi: 10.1016/j.yofte.2018.09.015.
[13] J. Li, X. Li, and L. Zhao, "Hyperspectral Unmixing Via Projected Mini-Batch Gradient Descent," vol. 0, no. 2, pp. 0–3.
[14] D. Ezzat, H. M. Afify, M. H. N. Taha, and A. E. Hassanien, Convolutional Neural Network with Batch Normalization for Classification of Endoscopic Gastrointestinal Diseases. Springer International Publishing, 2021.
[15] R. Rajeev, J. A. Samath, and N. K. Karthikeyan, "An Intelligent Recurrent Neural Network with Long Short-Term Memory (LSTM) BASED Batch Normalization for Medical Image Denoising," J. Med. Syst., vol. 43, no. 8, 2019, doi: 10.1007/s10916-019-1371-9.
[16] P. Murugan and S. Durairaj, "Regularization and Optimization Strategies in Deep Convolutional Neural Network," pp. 1–15, 2017, [Online]. Available: http://arxiv.org/abs/1712.04711.
[17] I. G. and Y. B. and A. Courville, Deep learning, vol. 29, no. 7553. Boston, MA, USA: MIT Press, 2016.
[18] Z. Gong and H. Yang, "Ill-Posed Fuzzy Initial-Boundary Value Problems Based on Generalized Differentiability and Regularization," Fuzzy Sets Syst., vol. 295, pp. 99–113, 2016, doi: 10.1016/j.fss.2015.04.016.
[19] L. Luo, Y. Xiong, Y. Liu, and X. Sun, "Adaptive Gradient Methods With Dynamic Bound of Learning Rate," no. 2018, pp. 1–19, 2019.
[20] R. Wang, H. Chen, and C. Guan, "Random convolutional neural network structure: An intelligent health monitoring scheme for diesel engines," Meas. J. Int. Meas. Confed., vol. 171, no. December 2020, p. 108786, 2021, doi: 10.1016/j.measurement.2020.108786.
[21] J. Liu, J. Kong, D. Xu, M. Qi, and Y. Lu, "Convergence analysis of AdaBound with relaxed bound functions for non-convex optimization," Neural Networks, vol. 145, pp. 300–307, 2022, doi: 10.1016/j.neunet.2021.10.026.

[22] S. Kusumadewi and H. Purnomo, Fuzzy Logic Application for Decision Support, Edition 2. Yogyakarta: Graha Ilmu, 2013.

[23] S. Thaker and V. Nagori, “Analysis of Fuzzification Process in Fuzzy Expert System,” Procedia Comput. Sci., vol. 132, pp. 1308–1316, 2018, doi: 10.1016/j.procs.2018.05.047.

[24] D. P. P. Astuti and Mashuri, “Application of Fuzzy Tsukamoto and Fuzzy Sugeno Methods in Determining the Selling Price of Motorcycles,” UNNES J. Math., vol. 1, no. 2252, pp. 75–84, 2020.

[25] A. K. Nisa, M. Abdy, and A. Zaki, “Application of Fuzzy Logic to Determine the Best Packaged Milk Drinks in Optimizing Nutrition,” J. Math. Comput. Stat., vol. 3, no. 1, p. 51, 2020, doi: 10.35580/jmathcos.v3i1.19902.

[26] Y. Cui, D. Wu, and J. Huang, “Optimize TSK Fuzzy Systems for Classification Problems: Minibatch Gradient Descent with Uniform Regularization and Batch Normalization,” IEEE Trans. Fuzzy Syst., vol. 28, no. 12, pp. 3065–3075, 2020, doi: 10.1109/TFUZZ.2020.2967282.

[27] A. Mustapha, L. Mohamed, and K. Ali, An Overview of Gradient Descent Algorithm Optimization in Machine Learning: Application in the Ophthalmology Field, vol. 1207 CCIS. Springer International Publishing, 2020.

[28] G. W. P. Data, K. Ngu, D. W. Murray, and V. A. Prisacariu, Interpolating convolutional neural networks using batch normalization, vol. 11217 LNCS. Springer International Publishing, 2018.

[29] B. Nugroho, E. Y. Puspaningrum, and M. S. Munir, “Performance of Root-Mean-Square Propagation Optimization Algorithm and Stochastic Gradient Descent on Covid-19 Pneumonia Classification Using CNN,” J. Edukasi dan Penelit. Inform., vol. 7, no. 3, p. 420, 2021, doi: 10.26418/jp.v7i3.49172.

[30] J. Qi, J. Du, S. M. Siniscalchi, X. Ma, and C. H. Lee, “Analyzing Upper Bounds on Mean Absolute Errors for Deep Neural Network-Based Vector-to-Vector Regression,” IEEE Trans. Signal Process., vol. 68, no. c, pp. 3411–3422, 2020, doi: 10.1109/TSP.2020.2993164.

[31] M. F. Rohmah, L. Ardiantoro, I. K. G. D. Putra, and R. H. Sari, “Forecasting the Regency Consumer Price Index in East Java with the Support Vector Regression Data Mining Method,” Semin. Nas. Apl. Teknol. Informasu 2019, no. 3 Agustus 2019, pp. 30–36, 2019, [Online]. Available: https://journal.uii.ac.id/Snati/article/viewFile/13434/9512.

[32] S. Prayudani, A. Hizriadi, Y. Y. Lase, Y. Fatmi, and Al-Khowarizmi, “Analysis Accuracy of Forecasting Measurement Technique on Random K-Nearest Neighbor (RKNN) Using MAPE and MSE,” J. Phys. Conf. Ser., vol. 1361, no. 1, 2019, doi: 10.1088/1742-6596/1361/1/012089.

[33] P. C. Chang, Y. W. Wang, and C. H. Liu, “The development of a weighted evolving fuzzy neural network for PCB sales forecasting,” Expert Syst. Appl., vol. 32, no. 1, pp. 86–96, 2007, doi: 10.1016/j.eswa.2005.11.021.