MaRz: A Fast, Transparent Fuzzy Machine Learning Technique

Eric Braude(✉) and Seth Gorrin

Boston University, Boston, MA 02215, USA
{ebraude,gorrin}@bu.edu

Abstract. There is significant interest in fast machine learning and in explainability. This paper's contribution is a novel, but straightforward, fuzzy model that learns on the fly, is accurate, and explains its conclusions in a literal manner. MaRz (Machine Learning in Realtime with Fuzziness) treats each record as fuzzy and applies classical fuzzy center-of-gravity calculations. In the interest of trustworthiness, MaRz does not attempt a generalized form of explanation. Instead, it shows the specific data that most contributed to the output and allows those data to be tested in the context of the remaining data. It places at the user's discretion how many such data to provide and thereby increase the explanation. The contribution of this paper is to demonstrate a machine learning approach for categorization and regression of competitive accuracy that is, at the same time, novel, real time, and explainable.

Keywords: Explainable machine learning · Real-time machine learning · Fuzzy machine learning

1 Introduction

1.1 Motivation

The widespread dissemination of machine learning has exposed the need for real-time performance as well as explainability—especially both at the same time. Billions-parameter models have achieved great success, but at the expense of training time and transparency [4]. In the literature, there is a great deal of interest in fast machine learning [1, 17] and real-time performance. Avoiding black box results is also receiving considerable attention [2, 3].

The contributions of this paper are as follows.

1. It introduces a novel machine learning approach competitive in accuracy.
2. Its addresses both real-time performance and explainability.
3. It shows the practicability of interpreting real-world data as fuzzy.

Our research goal is to create a machine learning model that learns on the fly, is accurate and provides simple, reliable explanations of its conclusions. The contribution of MaRz is its combination of novelty, real-time, and explainability.

© ICST Institute for Computer Sciences, Social Informatics and Telecommunications Engineering 2025
Published by Springer Nature Switzerland AG 2025. All Rights Reserved
T. Zlateva and G. Tuparov (Eds.): CSECS 2024, LNICST 609, pp. 65–73, 2025.
https://doi.org/10.1007/978-3-031-84312-9_4

1.2 MaRz Summarized

We present MaRz (Machine Learning in Realtime with Fuzziness), an approach based on fuzzy theory. MaRz assumes that most data can be interpreted fuzzily. For example, in the context of multiple data, the fact *John exercises for 1 h a month* is no better—and perhaps less relevant—than *John exercises Little*, where "Little" is a fuzzy value with center-of-gravity at 1 h a month. A query can be interpreted as "to what degree *is* it in the dataset?", i.e., in the fuzzy sense.

The MaRz learning phase is primarily a sorting process—to prepare for fuzzy calculations. This turns out to be comparatively fast. It also allows continual learning via insertion of new data rather than recompilation (as with neural nets, for example). MaRz explains its conclusions by exhibiting the data that most influence its conclusion.

On the Airfoil dataset [14], MaRz is 6–8 times faster compared with the results in [15], about 0.2% better measured by R-squared than one method, and 4% worse than a second. It is superior in accuracy, squared error, and F1-score separately on three data sets compared with Liquid Time-Constant experiments on real-time classification and regression examples [17]. Its explainability is demonstrated with the UCI heart disease dataset [20].

1.3 Prior Work

As dataset become very large, machine learning approaches are needed that do not simply require more and more training and data. The paradox has often been pointed out that human learning is enabled by limited examples, whereas machine learning requiring large amounts of data.

There have been numerous attempts to apply fuzzy theory to machine learning, including a graph-based approach with cognitive maps [5], time-series and clustering techniques [6], self-tuning and self-learning in fuzzy controllers [7], and defining a data structure for automatically deriving fuzzy if-then rules and membership functions from a set of training examples [8]. None, as far as we can determine, performs the MaRz process of replacing data with fuzzified versions.

Explainable artificial intelligence (XAI), of high interest—see, for example, [9, 10], and [11]—is also a MaRz property.

Hasani et al. (2021) showed how a faithful attention to the nervous system can result in machine learning that executes faster and more accurately than competing continuous-time neural networks [17]. We compare MaRz output with several of Hasani et al.'s results.

2 MaRz Methodology

We start with the data—typically cleaned. MaRz views each datum as fuzzy. Every input/output data pair $(\boldsymbol{i}, \boldsymbol{o})$ is treated as a set of pairs of scalars (a, b) where a is the value of $\boldsymbol{i}$ in one dimension and b the value of $\boldsymbol{o}$ in one dimension. Each such (a, b) is replaced with fuzzy values A and B, isosceles-triangular-shaped, peaking at 1, and centered on a and b respectively.

The input triangles' base widths w_j are fixed for every dimension j, computed as follows, and depends on a parameter $0 < \alpha \leq 1$. Suppose that the (cleaned, outlier-free) input part of the data set has total width $\boldsymbol{t} = (t_1, t_2, t_3, \ldots, t_n)$. We set $\boldsymbol{w} = \alpha \boldsymbol{t}$, maintaining this total shape. One can think of $\boldsymbol{w} = (w_1, w_2, w_3, \ldots, w_n)$ as forming a "hyper-box" in input space similar in shape to the data set as a whole and centered at $\boldsymbol{i}$. This is illustrated in Fig. 1, where $a \approx ¼$, and the hyper-box, centered on the input query, contains approximately one quarter of the total data space, and.

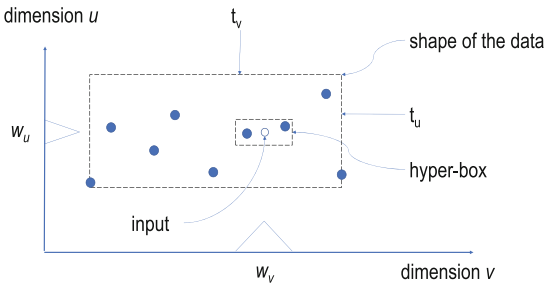

Fig. 1. Two-dimensional-input example—hyper-box from an input and widths of fuzzy values

For a given input and given a, only the data whose input values lie in the corresponding hyper-box contribute to output.

For input $\boldsymbol{i}$ and a given output dimension, each datum D in the hyper-box contributes to the output in that dimension as follows. Let I_j be the fuzzified version of i in input dimension j. As illustrated in Fig. 2, let d_j be the degree to which D's input part belongs to I_j (i.e., the degree to which D's input part "is" I_j). For each input dimension, D's contribution to the output is computed from the resulting trapezoid of height d_j, as in Fig. 2. In this example, there are two relevant data besides D (their inputs alone lie in the hyper-box). The degree to which a datum D "is" I, is d, the minimum of d_j, taken as above over all input dimensions j.

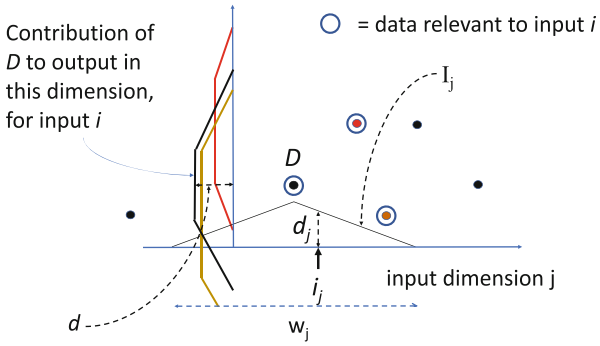

Fig. 2. Computation of output for 1-dimensional input and output

MaRz computes the output value in a traditional fuzzy manner. As illustrated in Fig. 2, this consists of determining the center of gravity of the trapezoids corresponding to all the data in the hyper-box for the given input. This computation is as follows.

Let B be the base length for all output triangular fuzzy values. The area of a trapezoid with height h is the difference between the areas of two similar triangles, the smaller with base b, say, and the larger with base B, which is fixed and has height 1. This is illustrated in Fig. 3. Since $b/(1-h) = B/1$, we have $b = B(1-h)$. The area of the trapezoid is thus:

$$(B \cdot 1/2) - B(1-h)(1-h)/2 \; = \; Bh(2-h)/2$$

—in other words, proportional to $h(2-h)$. The magnitude of B is irrelevant.

Consider two data, for example, in a hyper-box relevant to a given input, whose outputs in a given dimension are o_d and are o_e with degrees h_d and h_e respectively. As illustrated in Fig. 3, their contribution to output in that dimension is as follows (using center-of-gravity calculations):

$$[h_d(2-h_d)o_d \; + \; h_e(2-h_e)o_e] \; / \; [h_d(2-h_d) \; + \; h_e(2-h_e)]$$

This weighting calculation is for two relevant data, but the computation for more than two is similar. This computation is performed for each output dimension.

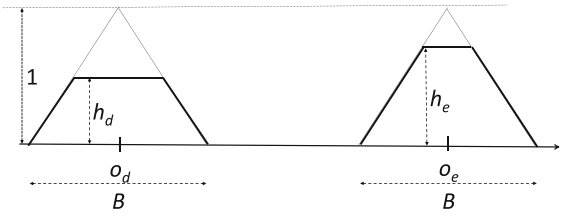

Fig. 3. Computation of Output

Proximity in input and output is implicit in the formulation of MaRz is because fuzzy calculations are based on it. This is easiest to calculate when the data are ordered. Consequently, all data—numerical, or converted as such—are first sorted. MaRz uses a table of indices, or "sorter" [12]. Each column of the sorter table contains the indices of the original dataset reordered as they would be if the dataset were sorted by the corresponding feature. Figure 4 illustrates this.

The sorter allows MaRz to treat the data in each dimension of as if sorted without altering the data set. Sorting is a potentially time-consuming aspect of MaRz, but it was not noticeably detrimental in our experiments compared with the timing of other machine learning methods. Indeed, speed is a MaRz asset.

MaRz avoids traditional time-consuming computation—performing backpropagation and adjusting weight parameters. It accomplishes this at the expense of creating a traditional model; instead, its training consists of sorting the data to prepare for queries rather than representing it or abstracting from it. Queries are serviced directly with the sorted dataset, via the table of indices. Given an input, we interrogate the latter for relevant records.

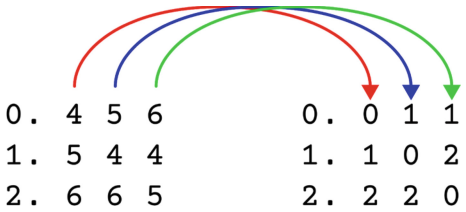

Fig. 4. Generating a table of sorting indices (right) from a 3-dimensional dataset (left)

MaRz currently uses *argsort*, an efficient NumPy sorting algorithm. Sorting facilitates binary search when creating the hyper-box. The sort operations are required once, as part of the preprocessing; binary search is performed for each query. With n the number of records and m the feature count, the time complexity of this binary search is $O(m \log(n))$ since MaRz must search on every feature. A list of record numbers within the given constraints for each feature is returned. These lists must be intersected—a process that requires nontrivial time. We found it effective to begin searching restrictively, then to expand the hyper-box as required.

3 Results

MaRz has been run for classification and regression problems on a variety of datasets and under several experimental conditions. MaRz is comparable to the best accuracy results for those datasets, and is fast (sometimes multiple times faster), real-time, and explainable.

MaRz allows us to use the entire dataset S as a test set as follows. For every record R, we query for R using $S \backslash R$ as the dataset. In other words, we ask *how does MaRz treat R if all it knows is S\R?* MaRz is fast enough to permit this process within practicable time. Traditional training/testing splits are not required.

We sometimes find it effective to select hyper-boxes with just one record for classification and two for regression. However, as illustrated below, the user can remove one of these records from the data set and re-run the application, repeating this process to gain confidence in the output.

The aggregate results are described next, with comparatively superior results shown in bold.

3.1 Airfoil Noise Dataset: Regression Example

MaRz produces good results on regression tasks, which we tested on NASA's Airfoil Noise dataset [14]. This involves 1500 instances and 6 measurements, including airfoil sound pressure levels in decibels under various conditions. The measurements take place in a wind tunnel. The MaRz results were compared with random forest and stochastic gradient tree boosting models are shown in Table 1, along with their computation times [16]. MaRz is about 0.04 less accurate measured by R-square value but 6- to 8 times faster than these, executed on a laptop computer.

Table 1. MaRz regression comparative *Airfoil* results

Model	R-Square Value	Computation time (seconds)
Random Forest	0.9291	4
Stochastic Gradient Tree Boosting	**0.9686**	3
MaRz	0.9315	**0.5**

3.2 Heart Disease Comparison: Transparency and Classification Example

To exemplify the transparency and explainability that MaRz provides, we compared it to a standard neural net implemented in TensorFlow, showing the type of information returned for queries. We used the UCI heart disease dataset [20], which consists of 468 records—303 heart disease patients and 165 healthy participants—each with 11 features.

The neural net made a prediction comparable in accuracy to MaRz but supplied no supporting information. MaRz returns not only a prediction, but also the most relevant records from which the prediction was derived. In other words, the diagnostician can explain to a patient, "among the data, *this* case was closest to yours and the conclusion for it was …" or "*these* 10 cases were closest to yours and the conclusion for them was …". This explanation is not abstracted or otherwise processed; it gives the user plain information. It is strongest when the details and provenance of the comparison cases are independently verified (in this example application, when the individual cited is found by direct means to have or not have heart disease).

To confirm MaRz's result, the supporting record(s) can be removed from the dataset and the query re-run to find the next-best supporting data and so on. The first column of Table 2 contains the first query in our experiment. (It was actually record 100 in the original dataset, but removed). MaRz output agrees with that record's output (*yes, heart disease*). Its explanation consists of exhibiting row (record) originally numbered 564.

To increase confidence—at the user's request—that record is removed from the dataset and an additional record is exhibited as further explanation. The second output, obtained after removing row 564 from the data, was derived from row 451 in the original dataset, and again correctly predicted the occurrence of heart disease. This process of iteratively finding additional results and being able to see what unprocessed data informed those results, can help to build user confidence in the system as a diagnostic tool.

If the output for the second-most relevant data *contradicts* that of the most relevant, the user's confidence is correspondingly reduced: more, and richer, data may be called for.

Table 2. Output for two rounds of confidence-building

Attribute (input)	Input to MaRz	MaRz explains: row 564	MaRz explains: row 451
Age	*65*	*61*	*60*
Resting BP	*130*	*141*	*140*
Cholesterol	*275*	*292*	*281*
Fasting BS	*0*	*0*	*0*
Max HR	*115*	*115*	*118*
Oldpeak	*1.0*	*1.7*	*1.5*
Sex	*Male*	*Male*	*Male*
Pain Type	*ASY*	*ASY*	*ASY*
Resting ECG	*ST*	*ST*	*ST*
Exercise Angina	*Yes*	*Yes*	*Yes*
ST Slope	*Flat*	*Flat*	*Flat*
Output	**Yes**	**Yes**	**Yes**

An additional process that builds confidence is to remove each successive explanatory record (and no other), then compare MaRz's output for that record's input with the record's output value. In other words, to check the veracity of each explanatory record in the context of dataset. The longer that these outputs agree, the more confident the user becomes.

Liquid Time-Constant Experiment Comparisons: Real-time Classification and Regression Examples

Table 3 shows that MaRz produces superior results on three tasks selected by Hasani et al. (2021), who used liquid time-constant (LTC) networks, a real-time machine learning approach [17]. The Occupancy dataset [18] (20,560 instances, each with 6 features) and Ozone [19] (2,536 instances, each with 72 features) dataset are binary classification tasks; the Power dataset [20] (2,075,259 instances, each with 9 features) concerns regression. Due to the size of the Power dataset, the results were aggregated from several executions, each testing 1% of the records in a 20% subset of the total dataset.

MaRz produced superior performance with all three tasks, as shown in bold.

Table 3. Comparison of MaRz with Hasani et al.'s Liquid Time-Constant results

Dataset	Metric	LTC	MaRz
Occupancy	accuracy	94.63%	**99.37%**
Power	squared-error	0.642	**0.005**
Ozone	F1-score	0.302	**0.3148**

4 Limitations, Summary, Conclusions, and Future Work

4.1 Limitations

To determine an appropriate hyper-box, MaRz must intersect lists, which is potentially time-consuming. The sorting step also presents a potential time penalty. As the experimental results show, however, these have not slowed the experiments unacceptably, but they remain potential limitations for larger data sets than those we used.

The benefit of MaRz lies in its combination of speed, accuracy, and explainability (it is not claimed to be supreme with respect to all these properties).

Since MaRz results are closely tied to individual records, data cleaning is especially important, including pruning of outliers. The process described above for the UCI heart disease dataset deals with mitigating this issue.

4.2 Summary, Conclusions, and Future Work

MaRz is a novel machine-learning approach that fuzzifies records. Its learning consists of sorting—essentially preparing the data to answer queries on the fly. It performs competitively, in speed, accuracy, and particularly explainability, on several datasets. MaRz explains its conclusions by exhibiting the data that most influence its output—it does not synthesize explanations.

MaRz can run in real time and may be faster than some well-known contemporary learning methods, as evidenced by several tests on data sets in the literature. Its accuracy is competitive.

MaRz may not perform as well as other architectures for multi-category classification, especially when the categories do not relate to each other (e.g., *hats*, *gloves*, *shoes*). Fuzzy methods do better for categories with a sense of order such as *small*, *medium*, and *large*. We have yet to measure the extent of this apparent limitation.

Future work on MaRz will include a close measurement and analysis of (a) its effectiveness for increasingly large datasets and (b) the real-world effectiveness of its instance-based style of explanation. We have compared MaRz with six different machine learning methods applied to six different applications. While experimental results are promising, it remains to prove the superiority of MaRz with respect to one of its benefits alone. Accordingly, we plan to compare it with more machine learning approaches with this in mind.

References

1. Deiana, A.M., et al.: Applications and techniques for fast machine learning in science. Frontiers in Big Data. **5** (2022). https://doi.org/10.3389/fdata.2022.787421
2. Rudin, C., Radin, J.: Why are we using black box models in ai when we don't need to? A lesson from an explainable AI competition. Harvard Data Science Review **1**(2) (2019). https://doi.org/10.1162/99608f92.5a8a3a3d
3. Adadu, A., Berrada, M.: Peeking inside the black-box: a survey on explainable artificial intelligence (XAI). IEEE Access. **6**, 52138–52160 (2018). https://doi.org/10.1109/ACCESS.2018.2870052

4. Rudin, C.: Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. Nat Mach Intell **1**, 206–215 (2019). https://doi.org/10.1038/s42256-019-0048-x
5. Posmakov, N.P., Emelyanenko, A.S., Kireev, V.S.: Fuzzy cognitive map ensembles to solve regression and time series tasks. In: Conference of Russian Young Researchers in Electrical and Electronic Engineering (ElConRus), pp. 406–409 (2002). IEEE, Saint Petersburg, Russian Federation. https://doi.org/10.1109/ElConRus54750.2022.9755460
6. Tanuwijaya, K., Chen, S.-M.: A new method to forecast enrollments using fuzzy time series and clustering techniques. In: The 2009 International Conference on Machine Learning and Cybernetics, pp. 3026–3029 (2009). IEEE, Baoding, China. https://doi.org/10.1109/ICMLC.2009.5212604
7. Pomares, H., Rojas, I., Fernandez, F.J., Anguita, M., Ros, E., Prieto, A.: A new approach for the design of fuzzy controllers in real time. In: FUZZ-IEEE'99. IEEE International Fuzzy Systems 1999. Cat. No.99CH36315, **1**, pp. 522–526 (1999). IEEE, Seoul, Korea (South). https://doi.org/10.1109/FUZZY.1999.793295
8. Hong, T.-P., Lee, C.-Y.: Data structure for a fuzzy machine learning algorithm. In*:* 6th International Fuzzy Systems Conference, **3**, pp. 1315–1319 (1997). IEEE, Barcelona, Spain. https://doi.org/10.1109/FUZZY.1997.619735
9. Kumar, A.: What is Explainable AI? Concepts & Examples. Data Analytics. https://vitalflux.com/what-is-explainable-ai-concepts-examples/. Accessed 14 July 2023
10. Shevskaya, N.V.: Explainable artificial intelligence approaches: challenges and perspectives. In: International Conference on Quality Management, Transport and Information Security, Information Technologies (IT&QM&IS) 2001*,* pp. 540–543 (2021). IEEE, Yaroslavl, Russian Federation. https://doi.org/10.1109/ITQMIS53292.2021.9642869
11. Rosenfeld, A.: Better metrics for evaluating explainable artificial intelligence. In: 20th international conference on autonomous agents and multiagent systems, pp. 45–50 (2021). Virtual.
12. numpy.searchsorted – NumPy v1.26 Manual. https://numpy.org/doc/stable/reference/generated/numpy.searchsorted.html. Accessed 20 April 2024
13. Pachipala, Y., Maddipati, H.C., Udimudi, P.S., Thota, V., Sree, V. N., Burra, L.R.: Iris flower classification by using random forest in AWS. In: 6th International Conference on Intelligent Computing and Control Systems (ICICCS), pp. 1772–1777 (2022). IEEE, Madurai, India. https://doi.org/10.1109/ICICCS53718.2022.9788415
14. Brooks, T., Pope, D., Marcolini, M.: Airfoil self-noise. UCI Machine Learning Repository (2014). https://doi.org/10.24432/C5VW2C
15. Patri, A., Patnaik, Y.: Random forest and stochastic gradient tree boosting based approach for the prediction of airfoil self-noise. Procedia Computer Science, **46**, 109–121 (2015). ISSN 1877–0509. https://doi.org/10.1016/j.procs.2015.02.001
16. Janosi, A., Steinbrunn, W., Pfisterer, M., Detrano, R.: Heart disease. UCI Machine Learning Repository (1988). https://doi.org/10.24432/C52P4X
17. Hasani, R., Lechner, M., Amini, A., Rus, D., Grosu, R.: Liquid time-constant networks. In: AAAI Conference on Artificial Intelligence **35**(9), pp. 7657–7666 (2021). https://doi.org/10.1609/aaai.v35i9.16936
18. Candanedo, L.: Occupancy detection. UCI Machine Learning Repository (2016). https://doi.org/10.24432/C5X01N
19. Zhang, K., Fan, W., Yuan, X.: Ozone level detection. UCI Machine Learning Repository (2008). https://doi.org/10.24432/C5NG6W
20. Hebrail, G, Berard, A.: Individual household electric power consumption. UCI Machine Learning Repository (2012). https://doi.org/10.24432/C58K54