DOI : https://doi.org/10.56452/7-5-205

ISSN: 0974-5823 Vol. 7 No. 5 May, 2022

International Journal of Mechanical Engineering

A Fuzzy Autoencoder-Based Self-Organizing Approach to a Hierarchical Fuzzy Logic System

Sudhanshu Maurya

Asst. Professor, School of Computing, Graphic Era Hill University,

Dehradun, Uttarakhand India 248002

Abstract: This study on Autoencoders focuses mostly on the challenge of dealing with unpredictability in neural networks. Recently, auto encoders and deep stack of autoencoders (DSAE) have found applications in machine learning.Recent years have seen significant progress in solving the challenges of decreasing dimensionality and data compression. Autoencoders, like conventional neural networks, were deterministic architectures that struggle with coping with data uncertainty, despite their importance in many practical applications. In this research, we provide a fuzzy method for automatically incorporating qualitative fuzzy data information into the input layer, hence lessening the degree of ambiguity in Autoencoder stacks. To this end, we may add a fuzzy layer-0 to our Autoencoder stack and use it to supplement the crisp data set with some fuzzy information. The method is potentially applicable to any neural network design since it is transparent for both the network and the user. The provided findings are quite promising, and represent a significant advancement, particularly when working with noisy data..

Keywords: Autoencoders ,Deep Stacks, fuzzy, convolutional neural networks

Introduction

Over the past few decades, the "dark side of dimensionality" [1, 2] has become a pressing issue in a number of scientific and technological disciplines. Many disciplines are beginning to recognise the value of dimensionality reduction methods in light of the growing significance of big data and real-time applications. Neural networks have been essential in this field ever since Autoencoders were developed as a tool for feature extraction [3, 4], data compression [5, 6], noise reduction [6, 7], and so on. The background and evolution of autoencoders [10] are briefly covered in [8], [9]. The mathematical foundations of both linear and non-linear autoencoders are outlined in [8]. Several state-of-the-art results have been accomplished since the introduction of Deep Autoencoders [11–13]. Even though mechanisms for stochastic behaviours are widely used, data uncertainty remains a significant challenge for neural networks. As is the case with the vast majority of current neural network and deep neural network architectures, uncertainty is a problem of enormous significance for real-world circumstances, especially when training and testing are independent and sequential activities. It is usual for performance and accuracy to decline when dealing with circumstances marked by large amounts of noise and ambiguity, or when testing data significantly deviates from training data. Books on this subject have been published by a number of writers. When dealing with ambiguous or noisy data, dropout techniques are often the best option [14, 15]. Gal et al. [16] expanded on this concept and used it as a Bayesian Approximation to define the uncertainty of deep learning. To demonstrate the efficacy of their methods, they employed the MNIST database of handwritten digits [17, 18]. A novel approach to learning a distribution of probabilities was developed by Blundell et al. [19], which not only obtained results competitive with dropout but also demonstrated how the acquired uncertainty aided in improving generalisation. As an alternative to dropping out and batch normalisation, Li et al. [20] presented Random Gradient Markov Chain Monte Carlo for training Deep Neural networks. In addition to confirming the improved identification accuracy, they also confirmed the approach's scalability. When it comes to dealing with data uncertainty, fuzzy systems excel as both universal approximators and reliable instruments. The first practical implementations of neural networks and fuzzy systems emerged about 30 years ago [21]. Despite the current popularity of deep neural networks, the full potential of integrating the two concepts has yet to be explored. The authors of this research suggest a hierarchical fused architecture to address the limitations of neural network systems based on a single representational model. Both the fuzzy plane and the network's neural layer had the same extraction design. They demonstrated that a non-sequential combination of fuzzy and neural network systems may provide significantly improved findings for segmentation in medical pictures. This joint-learning approach shows promise in a wide range of network architectures and computational settings. We propose an extended structure for a set of Autoencoders that uses the well-known qualities of fuzzy systems for handling data uncertainty to enhance the reduced form of the input, particularly when working with noisy data. To do this, the input to the Autoencoder stack is "fuzzed" by adding a fixed number of new dimensions. The improved method is compatible with the standard Autoencoder stacks and may be adjusted to suit a wide variety of other factors.

The three phases of fuzzy system design are fuzzification, inference, and defuzzification. To create tags for fuzzy sets, fuzzification methods take the input and transform it into meaningful words and phrases. A fundamental part of every inference system is its input-output rule base. The set of rules is a normalised set of input/output combination and membership functions.All the results from the inference system are converted into a machine-readable format by using defuzzification.

The original fuzzy system has spawned several offshoots. Neuro-fuzzy systems, fuzzy clustering, fuzzy logic image processing, etc., are all examples. Fuzzy logic was further developed into neuro-fuzzy logic with the advent of neural networks. Computing difficulties like as uncertainty, imprecision, and ambiguity may be effectively resolved and evaluated with the help of fuzzy logic. But methods like neural networks can only process data that is both consistent and well-defined. Their primary shortcoming is that fuzzy logic restricts how much data can be stored. Neuro-fuzzy is able to deal with complex calculations because to its foundation in established procedures. Since neuro-fuzzy systems rely on very specific inputs and outputs to function, it has been difficult to apply them to problems with high input-output dimensions. A number of researchers [8, 12] recently created and are further refining deep neural network techniques to solve the issue of picture categorization in AI. Irregular and imprecise input is common in many real-world circumstances, yet neural networks can't handle it. To cope with ambiguity and uncertainty, neuro-fuzzy logic methods have been proposed.

Neuro-fuzzy logic makes use of white-box techniques. These techniques not only enhance comprehension but also promote interaction between the fields of mathematics and language structure. However, their applicability has been constrained by finite rule and datadimensions, making them unsuitable for use in the design of extremely complex systems.

These limits have proved to be a serious obstacle in several highly complex real-world contexts.

This reduces the system's openness and generalizability. As the number of input parameters, datasets, or input-output combinations and mathematical connections grows, the constraints of the input-output dimensions for a fuzzy or neuro-fuzzy system must be tightened accordingly.

The idea of merging hierarchical systems with fuzzy logic to solve the difficulties of generalisation, transparency, and dimensionality has been proposed by a number of scientists. Instead of a single, large system, this one will be broken up into smaller, low-dimensional subsystems that interact in a hierarchical fashion. The hierarchical system is described by a model with many inputs and a single output. Without compromising on generality, MIMO systems may be decomposed into a large number of smaller MIMO subsystems. Hierarchical systems are often used for a variety of purposes, including classification, grouping, planning, tracking, etc. The limitation of the fuzzy logic rule-base when dealing with large image datasets is addressed in this research via the introduction of a technique for creating hierarchical fuzzy systems that makes use of picture thresholding.

The size of the system's rule base may be reduced using this way without compromising performance.

Counting the number of rules in a system has traditionally been a stand-in for its level of complexity. Because more inputs necessitate more rules, system complexity increases as input counts rise. Due to this limitation, fuzzy systems cannot be employed in high-dimensional, practical applications, such as those requiring very large images. The total number of rules increases as a result of this because of the positive correlation between the size of the system's input parameters and its complexity.

In recent years, the hierarchical fuzzy system has shown effective in helping researchers and practitioners overcome some of the drawbacks of more conventional fuzzy and neuro-fuzzy approaches. The authors have not come across any other work that synthesises small data subsets from large image datasets using hierarchical fuzzy logic. Several researchers [1] have demonstrated that the hierarchical structure leads to fewer rules and thus makes the system simpler than conventional fuzzy logic. The calculation time and complexity are both reduced by the hierarchical system due to the fine-tuning of the quantity of rules required for each fuzzy logic unit. This motivates us to suggest hierarchical fuzzy as a viable approach to solving problems with large image collections.

Neural networks are one example of a current-day technique that has the limitation of only being able to process input that is known and definite; they cannot handle material that is unknown, imprecise, or ambiguous. The ability of deep learning algorithms to handle real-world applications, some of which are critical to safety and security, is limited as a result of these limitations on system efficiency and output. This requires better data visualisation, which in turn requires greater data collection, analysis, and simplification. Problems with transparency are common in deep learning applications. This lack of transparency makes the data less accessible and useful to other users. In light of these limitations, fuzzy logic provides a strong answer and more precise assessment of human interpretations by factoring in uncertainty, imprecision, and ambiguity.

There is currently no well-established procedure for creating type-1 and type-2 fuzzy inference systems with a hierarchical structure. In this study, a hierarchical technique is used to analyse type-1 and type-2 fuzzy inference systems. Type-1 and type-2 fuzzy inference systems use distinct notations for rules and membership functions. Type-1 fuzzy inference systems have more precise membership definitions than type-2 fuzzy inference systems. Due to its enhanced flexibility in a wide variety of contexts, type-2 fuzzy excels above type-1 fuzzy. Since there are so many design parameters in a type-2 fuzzy [9], the type-reduction method is required during the defuzzification phase, and the resulting computation time is quite long. Advanced features for paying members. Type reducer defuzzification is used to create the outcome by averaging the intervals at each iteration. Type-1 fuzzy inference is the subject of this investigation, however the approach used might easily be applied to type-2 fuzzy inference with minor adjustments.

Literature Review

Qingshuo Zhang et.al.,(2021) While the extreme learning machine-based multi-label learning algorithm has strong efficiency and generalisation abilities, its classification power is low because it does not take into account the relationship between features and labels. In light of this, this work proposes a novel method for multi-label classification: the kernel extreme learning machine autoencoder (KELM-AE-fuzzy). The efficiency of the proposed approach is confirmed by experimental results showing that KELM-AE-fuzzy outperforms existing multi-label algorithms on a number of different multi-label datasets.

Kutay Bölat et.al.,(2020) The recent breakthroughs of Deep Learning (DL) across a wide range of application domains have spawned a slew of "how" and "why" inquiries. If DL approaches are interpretable, they should be able to give some level of explanation, allowing us to answer these issues. In this study, we offer a DL framework for the development of a new DL based Fuzzy Classifier (FC) by combining the disentanglement and linguistic representation benefits of -Variational Autoencoder (VAE) and Fuzzy Sets (FSs). We begin by outlining our design process for building the DL-FC, which consists of the encoder layer of -VAE, a Fuzzy Logic System (FLS), and finally a softmax layer. The -VAE is taught to extract meaning from high-dimensional datasets. Clustering the -VAE's latent space allows for the extraction of FSs. The antecedents of the DL-trained FLS are defined using the FSs.

Bruno Costa et.al.,(2019) In this study, we focus on Autoencoders and the challenge of dealing with uncertainty in neural networks. There has been a lot of interest in, and some promising outcomes from, applying the relatively new ideas of "Autoencoders" and "Deep Stacks of Autoencoders" (DSAE) to the challenges of reducing dimensionality and data compression. Auto encoders are predictable structures that, like standard neural networks, aren't great at handling data uncertainty, despite the fact that this is a crucial part of many practical applications. In this research, we present a fuzzy method for lowering the uncertainty in Autoencoder stacks by the incorporation of a layer of automatically generated qualitative fuzzy data information. In particular when confronting noisy data, the findings provided here are quite promising and represent a significant improvement.

Yevgeniy Bodyanskiy(2018) In this paper, we present a neo-fuzzy neuron-based autoencoder. Its learning process was also optimised; it makes advantage of the quadratic criteria. It's feasible to integrate such systems into an existing deep learning infrastructure. The proposed autoencoder features a faster learning rate and fewer tuning parameters than popular "bottle neck" autoencoders. The usefulness of the suggested strategy has been shown using a variety of benchmarks and real-world data sets.

Hierarchical Fuzzy Systems

The entities in a hierarchical system are positioned "above," "below," or "on the same plane as" one another, as defined by the wiki. When the dimensions and amount of the dataset are enormous, it is more challenging to find a solution to a complex issue utilising a fuzzy or neuro-fuzzy approach. One common method for dealing with this issue is to partition the system to individual subsystems and arrange them in a particular fashion, where each component is defined by a fuzzy system. These kinds of arrangements, known as "hierarchical fuzzy systems" [2], seem to be the most useful and productive.

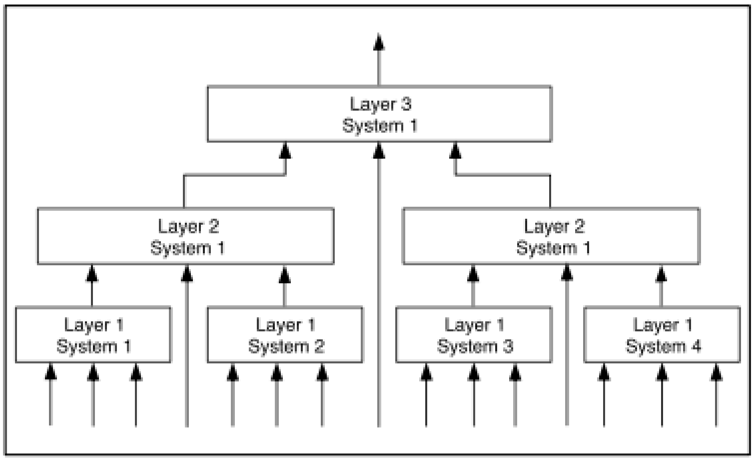

Figure 1. Cascaded structure of hierarchical systems

Some scientists have found that adding a hierarchical system on top of fuzzy logic improves the accuracy of their findings. Hierarchical fuzzy systems have several applications, including classification, grouping, planning, tracking, etc. MISO stands for "multi-input, single-output," which describes the system, however larger MIMO systems may be partitioned into many smaller MISO subsystems. The inputs themselves are made available to the lowest level of the hierarchy, while the subsequent levels are connected to both the preceding level's output and the inputs themselves. Depending on the underlying structure, the hierarchical fuzzy may be defined at several levels of detail. Each level has many fuzzy logic units.Incremental, aggregated, and cascade approaches are the three most common ways to handle hierarchical structures. As the following structures consist of multiple stages, they are often referred to as "multi-stage structures." Assume that the parameters for each level are determined by a single fuzzy system, and that the output from one level is used as input for the next. To maintain the system's precision, it's best to give first thought to the most important inputs and then to the less important ones. The tiered hierarchy is shown in Fig. 1.

Autoencoders

The Autoencoder is an example of neural network whose output is a faithful representation of its input. Its architecture consists of a single input layer, a single hidden layer, and a single output layer. There are often fewer nodes in the hidden layer than in the input and output layers. Both the input and output layers are of same size. Figure 2: Autoencoder Schematic.

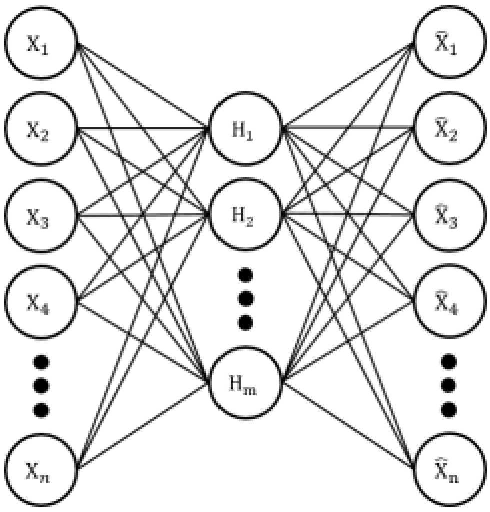

Figure. 2. Representation of a traditional Autoencoder

Among the many unsupervised learning-based applications for autoencoders are extraction of features, reducing dimensionality, and data compression. The objective is to produce as near to the input as possible from the compressed representation. Designing deep Autoencoder networks requires stacking, which our architecture makes easy [11, 23]. For guided and semi-supervised learning, a sequential layer may be added, or it can be utilised as part of an additional structure, such as a classifier.

Deep stacks of Autoencoders

When the outputs of one layer of a neural network are completely linked to the inputs of the next layer, we get what is called a deep stack of autoencoders [6]. Autoencoder stacking may be accomplished in a manner very similar to that of RBM stacking [11]. Each stacked Autoencoder may be trained independently in an unsupervised fashion, and together they can create additional non-linear layers. Figure 2 depicts the first step of the stacking process, which is training a single Autoencoder. Figure 3 depicts the structure after training, when the decoder component may be discarded. A freshly trained Autoencoder's hidden layer may be used as input for a subsequent Autoencoder; this allows the characteristics learnt in a lower-dimensional dataspace to be transferred to a higher-dimensional one. As can be seen in Figure 3, this novel Autoencoder is taught independently. This procedure is repeated with each successive buried layer containing fewer units than its predecessor.

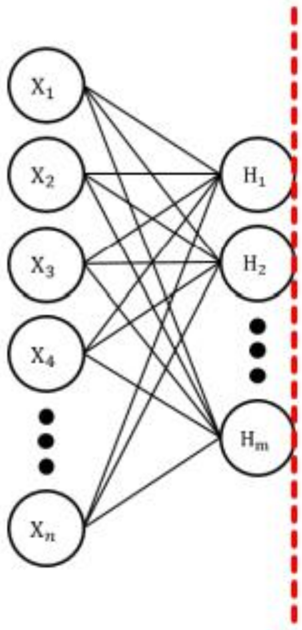

Figure. 3. Process of stacking Autoencoders

Conclusion

In this paper, we provide a new design for HFSs that combines a fuzzy auto encoder with a fuzzy partition. The method generates all of the fuzzy sets and fuzzy rules automatically, beginning with a rule set of zero. The input data's distribution characteristics may be more accurately reflected with the use of a refined method for standardising box plot data. The HFS is gradually trained over time using a fuzzy auto encoder, which ensures the efficacy of the hidden layer variables and provides interpretability. HFS simplifies matters by cutting down on the number of rules and complexity often associated with fuzzy logic systems. We put the proposed HFS through its paces on three separate sets of regression data.

References

1. Q. Zhang, E. C. C. Tsang, M. Hu, Q. He and D. Chen, "Fuzzt Set-Based Kernel Extreme Learning Machine Autoencoder for Multi-Label Classification," *2021 International Conference on Machine Learning and Cybernetics (ICMLC)*, Adelaide, Australia, 2021, pp. 1-6, doi: 10.1109/ICMLC54886.2021.9737260
2. K. Bölat and T. Kumbasar, "Interpreting Variational Autoencoders with Fuzzy Logic: A step towards interpretable deep learning based fuzzy classifiers," *2020 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, Glasgow, UK, 2020, pp. 1-7, doi: 10.1109/FUZZ48607.2020.9177631.
3. Asl, R. M., Palm, R., Wu, H., & Handroos, H. (2020). Fuzzy-based parameter optimization of adaptive unscented Kalman filter: Methodology and experimental validation. IEEE Access, 8, 54887–54904. https://doi.org/10.1109/ACCESS.2020.2979987
4. Kamthan, S., & Singh, H. (2020). Hierarchical fuzzy logic for multi-input multi-output systems. In IEEE Access, 8, 206966–206981. https://doi.org/10.1109/ACCESS.2020.3037901
5. Lin, Y., & Songcan, C. (April 2020). A Centroid auto-fused hierarchical fuzzy c-means clustering. IEEE Transactions on Fuzzy Systems, 99, sn 1941-0034

6. B. Costa and J. Jain, "Fuzzy Deep Stack of Autoencoders for Dealing with Data Uncertainty," *2019 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, New Orleans, LA, USA, 2019, pp. 1-6, doi: 10.1109/FUZZ-IEEE.2019.8859022.
7. Vedaldi, A., Lux, M., & Bertini, M. (2018). MatConvNet: CNNs are also for MATLAB users. ACM SIGMultimedia Records, 10(1), 9–9. https://doi.org/10.1145/3210241.3210250
8. Jarraya, Y., Bouaziz, S., Hagras, H., & Alimi, A. M. (2018). A multi-agent architecture for the design of hierarchical interval Type-2 beta fuzzy system. IEEE Transactions on Fuzzy Systems, 27(6), 1174–1188. https://doi.org/10.1109/TFUZZ.2018.2871800
9. Kamthan, S., Singh, H., & Meitzler, T. UAVs: On development of fuzzy model for categorization of countermeasures during threat assessment SPIE defense+ Security, pp. 1019518–1019518,International Society for Optics and Photonics. (2017).
10. Sun, L., & Huo, W. (2016). Adaptive fuzzy control of spacecraft proximity operations using hierarchical fuzzy systems. IEEE/ASME Transactions on Mechatronics, 21(3), 1629–1640. https://doi.org/10.1109/TMECH.2015.2494607
11. L. Sun Liang, H. Wei, "Adaptive fuzzy control of spacecraft proximity operations using hierarchical fuzzy systems." IEEE/ASME transactions on mechatronics 21, no. 3: 1629-1640, 2016
12. Sun, L., & Huo, W. (2016). Adaptive fuzzy control of spacecraft proximity operations using hierarchical fuzzy systems. IEEE/ASME Transactions on Mechatronics, 21(3), 1629–1640. https://doi.org/10.1109/TMECH.2015.2494607
13. W. Wang, Y. Huang, Y. Wang and L. Wang, "Generalized autoencoder: A neural network framework for dimensionality reduction", Proceedings of the IEEE conference on computer vision and pattern recognition workshops, pp. 490-497, 2014
14. Miranda, V., Krstulovic, J., Keko, H., Moreira, C., & Pereira, J. (2012). Reconstructing missing data in state estimation with autoencoders. IEEE Transactions on Power Systems, 27(2), 604–611. https://doi.org/10.1109/TPWRS.2011.2174810
15. Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. R. (2012). Improving neural networks by preventing coadaptation of feature detectors.
16. Gal, Y., & Ghahramani, Z. Dropout as a bayesian approximation: Representing model uncertainty in deep learning, international conference on machine learning, 201 (pp. 1050–1059).