Proceedings Paper:

Xi, Z. and Panoutsos, G. (2019) Interpretable machine learning: Convolutional neural networks with RBF fuzzy logic classification rules. In: 2018 International Conference on Intelligent Systems. 2018 International Conference on Intelligent Systems, 25-27 Sep 2018, Funchal, Madeira, Portugal. IEEE , pp. 448-454. ISBN 978-1-5386-7097-2

https://doi.org/10.1109/IS.2018.8710470

2018 International Conference on Intelligent Systems (IS)

Interpretable Machine Learning: Convolutional Neural Networks with RBF Fuzzy Logic Classification Rules

Zhen Xi
Department of Automatic Control and Systems Engineering
University of Sheffield
Sheffield, United Kingdom
zhen.xi@sheffield.ac.uk

George Panoutsos
Department of Automatic Control and Systems Engineering
University of Sheffield
Sheffield, United Kingdom
g.panoutsos@sheffield.ac.uk

***Abstract*—A convolutional neural network (CNN) learning structure is proposed, with added interpretability-oriented layers, in the form of Fuzzy Logic-based rules. This is achieved by creating a classification layer based on a Neural-Fuzzy classifier, and integrating it into the overall learning mechanism within the deep learning structure. Using this new structure, one could extract linguistic Fuzzy Logic-based rules from the deep learning structure directly, which enhances the interpretability of the overall system. The classification layer is realised via a Radial Basis Function (RBF) Neural-Network, that is a direct equivalent of a class of Fuzzy Logic-based systems. In this work, the development of the RBF neural-fuzzy system and its integration into the deep-learning CNN is presented. The proposed hybrid CNN RBF-NF structure can from a fundamental building block, towards building more complex deep-learning structures with Fuzzy Logic-based interpretability. Using simulation results on a benchmark data-driven modelling and classification problem (labelled handwriting digits, MNIST 70000 samples) we show that the proposed learning structure maintains a good level of forecasting/prediction accuracy ($>$ 96% on unseen data) compared to state-of-the-art CNN deep learning structures, while providing linguistic interpretability to the classification layer.**

***Index Terms*—Fuzzy Logic, Deep Learning, Convolutional Neural Networks.**

I. Introduction

In data-driven modelling systems and methods, machine learning has received considerable attention in recent decades. Machine Learning focuses on applied maths and computing algorithms for creating 'computational machines' that can learn to imitate system behaviours automatically [1]. As a subarea of Artificial Intelligence (AI), using machine learning (ML) one could also construct computer systems and algorithms to improve performance based on what has already been experienced (empirical-based, learning from examples) [1], [2]. ML has emerged as a popular method for process modelling, also used in natural language processing, speech recognition, computer vision, robot control, and other applications [1]–[3].

Unlike traditional system modelling methods (physics-based, numerical etc.), machine learning does not require a dynamic process model but sufficient data, including input data and output data of a specific system, hence a class of machine learning algorithms can be considered as data-driven modelling methods that are able to capture static or dynamic process behaviour in areas such as manufacturing and biomedical systems among others. Gong et al. introduced a way to analysis time series signals and to create a human body model using CNNs [4]. Segreto et al. evaluated the correlation between wavelet processed time series signals and the machining conditions using neural networks [5]. Based on the type of modelling structures used, machine learning could be broadly viewed in two parts with - to a certain extent - unclear boundaries, which are statistical modelling and learning, and neural and other hybrid network structures [2]. In deep learning in particular [2], convolutional neural networks (CNNs) have been widely used [6]–[8]. CNNs are a kind of feedforward neural network using convolutional cores to process data in multiple arrays. Multiple arrays could be in the form of variable data modalities: 1D for time-domain signals, 2D for images, and 3D for videos [3].

Using CNN deep learning structures has been very successful for certain class of applications, for example Szegedy et al. proved a deep enough network can classify ImageNet [9] efficiently [6], and He et al. provided a model structure to build deep neural networks without considerable gradient loss [10]. Simonyan and Zisserman show that CNNs could be designed as even 'deeper' structures, and perform even better in ImageNet classification problems [11]. Deep CNN networks however, lack any significant interpretation, and act as 'black boxes' that predict/classify data well, and this is understandable given their deep structure and overall complexity. There is an opportunity therefore, to use the paradigm of Fuzzy Logic (FL) theory, and attempt to add linguistic interpretability to deep learning structures. Successful implementation would be beneficial to a variety of problems, in particular in cases where there is a need for human-machine interaction, such as in decision support systems for critical applications (healthcare,

978-1-5386-7097-2/18/$31.00 ©2018 IEEE

biomedical, high-value manufacturing etc.). For example, one could use FL theory to provide linguistic interpretation to classification tasks performed by deep learning networks.

There are existing attempts in the literature to combine FL with deep learning. Muniategui et al. designed a system in spot welding monitoring [12]. In this approach the authors use the deep learning network only as a method for data pre-processing, followed by the FL classifier as a separate process step. In an attempt to reduce data size without affect monitoring performance, a system based on deep learning and FL classification was introduced. Using a deep convolutional autoencoder, an image could be compressed from resolution of 120×120 to 15×15 without affecting the overall performance of the fuzzy classification methodology. Deng et al. introduced a FL-based deep neural network (FDNN) which extracts information from both neural representation and FL simultaneously [13]. It was shown that the FDNN has higher classification accuracy than networks based on NN or FL separately and then fusions the results from the two kinds of networks. The current gap in the research literature is in that the deep learning methodologies, when combined with FL, are not integrated together as a single system.

In this research work, a CNN-based deep learning structure is used as the fundamental building block of a data-driven classification network. For the first time in the literature, a FL-based layer (in the form of a hybrid Neural-Fuzzy network) is introduced as an integral part of the overall CNN structure, which acts as the main classification layer of the deep learning structure. Consequently, one could extract directly from the deep learning structure linguistic rules in the form of a FL rule-base. Via simulation results based on a popular benchmark problem/dataset we show that the proposed network structure performs as well as state-of-the-art CNN-based structures, hence there is no significant loss of performance by introducing the FL layer as part of the deep learning structure. In addition, the robustness of the learning process is also assessed by consecutively reducing the sample size.

II. Relation to existing theories and work

A. *Radial Basis Function Neural-Fuzzy layer*

RBF networks were formulated in [14] as a learning network structure. RBF networks can also be used efficiently as as a kernel function for a variety of machine learning methodologies, for example in Support Vector Machines to solve non-linear classification problems [15]. Similar to SVM, RBF networks could be implemented as FL-based systems [16].

In this section, for the benefit of the reader, the RBF-NF network is summarised (Fig. 1), and its relevance to the deep learning structure is shown, while full details of the fundamental RBF network as a data-driven model can be found in [16], [17].

Equation (1) represents a multiple-input and single-output (MISO) FL system with m system inputs and p number of rules, where $\mu_{ij}(x_j)$ defined in (2) is the Gaussian membership function of input x_j belonging to the i-th rule and c_{ij} and σ_{ij} are the centre and width of the Gaussian membership function

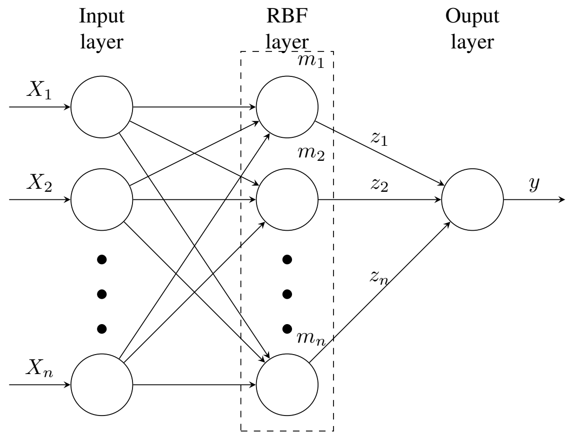


Fig. 1: RBF network structure

respectively [16]. The overall function $z(\vec{x})$ could be adjusted to represent one of the following three forms of FL-based systems:

- Singleton;
- Mamdani;
- Takagi-Sugeno.

In the proposed work, the overall system function $z(\vec{x})$ will be considered as a Singleton model. Fig. 1 depicts the structure of the RBF network, where X_n are the system's inputs, m_n the membership function of each rule-input combination, z_n the Tagaki-Sugeno polynomial function for each rule, and y the overall output of the system. Hence the output function takes the mathematical form shown in (3).

$$\begin{aligned} y &= \sum_{i=1}^{p} z_i \left[\frac{\prod_{j=1}^{m} \mu_{ij}(x_j)}{\prod_{i=1}^{p} \sum_{j=1}^{m} \mu_{ij}(x_j)} \right], \\ &= \sum_{i=1}^{p} z_i g_i(x), \end{aligned} \tag{1}$$

$$\mu_{ij}(x_j) = \exp\left(-\frac{(x_j - c_{ij})^2}{\sigma_{ij}^2} \right), \tag{2}$$

$$z_i = \sum_{j=1}^{p} b_{ij} x_j. \tag{3}$$

Equation (2) could be expressed in vector form, as follows (which is also the expression for a RBF in i dimensions):

$$m_i(\vec{x}) = \exp\left(-\|\vec{x} - \vec{c}_i\|^2 / \vec{\sigma}_i^2 \right), \tag{4}$$

thus this FL system could be written as:

$$y = \sum_{i=1}^{p} z_i m_i(x) \Big/ \sum_{i=1}^{p} m_i(x), \tag{5}$$

$$= \sum_{i=1}^{p} z_i g_i(x), \tag{6}$$

where

$$g_i = \left[\frac{\prod_{j=1}^{m} \mu_{ij}\left(x_j\right)}{\prod_{i=1}^{p} \sum_{j=1}^{m} \mu_{ij}\left(x_j\right)} \right],$$
$$= m_i\left(x\right) / \sum_{i=1}^{p} m_i(x). \quad (7)$$

A representative CNN structure for image classification would contain several layers, grouped in a way to perform specific tasks. Fig. 2 demonstrates a typical CNN architecture. The first few layers would be multiple pairs of convolution layers and pooling layers. The size of these convolution windows can be different, which ensure convolution layers can extract features in different scales. The pooling layers are proposed to subsample features into a smaller size, where a max pooling method is generally used. Then, fully connected layers would also be used, in which neurons are fully connected to all outputs from the previous layer. These layers also convert the data structure from a multiple-layer structure to a vector form. Rectangular linear units (ReLUs) would normally be the activation function of the convolution layers as well as in the fully connected layers as these can provide non-linear properties to those layers and are also convenient for the calculation of the error backpropagation [18]. To avoid exploding and vanishing gradients in deep networks, batch normalisation can also be applied in every layer [19]. CNNs are not considered as convex functions, which means parametric optimisation for CNNs is challenging, hence numerous optimisation strategies have been developed [20], such as stochastic gradient descent (SGD), Nesterov momentum [21], and adaptive subgradient (Adagrad) methods [22].

Fig. 3 depicts the overall structure of the CNN deep network. This model was designed to use 28×28 pixel greyscale images as input. After two convolutional layers, a max pooling layer was added. The dropout layers were applied to avoid overfitting. The Flatten layer was added to convert data structure into vectors, and two Dense layers are fully connected layers. All activiation functions in this model were ReLUs. The loss function of this model was cross entropy loss function, which is widely used in CNNs [6], [7]. In the proposed research work, the adaptive subgradient method was applied to perform the learning task, to take advantage of its fast convergence properties. In order to achieve a good balance between training speed and avoidance of overfitting the batch size was chosen as 128.

Table I shows the architecture of the designed CNN.

TABLE I: Basement CNN architecture

type	patch size/stride	output size	parameters
convolution	$3 \times 3/0$	$26 \times 26 \times 32$	320
convolution	$3 \times 3/0$	$24 \times 24 \times 64$	18496
maxpooling	$2 \times 2/0$	$12 \times 12 \times 64$	0
dropout (25%)		$12 \times 12 \times 64$	0
flatten		9216	0
linear		64	589888
dropout (50%)		128	0
linear		10	1290
softmax		10	0

TABLE II: FL RBF-CNN architecture

type	patch size/stride	output size	parameters
convolution	$3 \times 3/0$	$26 \times 26 \times 32$	320
convolution	$3 \times 3/0$	$24 \times 24 \times 64$	18496
maxpooling	$2 \times 2/0$	$12 \times 12 \times 64$	0
dropout (25%)		$12 \times 12 \times 64$	0
flatten		9216	0
linear		64	589888
dropout (50%)		64	0
RBF		rule numbers	2×rule numbers
defuzzy		1	rule numbers

III. Methodology

Adding interpretability features in deep learning structures could benefit certain applications of deep learning, where interpretability can be of benefit. For example, in advanced manufacturing systems, where understanding and modelling images and videos of complex processes are critical tasks. A process model (or classifier) based on CNNs could be developed to take advantage of processing data in array forms [3] which has already been proven to be very effective [6], [23] in a number of applications. Adding interpretability in a CNN deep learning structure could be achieved by performing the final classification task using a FL-based structure. In this section, we describe the integration of a Radial-Basis-Function Neural-Fuzzy layer into the deep learning structure, that provides the mechanism to extract a linguistic rule base from the CNN.

A. *Convolutional neural network with an RBF fuzzy logic rule-base classification layer*

In this section, the main CNN structure is summarised, and it is shown how the RBF-NF layer is integrated into the overall network structure and learning methodology.

Fuzzy Logic RBF CNN: In [3], LeCun states the usage of convolution layers of CNNs is to extract different scale features. In this research work, it is proposed that a deep learning network, which includes a convolution layered structure, and for the first time in the literature include a FL layer (RBF) to perform the classification task. An extra layer was proposed here, which is an RBF layer to maintain the rulebase of the system. To defuzzify the FL statements into crisp classification labels, a normalised exponential function (softmax) is used. Due to the addition of the FL layer one has to consider the credit assignment and error backpropagation for these layers which is not a trivial task.

Fig. 4 depicts the architecture of the FL RBF-CNN, and Table II shows parameter setting of the FL RBF-CNN.

Similar to FL RBF networks, FL RBF-CNNs will also be sensitive to initial conditions (initial model structure and parameters) of the RBF and defuzzification layers. Therefore, one has to establish some initial conditions for the FL rulebase for successful model training. The overall training would rely on a cross entropy loss function and it would be performed as follows:

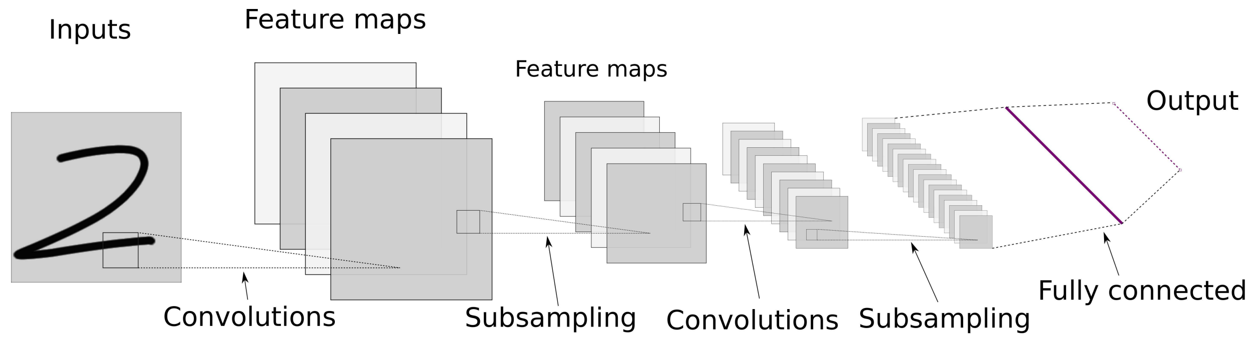


Fig. 2: Representative CNN structure

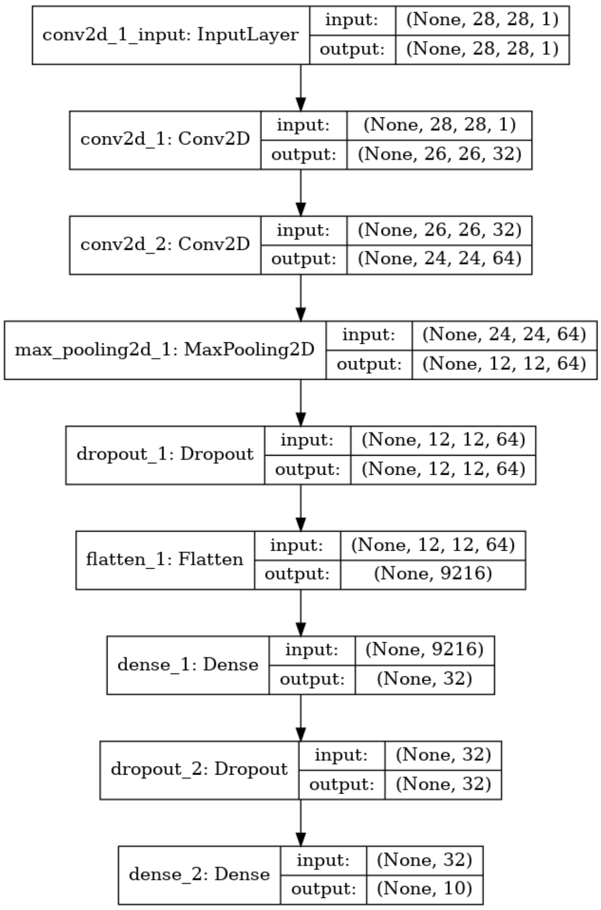


Fig. 3: basic CNN structure

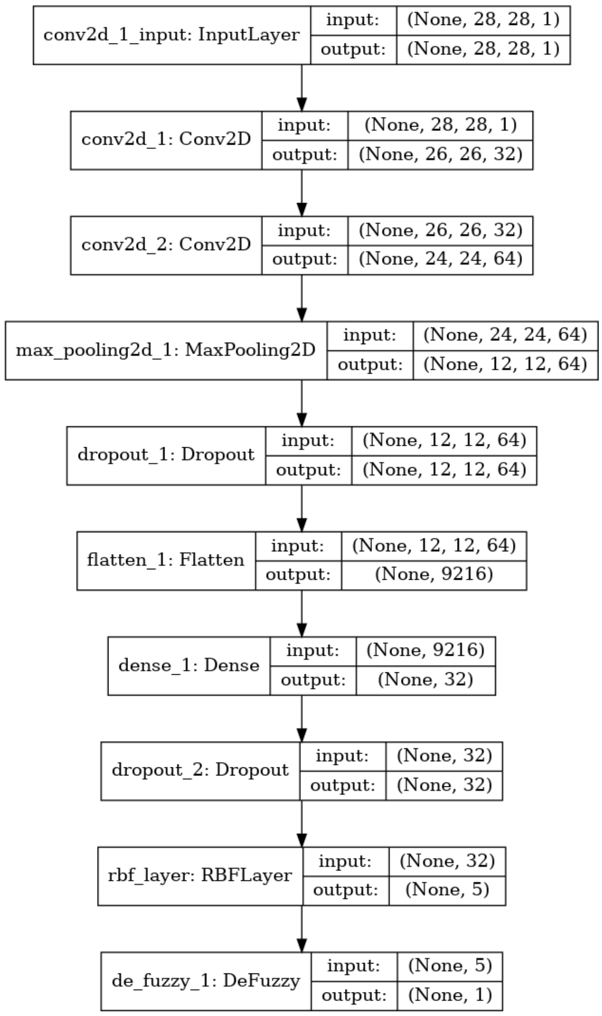


Fig. 4: FL RBF-CNN structure

a) RBF layer: Following from (4) and (7), the activation function becomes:

$$m_j^l = \exp\left(-\left\|\vec{x}^{l-1} - \vec{c}_i\right\|^2 / \vec{\sigma}_i^2\right), \tag{8}$$

$$g_j^l = m_j^j / \sum_{j=1}^{p} m_j^l, \tag{9}$$

therefore,

$$g_j^l = s\left(-\left\|\vec{x}^{l-1} - \vec{c}_i\right\|^2\right) / \vec{\sigma}_i^2, \tag{10}$$

where $s(x)$ is a softmax function.

b) Defuzzyfication layer: In defuzzy layer, using $g_k^l = g_j^{l-1}$, there would be

$$y^l = z^l \cdot g^l. \tag{11}$$

Noteworthily, the outputs of a RBF layer would be continuous floating numbers rather than discrete integers. Rounding the

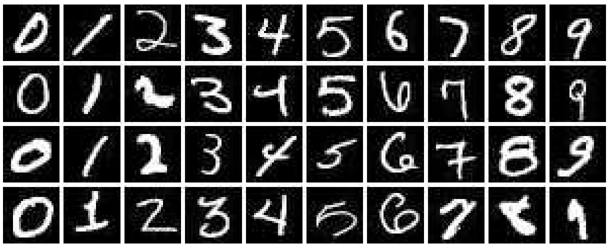


Fig. 5: Several examples from MNIST dataset

output of this layer to the nearest integer (based on a predetermined threshold) would give the integer class.

IV. Simulation Results

Simulation results were created to assess the performance of the developed deep learning structure. This is done in two parts, first the learning performance on a popular benchmark data set is assessed. This is achieved by comparing the proposed learning structure against a classical and state-of-the art CNN structure. On the second part, the robustness of the learning ability of the proposed system is assessed by reducing consecutively the sample size and evaluating the learning and recall performance.

The modified National Institute of Standards and Technology (MNIST) database was chosen as a case study; the MNIST database is a labelled handwriting digits dataset containing 60000 training images and 10000 testing images. The MNIST data set has a 60000-sample of training images and a 10000-sample of testing images as showing in Fig. 5. The training images were separated into two parts randomly in each model training process, which were a 50000-sample training set and a 10000-sample validation set.

A. MNIST training and testing simulation results: baseline CNN

The presented results include the mean classification accuracy as well as the standard deviation in each case. Each set of simulation results shows the loss function during training and validation as well as the classification accuracy for training and validation. This is presented for a number of different rules, for the rulebase of the Fuzzy-Logic-based classification layer (varying from 3 — simpler — to 15 rules — more complex). The learning model makes use of an adaptive learning rate method to optimise the model weights. The model is trained for 50 epochs, but also includes an early stopping criterion, to stop earlier if the validation performance is not improved, with an improvement window of 10 epochs. As shown in Fig. 6, the training of this network with 64 features converges within the first 30 epochs. The mean training accuracy (for 10 repeats) of this model was 99.80%, and both the validation and test accuracy of this model are at around 99% which is comparable with other state-of-the-art CNN classification structures. As an example comparison, LeNet-5 [8], which has a similar structure, achieves an accuracy of 99%. A higher test classification accuracy (99.77%) is achieved in [24], however

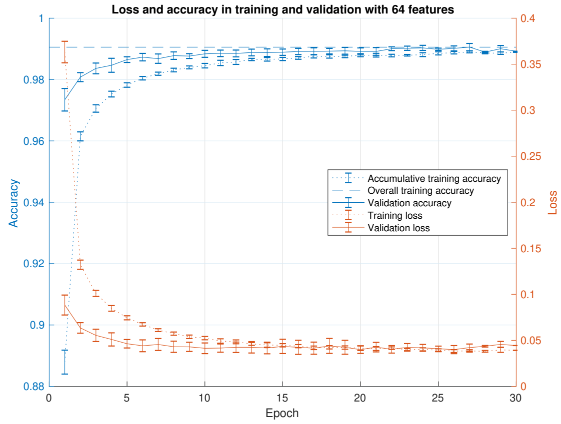


Fig. 6: Origin CNN model with 64 features result using 10 MNIST data sets

this is achieved with a significantly more complex structure. One can therefore conclude that the proposed structure does not sacrifice significant performance in this case study, despite the much simpler overall structure that aims at enhancing the interpretability of the model rather than its accuracy.

B. Fuzzy logic RBF model results: with variable rules

While in some cases, the interpretability of models would be the key part to understand the processing. For example in real industrial/manufacturing processes, the conditions causing faults and defects are eager for understanding.

The performance of this FL RBF-CNN is further assessed via reducing the number of classification features from 64, to 32 and to 16. The same algorithmic approach was followed, as presented in the previous section. Tables III, IV, and V were generated with using the raw simulation results (10 repeats per training case). In each of these three tables, there are two columns whose values are average accuracy and standard variance for training, validation, and test case respectively, and every feature case were trained from 3 to 15 rules as listed in with a reference CNN network result (labelled as REF). As shown in Table III, the mean accuracy has a trend that would reach the best performance when fuzzy rules equals to 5 or 7, and the standard deviation also shows a similar trend. However, to a certain extent, despite of the good performance, a model having 64 features may not be very interpretable, hence models with 32 and 16 features were also simulated to 'stress-test' the performance of the proposed structure. When the size of the classification features decreases, the neurons of the last fully connected layers also gets reduced. It is expected to observe a reduced classification power due to the fewer model parameters available to capture the classification problem. In general, the classification accuracy is reduced, as demonstrated in Table IV and Table V. Similarly, as in the case with 64 features, the best performance is observed between 5 and 7 rules . In the case of 32 features, the test accuracy of 93.11% could be considered as acceptable, however the test

TABLE III: Accuracy mean and standard deviation of the FL RBF-CNN model using 64 features

Rule	Training		Validation		Test	
3	96.64%	1.63%	94.79%	1.71%	94.67%	1.59%
5	98.41%	0.97%	96.79%	1.03%	96.69%	1.00%
7	98.48%	1.47%	96.89%	1.46%	96.92%	1.52%
9	97.78%	3.20%	96.16%	3.05%	96.28%	3.05%
11	94.14%	8.76%	92.55%	8.63%	92.63%	8.52%
13	95.48%	4.39%	94.03%	4.30%	93.97%	4.40%
15	94.90%	5.00%	93.43%	4.96%	93.44%	4.84%
REF	99.75%	0.04%	98.97%	0.13%	99.06%	0.07%

TABLE IV: Accuracy means and standard variances of the FL RBF-CNN model using 32 features

Rule	Training		Validation		Test	
3	87.17%	10.77%	85.54%	10.67%	85.76%	10.63%
5	94.50%	4.08%	92.91%	3.93%	93.11%	3.90%
7	92.19%	5.05%	90.81%	4.81%	91.02%	4.86%
9	91.48%	7.43%	90.12%	7.32%	90.33%	7.25%
11	90.90%	5.21%	89.58%	5.19%	89.67%	5.02%
13	86.38%	7.19%	85.00%	7.15%	85.32%	7.16%
15	73.84%	33.13%	72.72%	32.59%	73.01%	32.61%
REF	99.59%	0.07%	98.84%	0.12%	98.98%	0.08%

accuracy of 67.90% in the case with 16 features demonstrates that there is a significant performance loss when the number of features is very low.

C. Model Interpretability

With the fully connected layer of the CNN structure being a Fuzzy Logic based layer, one can enhance the interpretability of the classification task, by extracting Fuzzy Logic rules directly from the classification layer. such information can be, for example, further used to aid decision making, or to create human-machine interfaces. Fig. 7, as an example, depicts two different rules from the rulebase of the 32-feature FL RBF-CNN model; just four inputs (features) and one output (classification weight) are shown for simplicity. Rule 1 for example, translates into the following Singleton-based Fuzzy rule:

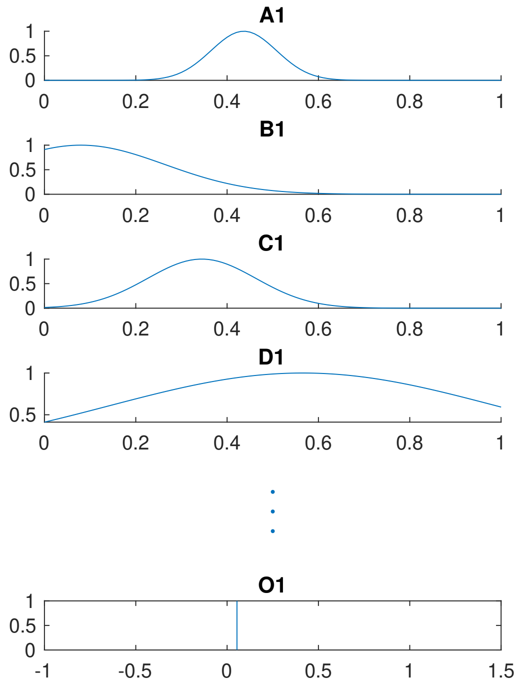
'IF Feature 1 is A1, and Feature 2 is B1, and Feature 3 is C1, and.. etc. THEN the Output class is O1.'

V. Conclusion

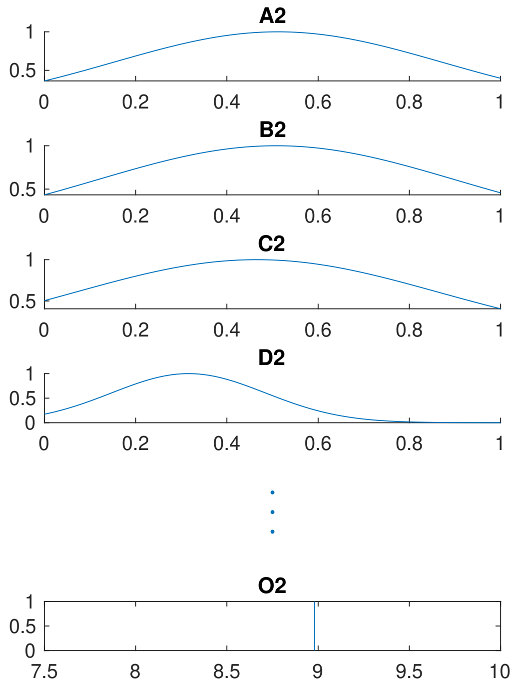
In this research work, an interpretability-oriented deep learning network is presented, based on a CNN structure

TABLE V: Accuracy means and standard variances of the FL RBF-CNN model using 16 features

Rule	Training		Validation		Test	
3	66.61%	8.60%	65.35%	8.35%	65.56%	8.52%
5	62.93%	12.82%	62.03%	12.64%	62.32%	12.61%
7	68.74%	9.12%	67.53%	8.72%	67.90%	8.76%
9	65.48%	9.60%	64.61%	9.49%	65.04%	9.19%
11	60.60%	5.91%	59.88%	5.90%	59.88%	5.70%
13	59.98%	18.73%	59.06%	18.41%	59.51%	18.50%
15	56.49%	24.31%	55.86%	23.96%	56.08%	24.09%
REF	99.27%	0.11%	98.56%	0.17%	98.66%	0.12%



(a) Rule 1



(b) Rule 2

Fig. 7: Two of five rule bases of a FL RBF-CNN model with 32 features

combined with a Fuzzy Logic structure to perform the classification task and also provide the capability to linguistically interpret the structure's rulebase. By combining the good feature extraction property of CNNs and the classification and generalisation ability of FL based systems, a FL RBF-CNNs was developed. The proposed structure relies on a Radial Basis Function realisation of the Neural-Fuzzy network, which is integrated into the CNN structure via an adaptive subgradient method for the credit assignment and error backpropagation.

In simulation results (training, validation and testing/recall) using a popular dataset often used for benchmarking (MNIST 70000 handwriting digit samples) it is shown that the proposed network performs equally well when compared to state-of-the-art CNN-based networks of similar complexity and size. However, the advantage of the proposed structure, is that due to the added classification layer in the form of a FL rulebase, one could extract linguistic FL statements for the overall model, which would enhance the interpretability of the system. For example, in decision making applications, one could extract autonomously linguistic rules to assist a human user/operator. To further extend this research work, it would be interesting to capture and visualise the connection between features and predictions via FL RBF-CNN layers.

REFERENCES

[1] R. Schaprie, "Computer science 511 theoretical machine learning," February 2008. [Online]. Available: https://www.cs.princeton.edu/courses/archive/spr08/cos511/scribe_notes/0204.pdf

[2] M. I. Jordan and T. M. Mitchell, "Machine learning: Trends, perspectives, and prospects," *Science*, vol. 349, no. 6245, pp. 255–260, 2015.

[3] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[4] J. Gong, M. D. Goldman, and J. Lach, "Deepmotion: A deep convolutional neural network on inertial body sensors for gait assessment in multiple sclerosis," in *2016 IEEE Wireless Health, WH 2016*, 2016, Conference Proceedings, pp. 164–171.

[5] T. Segreto, A. Caggiano, S. Karam, and R. Teti, "Vibration sensor monitoring of nickel-titanium alloy turning for machinability evaluation," *Sensors (Switzerland)*, vol. 17, no. 12, 2017.

[6] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, Conference Proceedings, pp. 1–9.

[7] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[8] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the Ieee*, vol. 86, no. 11, pp. 2278–2324, 1998.

[9] J. Deng, W. Dong, R. Socher, L. J. Li, L. Kai, and F.-F. Li, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, Conference Proceedings, pp. 248–255.

[10] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *CoRR*, vol. abs/1512.03385, 2015.

[11] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014. [Online]. Available: http://arxiv.org/abs/1409.1556

[12] A. Muniategui, B. Hériz, L. Eciolaza, M. Ayuso, A. Iturrioz, I. Quintana, and P. Álvarez, "Spot welding monitoring system based on fuzzy classification and deep learning," in *2017 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, 2017, Conference Proceedings, pp. 1–6.

[13] Y. Deng, Z. Ren, Y. Kong, F. Bao, and Q. Dai, "A hierarchical fused fuzzy deep neural network for data classification," *IEEE Transactions on Fuzzy Systems*, vol. 25, no. 4, pp. 1006–1012, 2017.

[14] D. S. Broomhead and D. Lowe, "Radial basis functions, multi-variable functional interpolation and adaptive networks," Royal Signals and Radar Establishment Malvern (United Kingdom), Report, 1988.

[15] K. R. Muller, S. Mika, G. Ratsch, K. Tsuda, and B. Scholkopf, "An introduction to kernel-based learning algorithms," *IEEE Transactions on Neural Networks*, vol. 12, no. 2, pp. 181–201, 2001.

[16] G. Panoutsos and M. Mahfouf, "A neural-fuzzy modelling framework based on granular computing: Concepts and applications," *Fuzzy Sets and Systems*, vol. 161, no. 21, pp. 2808–2830, 2010.

[17] M. Y. Chen and D. A. Linkens, "A systematic neuro-fuzzy modeling framework with application to material property prediction," *IEEE Trans Syst Man Cybern B Cybern*, vol. 31, no. 5, pp. 781–90, 2001. [Online]. Available: https://www.ncbi.nlm.nih.gov/pubmed/18244842

[18] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, 2011, Conference Proceedings, pp. 315–323.

[19] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *CoRR*, vol. abs/1502.03167, 2015. [Online]. Available: http://arxiv.org/abs/1502.03167

[20] T. Schaul, I. Antonoglou, and D. Silver, "Unit tests for stochastic optimization," *CoRR*, vol. abs/1312.6055, 2013. [Online]. Available: http://arxiv.org/abs/1312.6055

[21] Y. Bengio, N. Boulanger-Lewandowski, and R. Pascanu, "Advances in optimizing recurrent networks," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 8624–8628.

[22] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *Journal of Machine Learning Research*, vol. 12, no. Jul, pp. 2121–2159, 2011.

[23] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[24] D. C. Ciresan, U. Meier, and J. Schmidhuber, "Multi-column deep neural networks for image classification," *CoRR*, vol. abs/1202.2745, 2012. [Online]. Available: http://arxiv.org/abs/1202.2745