Fuzzy-Rough Cognitive Networks: Theoretical Analysis and Simpler Models

Concepcion, Leonardo; Napoles, Gonzalo; Grau, Isel; Pedrycz, Witold

Published in:
IEEE Transactions on Cybernetics

DOI:
10.1109/TCYB.2020.3022527

Publication date:
2022

Document Version:
Accepted author manuscript

Fuzzy-Rough Cognitive Networks: Theoretical Analysis and Simpler Models

Leonardo Concepción, Gonzalo Nápoles, Isel Grau, Witold Pedrycz

***Abstract*—Fuzzy-Rough Cognitive Networks (FRCNs) are recurrent neural networks intended for structured classification purposes in which the problem is described by an explicit set of features. The advantage of this granular neural system relies on its transparency and simplicity while being competitive to state-of-the-art classifiers. Despite of their relative empirical success in terms of prediction rates, there are limited studies on FRCNs' dynamic properties and how their building blocks contribute to algorithm's performance. In this paper, we theoretically study these issues and conclude that boundary and negative neurons always converge to a *unique* fixed-point attractor. Moreover, we demonstrate that negative neurons have no impact on algorithm's performance and that the ranking of positive neurons is invariant. Moved by our theoretical findings, we propose two simpler fuzzy-rough classifiers that overcome the detected issues and maintain the competitive prediction rates of this classifier. Toward the end, we present a case study concerned with image classification in which a Convolutional Neural Network is coupled with one of the simpler models derived from the theoretical analysis of the FRCN model. The numerical simulations suggest that, once the features have been extracted, our granular neural system performs as well as other recurrent neural networks.**

***Index Terms*—rough cognitive mapping, fuzzy-rough cognitive networks, convergence, granular computing.**

I. Introduction

Granular Neural Networks (GNNs) were introduced in [1] as the amalgamation between Neural Networks and Granular Computing. Roughly speaking, this synergy aims at reconciling the black-box behavior and the lack of transparency of neural networks with the aid of information granules, such as classes, clusters, subsets, etc [2]. Through similarity among objects and granulation [3], GNNs bypass the use of large datasets and precise information, while building interpretable and lighter models. Several GNN architectures have been proposed over the years, varying the methods of data granulation, the network architecture, the learning procedure to adjust the model, among other modifications [4] [5] [6].

Rough Cognitive Networks (RCN) are a type of GNN presented in by Nápoles et al. [7] to solve decision-making and pattern classification problems. In this model, the information space is granulated by using the Rough Set Theory [8] [9] and then a recurrent neural network is built. Interestingly, no synaptic learning is needed since weights are prescriptively derived. According to simulations, this network is capable of outperforming standard classifiers while remaining akin to rough recognition techniques [7] [10]. However, the granulation process has its Achilles heel at learning the similarity threshold for comparing objects, because this procedure demands significant computational cost.

Rough Cognitive Ensembles (RCEs) attempted to overcome the burden of tuning the similarity threshold parameter [11]. RCEs are granular multiclassifiers composed of several RCNs, each operating at a different level of granularity. In order to promote the diversity among the base classifiers, this granular ensemble uses instance bagging. During the exploitation process, the base classifiers produce output vectors which are aggregated by means of a voting procedure over all decision classes. The RCE model outperformed most of the state-of-the-art classifiers [11] on 140 datasets. However, the ensemble architecture harms the system transparency although it does suppress the similarity threshold parameter.

Fuzzy-Rough Cognitive Networks (FRCNs) [12] completely suppressed the requirement of a similarity threshold parameter while outperforming the RCEs' performance. The main characteristic of FRCNs is that they replace the crisp information granules with fuzzy-rough ones [13] [14] [6]. These fuzzy-rough information granules are deemed pivotal when activating the input neurons for a given instance. When contrasted with other classifiers, it results that FRCNs' performance is equivalent to the most successful black boxes, while outperforming other instance-based learners [12].

Despite the FRCNs' promising performance, we have little knowledge on the network's dynamic properties. For example, RCN-based models use a stopping criterion based on the number of iterations, hence lightening algorithm's computational burden. However, abruptly stopping the inference mechanism might cause the model to report inconsistent results. On the other hand, the network should not converge to a unique fixed-point attractor, otherwise the model will produce the same decision class regardless of the input vector.

Although the above issues are indeed interesting, what have motivated this paper are the empirical simulations conducted by Vanloffelt et all. [15]. After testing several architectures, they concluded that the connections among the positive regions might not be necessary to maintain FRCNs' performance. No conclusion was drawn for boundary and negative neurons. Based on these results, it seems evident that not all building-blocks contribute equally when determining the decision class for a given instance. Which granular neurons contribute the most and why remain open questions.

G. Nápoles (g.r.napoles@uvt.nl) is with the Department of Cognitive Science & Artificial Intelligence, Tilburg University, The Netherlands.

G. Nápoles and L. Concepción are with the Faculty of Business Economics, Universiteit Hasselt, Belgium.

W. Pedrycz is with the Department of Electrical and Computer Engineering, University of Alberta, Canada.

I. Grau is with the Artificial Intelligence Lab, Vrije Universiteit Brussel, Belgium.

This paper analytically explores the above issues and brings up three main contributions. Firstly, we prove that negative and boundary neurons converge to a unique fixed point when the number of iterations is large enough. Secondly, we study the influence of each kind of neuron on FRCNs' performance and conclude that negative ones have no influence at all, so the classification relies on both positive and boundary neurons. Besides, we derive some interesting properties such as the invariance of the ranking among all positive neurons, and the dominance relation between two decision classes based on the incoming boundary connections or the difference among the related positive neurons. Based on these theoretical findings, we propose two simpler FRCN-based classifiers that surpass the detected issues while retaining algorithm's performance. Finally, we present a case study concerned with image classification that shows promising results.

The remainder of this paper is organized as follows. Section II presents the theoretical background surrounding FRCNs, while Section III investigates how every component in the network contributes to the classification process. Section IV brings up two models to overcome the investigated limitations, while Section V compares these models against state-of-the-art classifiers. Finally, conclusions are presented.

II. Fuzzy-Rough Cognitive Networks

In this section, we describe how to build an FRCN for pattern classification. Let $\mathcal{U}$ denote the universe with all objects in the training dataset and $X_c \subset \mathcal{U}$ as the subset containing all instances labeled with decision class Y_c. Equation (1) shows the membership degree of $x \in \mathcal{U}$ to X_c, which is computed in a binary way for the sake of simplicity,

$$\mu_{X_c}(x) = \begin{cases} 1 & , x \in X_c \\ 0 & , x \notin X_c \end{cases}. \tag{1}$$

The membership function $\mu_P(y, x)$ in Equation (2) uses the similarity between two instances x and y,

$$\mu_P(y, x) = \mu_{X_c}(x)\varphi(x, y) = \mu_{X_c}(x)(1 - \delta(x, y)). \tag{2}$$

where $\mu_P : \mathcal{U} \times \mathcal{U} \to [0, 1]$ is the membership degree of y to X_c given that x belongs to the fuzzy set X_c. To achieve this, we combine the previously described membership degree $\mu_{X_c}(x)$ with the similarity degree $\varphi(x, y)$. Such a similarity degree is expressed in terms of a normalized distance function $\delta(x, y)$ for heterogeneous instances.

Equations (3) and (4) denote the membership functions for the lower and upper approximations, respectively, associated with any fuzzy set X_c as proposed in [16]:

$$\mu_{P_*(X_c)}(x) = \min\left\{\mu_{X_c}(x), \inf_{y \in \mathcal{U}} \mathcal{I}(\mu_P(y, x), \mu_{X_c}(y))\right\}, \tag{3}$$

$$\mu_{P^*(X_c)}(x) = \max\left\{\mu_{X_c}(x), \sup_{y \in \mathcal{U}} \mathcal{T}(\mu_P(x, y), \mu_{X_c}(y))\right\}. \tag{4}$$

where $\mathcal{I}$ represents an implication function for the lower approximations, such that $\mathcal{I}(0,0) = \mathcal{I}(0,1) = \mathcal{I}(1,1) = 1$ and $\mathcal{I}(1,0) = 0$. Similarly, for the upper approximations we use a conjunction function $\mathcal{T}$ such that $\mathcal{T}(0,0) = \mathcal{T}(0,1) = \mathcal{T}(1,0) = 0$ and $\mathcal{T}(1,1) = 1$.

Later on, we can compute the membership functions associated with the positive, negative and boundary regions as $\mu_{POS(X_c)}(x) = \mu_{P_*(X_c)}(x)$, $\mu_{NEG(X_c)}(x) = 1 - \mu_{P^*(X_c)}(x)$ and $\mu_{BND(X_c)}(x) = \mu_{P^*(X_c)}(x) - \mu_{P_*(X_c)}(x)$, respectively. Such fuzzy information granules enclose the main building-blocks of the FRCN classifier.

After computing the membership functions associated with each decision class, we can build a causal network involving four types of neurons. Let $\mathcal{D} = \{D_1, D_2, \dots D_M\}$ denote the set of decision neurons, while $\mathcal{P} = \{P_1, P_2, \dots P_M\}$, $\mathcal{N} = \{N_1, N_2, \dots N_M\}$ and $\mathcal{B} = \{B_1, B_2, \dots B_M\}$ are the sets of positive, negative and boundary neurons, respectively. This recurrent neural network contains $|\mathcal{D}|$ output neurons, between $2|\mathcal{D}|$ and $3|\mathcal{D}|$ input neurons and between $2|\mathcal{D}|(1+|\mathcal{D}|)$ and $3|\mathcal{D}|(1+|\mathcal{D}|)$ weights, depending on the number of non-empty boundary regions. It is worth mentioning that weights are non-trainable as they are prescriptively determined based on the semantics of information granules.

Let $\mathcal{C} = \{C_1, C_2, \dots C_N\}$ be the set of neurons, which can be obtained as the union of the disjoint sets $\mathcal{D}$,$\mathcal{P}$, $\mathcal{N}$ and $\mathcal{B}$. Algorithm 1 shows the construction steps when building an FRCN classifier. Firstly, fuzzy-rough regions are mapped onto input neurons, while output neurons denote decision classes. Positive, negative and boundary neurons influence themselves with an intensity of 1.0. This prevents the initial activation values to vanish when performing the reasoning process. Secondly, there is a positive causal relation between each positive neuron and the decision neuron related to it. Such neurons negatively affect the remaining positive and decision neurons. Thirdly, negative neurons only affect their corresponding decisions in a negative way but not the opposed ones. This happens because rejecting a decision does not endorse the acceptance of any specific decision, unless the problem is binary. Finally, if two decision classes share non-empty boundary regions, then each boundary neuron influences both decision neurons with 0.5 intensity.

Figure 1 shows the FRCN model for a classification problem with only two decision classes.

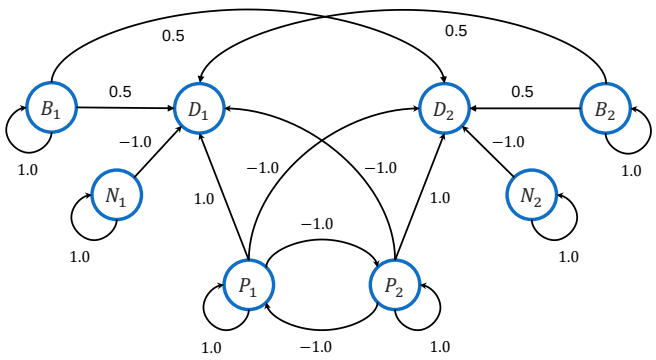


Fig. 1: FRCN model for binary classification.

Algorithm 1 Network construction procedure

for each subset X_c **do**
  Add a neuron P_c as the c-th positive region
  Add a neuron N_c as the c-th negative region
  Add a neuron B_c as the c-th boundary region
  Add a neuron D_c as the c-th decision class
end for
for each neuron C_i **do**
  if $C_i \neq D_c$ **then**
    $w_{ii} = 1.0$
  end if
  for each neuron C_j **do**
    if $C_i = P_c$ **then**
      if $C_j = D_c$ **then**
        $w_{ij} = 1.0$
      else if $C_j = D_{v \neq c}$ **then**
        $w_{ij} = -1.0$
      else if $C_j = P_{v \neq c}$ **then**
        $w_{ij} = -1.0$
      end if
    end if
    if $C_i = N_c$ **and** $C_j = D_c$ **then**
      $w_{ij} = -1.0$
    end if
    if $C_i = B_c$ **and** $C_j = D_v$ **and** $\min_{x \in \mathcal{U}} \{\mu_{BND(X_c)}(x), \mu_{BND(X_v)}(x)\} > 0$ **then**
      $w_{ij} = 0.5$
    end if
  end for
end for

The initial activation value $A_i^{(0)}$ of the neuron C_i is computed based on the similarity degree between the new object y and all $x \in \mathcal{U}$, and the membership degree of every x to each fuzzy-rough granular region. Moreover, for decision neurons the initial activation value is 0. Equations (5), (6) and (7) formalize the initial activation rules for positive, negative and boundary neurons, respectively,

$$A_i^{(0)} = \frac{\sum_{x \in \mathcal{U}} \mathcal{T}(\varphi(x,y), \mu_{POS(X_c)}(x))}{\sum_{x \in \mathcal{U}} \mu_{POS(X_c)}(x)},\ C_i = P_c \quad (5)$$

$$A_i^{(0)} = \frac{\sum_{x \in \mathcal{U}} \mathcal{T}(\varphi(x,y), \mu_{NEG(X_c)}(x))}{\sum_{x \in \mathcal{U}} \mu_{NEG(X_c)}(x)},\ C_i = N_c \quad (6)$$

$$A_i^{(0)} = \frac{\sum_{x \in \mathcal{U}} \mathcal{T}(\varphi(x,y), \mu_{BND(X_c)}(x))}{\sum_{x \in \mathcal{U}} \mu_{BND(X_c)}(x)},\ C_i = B_c. \quad (7)$$

Equation (8) displays how to update neurons' activation values at each iteration as proposed in [17],

$$A_i^{(t+1)} = f(\sum_{j=1}^{N} w_{ji} A_j^{(t)}) \quad (8)$$

where w_{ji} is the causal weight representing the influence of C_j on C_i. The sigmoid transfer function $f(x) = \frac{1}{1+e^{-\lambda x}}$ with $\lambda = 5$ is used to keep the neurons' activation values within the $[0, 1]$ interval. It is worth mentioning that high λ values cause the sigmoid function to become "more binary", which will translate into several decision neurons having near maximal activation values. If this situation comes to light, then it would be difficult to determine which decision class should be produced based on neurons' activation values. Conversely, small λ values are prone to produce unique fixed-point attractors, as explained in Section III-A. Therefore, $\lambda = 5$ seems to be a good commitment to avoid both issues.

Algorithm 2 depicts the FRCNs' reasoning process when determining the proper decision class for a given instance, after having activated the input neurons using Equations (5), (6) and (7). Notice that this update process is performed until either the network converges to a fixed-point attractor or a predefined number of iterations is reached. We say that a fixed-point attractor has been found when $A_i^{(t+1)} \approx A_i^{(t)}$ for each neuron in the causal network.

Algorithm 2 Classification process

for $t = 0$ to T **do**
  converged $\leftarrow$ True
  for each neuron C_i **do**
    Compute $A_i^{(t+1)}$ according to (8)
    if $A_i^{(t)} \neq A_i^{(t+1)}$ **then**
      converged $\leftarrow$ False
    end if
  end for
  if converged **then**
    return $\text{argmax}_c\{A_x^{(t+1)}(D_c)\}$
  end if
end for
if not converged **then**
  return $\text{argmax}_c\{A_x^{(T)}(D_c)\}$
end if

In the following section, we theoretically analyze the dynamic behavior of FRCN models.

III. Theoretical analysis

When performing the classification, FRCNs should be stable but not convergent to a unique fixed-point attractor. While the stability provides consistent interpretation of results, unique fixed-point attractors cause the network to produce the same decision class despite the initial stimulus.

The following subsections are devoted to analyzing the conditions causing some FRCN neurons to converge to unique fixed-point attractors. Furthermore, we are interested in studying the impact of each neural processing entity on algorithm's discriminatory capability.

A. Analyzing fixed-point attractors

In this section, we use insights reported in [18] to analyze which λ values are not desired in the transfer function, i.e. they lead to a unique fixed-point attractor. These findings disprove the results in [19], thus leading to new theorems. They also proved that a Fuzzy Cognitive Map (FCM) [17] whose weight

matrix is comprised of non-negative values has a fixed-point for any λ value. This result does not mean that there is a unique fixed-point attractor or that there is always an attractor for every initial stimulus. The actual implication is that there is at least an activation vector which is mapped to itself after every FCM iteration, thus leading to a fixed point.

According to Theorem 3 in [18], if the Frobenius norm of the weight matrix of an FCM is smaller than $4/\lambda$ (with $\lambda > 0$), then the FCM has one and only one fixed point. Therefore, for any FRCN we have that $\|W\|_F = \frac{1}{2}\sqrt{9M^2 + 12M}$, with W being the weight matrix and M being the number of decision neurons. Now, if the following inequality holds, the FRCN has one and only one fixed-point attractor:

$$\frac{1}{2}\sqrt{9M^2 + 12M} < \frac{4}{\lambda} \quad \equiv \quad \lambda < \frac{8}{\sqrt{9M^2 + 12M}}. \tag{9}$$

For example, when having two decision neurons we get that for $\lambda < \frac{4}{\sqrt{15}} \approx 1.03$ the FRCN converges to a fixed-point attractor. More generally, the bound of λ decreases with the number of decision neurons. FRCNs use $\lambda = 5$ so that decision neurons produce values with higher discriminatory power, thus the result in [18] cannot ensure the existence and uniqueness of the fixed-point attractor in such networks.

Neurons with self-feedback: FRCNs have boundary and negative neurons which are influenced only by themselves, so each of them has a self-feedback connection. Aiming at analyzing the convergence of these neurons some definitions and theorems [20] need to be unveiled.

Definition 1. *A fixed point of mapping $f : X \to X$ is a point $x^* \in X$ such that $f(x^*) = x^*$.*

Theorem 1. *Let f be continuous in $\big[a, b\big]$*

(i) *If $f\big(x\big) \in \big[a, b\big]$ $\forall x \in \big[a, b\big]$, then f has at least one fixed point in $\big[a, b\big]$.*

(ii) *If, in addition, $f^{'}\big(x\big)$ exists on $\big(a, b\big)$ and a positive constant $k < 1$ exists with*

$$|f^{'}\big(x\big)| \leq k, \ \forall x \in \big(a, b\big),$$

then there is exactly one fixed point in $\big[a, b\big]$.

Theorem 2. *Let f be continuous in $\big[a, b\big]$ such that $f\big(x\big) \in \big[a, b\big]$ $\forall x \in \big[a, b\big]$. Suppose in addition that $f^{'}$ exists on $\big(a, b\big)$ and a positive constant $k < 1$ exists with*

$$|f^{'}\big(x\big)| \leq k, \ \forall x \in \big(a, b\big),$$

then for any number $p_0 \in \big[a, b\big]$ the sequence defined by

$$p_n = f\big(p_{n-1}\big), n \geq 1,$$

converges to the unique fixed point p in $\big[a, b\big]$.

Theorem 3 gives insights into the dynamic behavior of self-connected neurons that do not have any incoming connection. This result is one of the contributions of our research and can be generalized to other scenarios.

Theorem 3. *In an FCM, a sigmoid neuron with self-feedback and no other incoming connection will always converge to a unique fixed-point regardless its initial stimulus.*

Corollary 1.1. *In an FRCN, negative and boundary neurons converge to a unique fixed point.*

Proof. The neurons' update rule transforms the raw activation value at t-th iteration ($x^{(t)}$) into a bounded interval by using the sigmoid transfer function such that $x^{(t+1)} = \frac{1}{1+e^{-\lambda(x^{(t)})}}$. Furthermore, *Theorem 1* ensures that this function has at least one fixed point in the $\big[0, 1\big]$ interval, because it maps the $\big[0, 1\big]$ space into itself (even though bounds of the interval are not reached). Moreover, this function is also continuous in all its domain. But we could make the proof simpler by taking into account that $f(x) = \frac{1}{1+e^{-\lambda x}}$ produces values into $\big[0.5, 1\big)$, since the raw activation values of such neurons always lie within the $\big[0, 1\big]$ interval. This means that the initial stimulus becomes irrelevant when analyzing the existence, uniqueness and attractiveness of fixed points. The following definition and lemmas depict the kind of points for which we need to prove the attraction to a fixed point.

Definition 2. *A secondary point of mapping $f : X \to X$ is a point $x^* = f(x)$, being x an initial stimulus belonging to the function's domain.*

Lemma 4. *If mapping $f : X \to X$ has a fixed-point, then it is a secondary point.*

Lemma 5. *If for any secondary point $p_1 \in \big[a, b\big]$, the sequence defined by $p_n = f\big(p_{n-1}\big), n \geq 2$, converges to the unique fixed point p in $\big[a, b\big]$, then for every initial stimulus p_0 such that $p_1 = f\big(p_0\big)$, the sequence also converges to p.*

Now, we analyze the premises of *Theorem 2* for $f : [0.5, 1] \to [0.5, 1]$ since secondary points for $f : [0, 1] \to [0, 1]$ belong to the $[0.5, 1]$ interval. Likewise, we rely on *Lemmas 4* and 5 to complete the proof. To investigate if the fixed point is unique and also an attractor from every secondary point, the following inequality must hold:

$$|f^{'}(x)| < 1. \tag{10}$$

After doing some algebraic transformations and removing the modulus operator (the derivative is always no negative given that λ is positive) we obtain:

$$|f^{'}(x)| = |\Big(\frac{1}{1+e^{-\lambda x}}\Big)^{'}| = \frac{\lambda e^{-\lambda x}}{1+e^{-\lambda x}} = \lambda f(x)\big(1 - f(x)\big).$$

Therefore, we have to prove that:

$$\lambda f(x)\big(1 - f(x)\big) < 1. \tag{11}$$

To solve this inequality we study the cases when $\lambda < 4$ and $\lambda \geq 4$. Such cases are derived from the inequality we attempt to solve, so they will be unveiled opportunely.

Case 1 ($\lambda < 4$). Given the fact that $0 < f(x) < 1$ and using the Arithmetic Mean - Geometric Mean Inequality [21] we can arrive at the following inequality:

$$f(x)(1 - f(x)) \leq \Big(\frac{f(x) + (1 - f(x))}{2}\Big)^2 = \frac{1}{4}.$$

Now, by multiplying the whole expression by λ we have that $\lambda f(x)(1 - f(x)\big) \leq \frac{\lambda}{4}$, which implies that if $\lambda < 4$ then

$|f^{'}(x)| < 1$. Therefore, the fixed point is unique and it is an attractor for every secondary point.

Case 2 ($\lambda \geq 4$). After manipulating algebraically the inequality in (11) we obtain:

$$-\lambda f^2(x) + \lambda f(x) - 1 < 0$$

$$\lambda f^2(x) - \lambda f(x) + 1 > 0.$$

This inequality holds when $f(x) < \frac{1}{2} - \frac{\sqrt{1-\frac{4}{\lambda}}}{2}$ or $f(x) > \frac{1}{2} + \frac{\sqrt{1-\frac{4}{\lambda}}}{2}$. Nevertheless, as we know that $f(x)$ will never be smaller than 0.5, we can focus on proving that $f(x) > \frac{1}{2} + \frac{\sqrt{1-\frac{4}{\lambda}}}{2}$. Starting with

$$\frac{1}{1+e^{-\lambda x}} > \frac{1}{2} + \frac{\sqrt{1-\frac{4}{\lambda}}}{2}, \tag{12}$$

and after some algebraic transformations we get

$$\frac{1-\sqrt{1-\frac{4}{\lambda}}}{1+\sqrt{1-\frac{4}{\lambda}}} > e^{-\lambda x}. \tag{13}$$

When $\lambda \geq 4$ we obtain that $1 - \sqrt{1-\frac{4}{\lambda}} \geq 0$. Notice that the equality holds only if $\lambda = 4$. Therefore, with $\lambda = 4$ the inequality in (12) becomes:

$$\frac{1}{1+e^{-4x}} > \frac{1}{2}.$$

It does not hold only when $x \leq 0$ but we have $x \in [0.5, 1]$, which implies it suffices.

Assuming that $1 - \sqrt{1-\frac{4}{\lambda}} > 0$ we can apply the Neperian logarithm to both sides of inequality, thus yielding:

$$\ln\left(\frac{1-\sqrt{1-\frac{4}{\lambda}}}{1+\sqrt{1-\frac{4}{\lambda}}}\right) > -\lambda x$$

$$x > \left(-\frac{1}{\lambda}\right)\ln\left(\frac{1-\sqrt{1-\frac{4}{\lambda}}}{1+\sqrt{1-\frac{4}{\lambda}}}\right).$$

Applying logarithmic properties we have that:

$$x > \left(\frac{1}{\lambda}\right)\ln\left(\frac{1+\sqrt{1-\frac{4}{\lambda}}}{1-\sqrt{1-\frac{4}{\lambda}}}\right). \tag{14}$$

To lighten the algebraic work we encapsulate the right-hand side of Equation (14) into the following expression:

$$\mathbf{r}(\lambda) = \left(\frac{1}{\lambda}\right)\ln\left(\frac{1+\sqrt{1-\frac{4}{\lambda}}}{1-\sqrt{1-\frac{4}{\lambda}}}\right).$$

Since $x \geq \frac{1}{2}$, we just need to prove that $\mathbf{r}(\lambda) < \frac{1}{2}$ because it would be $x \geq \frac{1}{2} > \mathbf{r}(\lambda)$. This can be done as follows:

$$\left(\frac{1}{\lambda}\right)\ln\left(\frac{1+\sqrt{1-\frac{4}{\lambda}}}{1-\sqrt{1-\frac{4}{\lambda}}}\right) < \frac{1}{2}$$

then,

$$\ln\left(\frac{1+\sqrt{1-\frac{4}{\lambda}}}{1-\sqrt{1-\frac{4}{\lambda}}}\right) < \frac{\lambda}{2}$$

and finally we obtain

$$\frac{1+\sqrt{1-\frac{4}{\lambda}}}{1-\sqrt{1-\frac{4}{\lambda}}} < e^{\frac{\lambda}{2}}. \tag{15}$$

By using the Maclaurin series expansion [22] we have that $e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!}$. This series converges for every real value of x, but we have special interest in positive values of x, for which we also have that $e^x > \sum_{n=0}^{k} \frac{x^n}{n!}$ for any $k \geq 0$. Such inequality results immediately because the expansion is monotonically increasing in terms of k (with $x > 0$) and e^x is the limit of series when $k \to \infty$.

Since $\lambda > 0$, the inequality becomes $e^{\frac{\lambda}{2}} > \sum_{n=0}^{k} \frac{\left(\frac{\lambda}{2}\right)^n}{n!}$, but we only need the expansion for $k = 2$. This gives us that $e^{\frac{\lambda}{2}} > 1 + \frac{\lambda}{2} + \frac{\lambda^2}{8}$, which can be used to prove the inequality in (15). More specifically,

$$1 + \frac{\lambda}{2} + \frac{\lambda^2}{8} \geq \frac{1+\sqrt{1-\frac{4}{\lambda}}}{1-\sqrt{1-\frac{4}{\lambda}}} = 1 + \frac{2\sqrt{1-\frac{4}{\lambda}}}{1-\sqrt{1-\frac{4}{\lambda}}}.$$

Simplifying and multiplying by 8 we get:

$$4\lambda + \lambda^2 \geq \frac{16\sqrt{1-\frac{4}{\lambda}}}{1-\sqrt{1-\frac{4}{\lambda}}}.$$

After multiplying by $1 - \sqrt{1-\frac{4}{\lambda}}$ and performing some algebraic transformations we obtain:

$$4\lambda + \lambda^2 \geq \sqrt{1-\frac{4}{\lambda}}\left(16 + 4\lambda + \lambda^2\right).$$

Now, given the fact that $\lambda > 4$ we can square both sides while loosing no solutions as follows:

$$\lambda^4 + 8\lambda^3 + 16\lambda^2 \geq \frac{\left(\lambda^5 + 4\lambda^4 + 16\lambda^3 - 64\lambda^2 - 256\lambda - 1024\right)}{\lambda}$$

which is equivalent to:

$$4\lambda^4 + 64\lambda^2 + 256\lambda + 1024 \geq 0.$$

This inequality always holds because λ is positive and this fact implies that $\mathbf{r}(\lambda) < \frac{1}{2}$ is true, which proves that the inequality in (12) holds.

Such results ensure that $f(x) = \frac{1}{1+e^{-\lambda x}}, f : [0.5, 1] \to [0.5, 1]$ has a unique fixed-point (according to *Theorem 1*) and also that, for any secondary stimulus p_1, the sequence defined

by $p_n = f\big(p_{n-1}\big), n \geq 2$, converges to the unique fixed point p in $\big[0.5, 1\big]$ (according to *Theorem 2*).

Similarly, according to *Lemma 4*, we can conclude that $f(x) = \frac{1}{1+e^{-\lambda x}}, f : [0,1] \rightarrow [0,1]$ has the same fixed point p that it has when changing domain and image sets ($f : [0.5,1] \rightarrow [0.5,1]$). In agreement with *Lemma 5*, the sequence defined by $p_n = f\big(p_{n-1}\big), n \geq 1$, also converges to the unique fixed point p in $\big[0,1\big]$.

Therefore, a neuron with self-feedback as its only incoming relation and with $f(x) = \frac{1}{1+e^{-\lambda x}}, \lambda > 0$ as transfer function, always converges to a unique fixed-point. ■

B. Influence of neurons on FRCNs' performance

Next, we analyze the behavior of boundary, negative and positive neurons on performance. More specifically, we are interested in determining the contribution of each neuron to FRCNs' classification process.

1) Boundary and negative neurons: These neurons have self-connections and no other neuron has influence on them. According to *Theorem 3* and *Corollary 1.1*, boundary and negative neurons converge to a unique fixed point, which only depends on the λ value. Although FRCNs use $\lambda = 5$ for every neuron's transfer function, *Corollary 1.1* holds for any positive λ value. This implies that the convergence of such neurons to a unique fixed-point attractor cannot be prevented by altering the aforementioned parameter.

2) Positive neurons: Such neurons have self-connections, but they are also negatively connected with each other. Thus, each positive neuron is positively influenced by itself and negatively influenced by all other positive neurons. The activation rules for any pair of these neurons are:

$$P_x^{(t+1)} = \frac{1}{1 + e^{-\lambda\left(P_x^{(t)} - \left[\sum_{i=1}^{M} P_i^{(t)} - P_x^{(t)}\right]\right)}}, \tag{16}$$

$$P_y^{(t+1)} = \frac{1}{1 + e^{-\lambda\left(P_y^{(t)} - \left[\sum_{i=1}^{M} P_i^{(t)} - P_y^{(t)}\right]\right)}}. \tag{17}$$

According to (16) and (17), we have that:

$$P_x^{(t+1)} = f(P_x^{(t)} - \big[\sum_{i=1}^{M} P_i^{(t)} - P_x^{(t)}\big])$$

and

$$P_y^{(t+1)} = f(P_y^{(t)} - \big[\sum_{i=1}^{M} P_i^{(t)} - P_y^{(t)}\big]).$$

To define an order relation among the activation values of positive neurons, and taking into account that the sigmoid function monotonically increases, we say that:

$$P_x^{(t+1)} < P_y^{(t+1)}$$

if and only if

$$P_x^{(t)} - \big[\sum_{i=1}^{M} P_i^{(t)} - P_x^{(t)}\big] < P_y^{(t)} - \big[\sum_{i=1}^{M} P_i^{(t)} - P_y^{(t)}\big].$$

Consequently, $2P_x^{(t)} < 2P_y^{(t)}$ and

$$P_x^{(t+1)} < P_y^{(t+1)} \iff P_x^{(t)} < P_y^{(t)}.$$

Inductively, we can establish that $P_x^{(t-1)} < P_y^{(t-1)}$ and so on, until reaching the initial activation values and $P_x^{(0)} < P_y^{(0)}$. This relation yields

$$P_x^{(t+1)} < P_y^{(t+1)} \iff P_x^{(t)} < P_y^{(t)}$$

for every iteration t. Analogously,

$$P_x^{(t+1)} > P_y^{(t+1)} \iff P_x^{(t)} > P_y^{(t)}$$

and

$$P_x^{(t+1)} = P_y^{(t+1)} \iff P_x^{(t)} = P_y^{(t)}$$

for every iteration t. From this result we can conclude that the order between any pair of positive neurons remains invariant through the whole FRCN reasoning process. It also holds that the ranking among all positive neurons also remains invariant. This result is in line with the results in [15] and explains why the connections among the positive regions are not necessary to maintain FRCNs' performance.

The ranking can be also useful for deriving further insight. Assuming that $P_x^{(t)}$ is not the highest activation among the positive neurons at the t-th iteration, it will never be the highest at any iteration, because of the invariance of the ranking. Then, with $M \geq 2$ we can assert that:

$$P_x^{(t)} - \big[\sum_{i=1}^{M} P_i^{(t)} - P_x^{(t)}\big] = 2P_x^{(t)} - \sum_{i=1}^{M} P_i^{(t)} \leq 0$$

$$\implies P_x^{(t+1)} = \frac{1}{1 + e^{-\lambda\left(2P_x^{(t)} - \sum_{i=1}^{M} P_i^{(t)}\right)}} \leq \frac{1}{2}.$$

Lemma 6. *In an FRCN with M decisions, there will always be $M-1$ positive neurons with activation values smaller or equal to $\frac{1}{2}$ when $t > 1$.*

Corollary 2.1. *In an FRCN, there will be at most one neuron with activation value higher or equal to $\frac{1}{2}$ when $t > 1$.*

Even though the order relation remains invariant, these neurons could be stable, cyclic or chaotic. These possible states might have direct influence on the network's interpretation. Ideally, we should have a stable neural system. Let us assume then that positive neurons are stable. Then, after reaching such a state, for every x it holds that:

$$P_x^{(t)} = P_x^{(t+1)} = \frac{1}{1 + e^{-\lambda\left(P_x^{(t)} - \left[\sum_{i=1}^{M} P_i^{(t)} - P_x^{(t)}\right]\right)}}. \tag{18}$$

After some algebraic transformations, we obtain

$$\sum_{i=1}^{M} P_i^{(t)} = \frac{\ln\left(\frac{1-P_x^{(t)}}{P_x^{(t)}}\right)}{\lambda} + 2P_x^{(t)}. \tag{19}$$

This result tells us the sum of all activation values depends on the value of $P_x^{(t)}$. It must be noticed that P_x could be any positive neuron. Let us define

$$s(x) = \frac{\ln\left(\frac{1-x}{x}\right)}{\lambda} + 2x, \tag{20}$$

as the function that calculates such a sum, where the activation values are within the $(0, 1)$ interval.

A close inspection to the properties of $s(x)$ yields that it has vertical asymptotes at $x = 0$ and $x = 1$, at which the function tends to positive and negative infinity, respectively. On the other hand, this function also has a relative minimum at $\frac{1}{2} - \sqrt{\frac{\lambda-2}{4\lambda}}$ and a relative maximum at $\frac{1}{2} + \sqrt{\frac{\lambda-2}{4\lambda}}$, when $\lambda > 2$. If $\lambda \leq 2$, the function is monotone-decreasing and thus injective. Therefore, if $s(x)$ is injective in all its domain or particularly in a sub-domain, then a specific value can be produced by a single input in all its domain or its sub-domain, respectively. This means that, if the positive neurons are stable and $\lambda <= 2$, such neurons will converge to a fixed-point attractor. As FRCNs employ $\lambda = 5$ we can state that positive neurons converge to three values at most for any value of $s(x)$. This holds because the function $s(x)$ produces the same value for at most three inputs in $(0, 1)$. As such, there are at most three disjoint sets of positive neurons such that all neurons in a set converge to the same fixed point. Let us assume that p_1, p_2, p_3 refer to these fixed points, then we have that:

$$0 < p_1 \leq \frac{1}{2} - \sqrt{\frac{\lambda-2}{4\lambda}} \leq p_2 \leq \frac{1}{2} + \sqrt{\frac{\lambda-2}{4\lambda}} \leq p_3 < 1.$$

In the case of FRCN models we have:

$$0 < p_1 \leq 0.1127 \leq p_2 \leq 0.8872 \leq p_3 < 1.$$

3) Decision neurons: Such neurons do not influence other neurons or themselves. Let us formalize the equations to obtain the $(t+1)$-th activation values for decision neurons given the activation values at the t-th iteration,

$$D_x^{(t+1)} = f\big(\sum_{B_i \to D_x} 0.5B_i^{(t)} - N_x^{(t)} + P_x^{(t)} - \big[\sum_{i=1}^{M} P_i^{(t)} - P_x^{(t)}\big]\big), \tag{21}$$

$$D_y^{(t+1)} = f\big(\sum_{B_i \to D_y} 0.5B_i^{(t)} - N_y^{(t)} + P_y^{(t)} - \big[\sum_{i=1}^{M} P_i^{(t)} - P_y^{(t)}\big]\big). \tag{22}$$

Since boundary neurons may influence or not a particular decision neuron, $\sum_{B_i \to D_x} 0.5B_i^{(t)}$ will only involve boundary neurons connected to D_x. When labeling new observations, we need to define an order relation among the activation values of decision neurons. If the decision class is determined at the $(t+1)$-th iteration, the class associated with D_y is taken over D_x if and only if $D_x^{(t+1)} < D_y^{(t+1)}$. As the sigmoid function monotonically increases, we can say that:

$$D_x^{(t+1)} < D_y^{(t+1)}$$

if and only if

$$\sum_{B_i \to D_x} 0.5B_i^{(t)} - N_x^{(t)} + P_x^{(t)} - \big[\sum_{i=1}^{M} P_i^{(t)} - P_x^{(t)}\big]$$

$$< \sum_{B_i \to D_y} 0.5B_i^{(t)} - N_y^{(t)} + P_y^{(t)} - \big[\sum_{i=1}^{M} P_i^{(t)} - P_y^{(t)}\big],$$

which implies that

$$\sum_{B_i \to D_x} 0.5B_i^{(t)} - N_x^{(t)} + 2P_x^{(t)} < \sum_{B_i \to D_y} 0.5B_i^{(t)} - N_y^{(t)} + 2P_y^{(t)}. \tag{23}$$

As mentioned, we should perform a large enough number of iterations for the FRCN to converge, otherwise the reasoning process would not be stable. Let us suppose that such an iteration is reached, then *Corollary 1.1* ensures that negative and boundary neurons are convergent to a fixed-point attractor. As a consequence, we state that:

$$N_x^{(t)} = N_y^{(t)} = B_i^{(t)}, \forall i.$$

Given the fact that FRCN-based models always use $\lambda = 5$, we can numerically approximate with high precision the aforementioned fixed-point attractor, which is $b \approx 0.9930$ (we use b for clarity in formulas). Therefore, the inequality in (23) can be further simplified:

$$\sum_{B_i \to D_x} 0.5b + 2P_x^{(t)} < \sum_{B_i \to D_y} 0.5b + 2P_y^{(t)},$$

$$0.5b\big(\sum_{B_i \to D_x} 1 - \sum_{B_i \to D_y} 1\big) < 2\big(P_y^{(t)} - P_x^{(t)}\big),$$

$$b\big(\sum_{B_i \to D_x} 1 - \sum_{B_i \to D_y} 1\big) < 4\big(P_y^{(t)} - P_x^{(t)}\big). \tag{24}$$

This result suggests that, when discriminating between two decision classes, the decision neuron receiving more boundary connections will be favored. The inequality in (24) also brings to life four special cases:

- If both decision neurons D_x and D_y receive the same number of boundary connections, the decision neuron with the highest activation value will be the one associated with the positive neuron having the highest activation value. Besides, since the ranking among positive neurons is invariant, the preferred neuron between D_x and D_y will be D_y iff $P_y^{(0)} > P_x^{(0)}$, or D_x iff $P_x^{(0)} > P_y^{(0)}$, otherwise both neurons have the same activation value. Extending this result to the whole FRCN, if every decision neuron receives the same number of boundary connections, the classification goes to the decision class linked to D_x iff $P_x^{(0)} > P_y^{(0)}, \forall y$. Long story short, when determining the decision class of new instances, only the initial activation values of positive neurons matter.
- If $P_x^{(t)}$ and $P_y^{(t)}$ are equal or close enough ($|P_y^{(t)} - P_x^{(t)}| < \frac{b}{4}$), the decision neuron with the highest activation value (between D_x and D_y) will be the one connected to more boundary neurons. This would suggest that boundary regions do matter when classifying new instances, however, the contribution of each boundary neuron to the decision values will always be the same.
- If neither $P_x^{(t)}$ nor $P_y^{(t)}$ are the positive neurons having the maximal activation value, then $|P_y^{(t)} - P_x^{(t)}| < \frac{1}{2}$ (according to *Corollary* 2.1). Thus, if D_y (D_x) exceeds by at least two the number of boundary connections of D_x (D_y), the first one will be chosen.

- If D_y has at least five more boundary connections than D_x, then D_y will be the decision class to be produced. If D_y has at least five more boundary connections than any other neuron, then D_y will be the neuron having the maximal activation value. Furthermore, the classification will always be the class linked to this neuron, no matter the initial activation values.

In summary, negative regions have no influence on FRCNs' performance, while the ranking of positive neurons' activation values and the number of boundary neurons connected to each decision neuron have high impact.

IV. The Simpler, The Better

In this section, we present two simple solutions to overcome the theoretical limitations of FRCN models.

A. Using linear processing units

The first approach consists in replacing the sigmoid boundary and negative neurons with linear ones. This modification has some positive implications. Firstly, the boundary and negative neurons will contribute to FRCNs' performance according with their initial activation values. Secondly, linear neurons with no incoming connection lead to fixed-point attractors as their activation values are not altered. Such fixed points are not unique and depend on neurons' initial activation values, as desired when classifying patterns.

Aiming at illustrating the power behind this simple modification, let us consider a pattern classification problem having three decision classes. Besides, let us assume an extreme case in which fuzzy-rough positive regions are empty, therefore the decision class is determined by using information coming from negative and boundary regions. A setting for this hypothetical scenario is $P_c^{(0)} = 0, \forall c$, $N_1^{(0)} = 0.4$, $N_2^{(0)} = 0.04$, $N_3^{(0)} = 0.26$, $B_1^{(0)} = 0.6$, $B_2^{(0)} = 0.74$, $B_3^{(0)} = 0.75$, while decision neurons are inactive (i.e., $D_c^{(0)} = 0, \forall c$).

Fig. 3 shows the recurrent reasoning process of an FRCN for this hypothetical classification problem. The vertical axis represents the activation value of positive, negative, boundary and decision neurons in each iteration, while the horizontal axis is the iterations. The reader can notice that in the case of the positive neurons, only P_3 seems to be visualized since all positive activation values are overlapped.

Aiming at making a decision, the FRCN relies on the boundary and negative information. The latter suggests rejecting the decision classes in the following order: $D_1 \succ D_3 \succ D_2$, therefore implying that D_2 is the most suitable decision class. Likewise, the boundary information suggests accepting D_2 and D_3 with the same degree, while it rejects the hypothesis of x to be associated with D_1. Overall, the evidence extracted from information granules advocates for the strong rejection of D_1 while accepting D_2 over D_3.

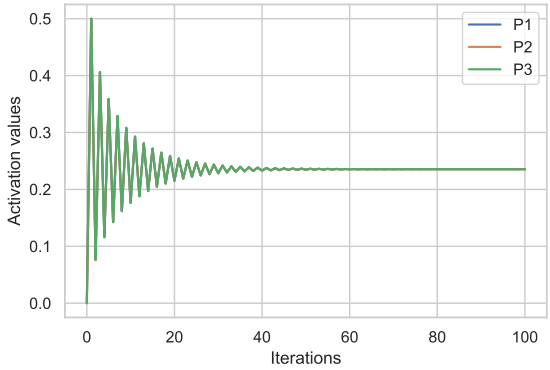
B. Fuzzy-Rough Cognitive Regression

After the theoretical analysis conducted in Section III, the reader might wonder whether a recurrent reasoning rule significantly contributes to FRCNs' performance. In this section, we put forth a simpler model referred to as *Fuzzy-Rough Cognitive Regression* (FRCR) to combine the information derived from fuzzy-rough information granules.

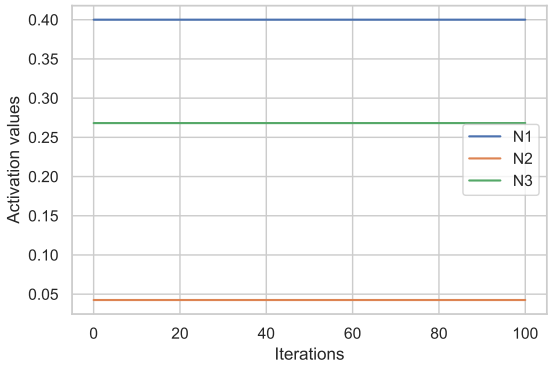
The confidence value for the c-th decision class as computed by the FRCR model is expressed as follows:

$$\vartheta_c(y) = f\Big(\frac{\sum_{x\in\mathcal{U}} \mathcal{T}(\varphi(x,y), \mu_{POS(X_c)}(x))}{\sum_{x\in\mathcal{U}} \mu_{POS(X_c)}(x)} - \frac{\sum_{x\in\mathcal{U}} \mathcal{T}(\varphi(x,y), \mu_{NEG(X_c)}(x))}{\sum_{x\in\mathcal{U}} \mu_{NEG(X_c)}(x)} + \frac{\gamma \sum_{x\in\mathcal{U}} \mathcal{T}(\varphi(x,y), \mu_{BND(X_c)}(x))}{\sum_{x\in\mathcal{U}} \mu_{BND(X_c)}(x)}\Big) \tag{25}$$

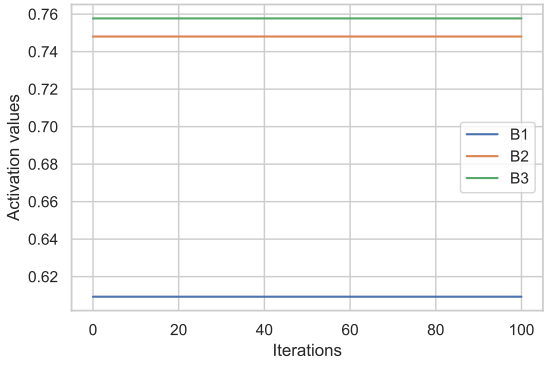
where y is the instance to be classified, $f(x)$ is the sigmoid transfer function, $\varphi(x,y)$ is the similarity degree between y



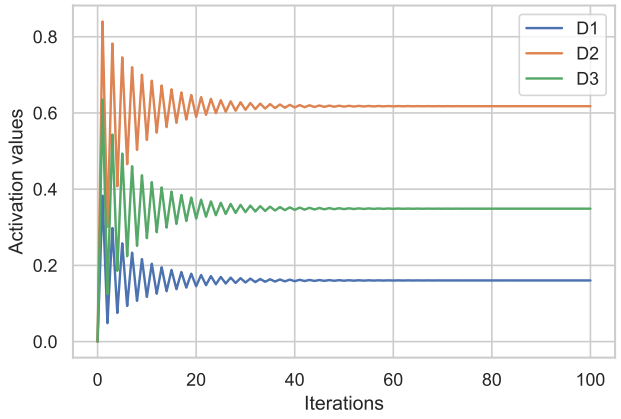
(a) Positive neurons.



(b) Negative neurons.



(c) Boundary neurons.



(a) Decision neurons.

Fig. 3: Activation values of positive, negative, boundary and decision neurons for a pattern classification problem having three decision classes.

and $x \in \mathcal{U}$, while $\mathcal{T}(.,.)$ is a conjunction function. Moreover, $\mu_{POS(X_c)}(x)$, $\mu_{NEG(X_c)}(x)$ and $\mu_{BND(X_c)}(x)$ stand for the membership functions attached to the positive, negative and boundary fuzzy-rough regions, respectively. Finally, $\gamma = 1/M$ regulates the contribution of the boundary information to the confidence value of each decision.

Therefore, given an unlabeled instance y, the classification process consists in assigning to y the class having the highest confidence value. The reader can notice that this model could be seen as a type of logistic regression algorithm in which the inputs come from the fuzzy-rough granules.

C. Computational complexity analysis

In this subsection, we compare the computational complexity of the original FRCN algorithm and the proposed FRCR model. Let $\mathcal{V}$ and $\mathcal{A}$ denote the test set and the attribute set, respectively. The granulation process is an active component in both models and its temporal complexity is $O(|\mathcal{U}|^2 |\mathcal{D}| |\mathcal{A}|)$. In the FRCN model, the complexity of building the network is $O(|\mathcal{D}|^2)$, while calculating the initial activation values for every object in the test dataset costs $O(|\mathcal{U}| |\mathcal{V}| |\mathcal{D}| |\mathcal{A}|)$. Also, the complexity of the exploitation (test) phase is $O(|\mathcal{V}| |\mathcal{D}|^2 T)$, whereas T is the maximal number of iterations. On the other hand, the complexity of the exploitation phase in the FRCR model is $O(|\mathcal{U}| |\mathcal{V}| |\mathcal{D}| |\mathcal{A}|)$. The reader can notice that the computational complexity of the FRCN model is higher when compared with the new classifiers since the step of building the network is no longer necessary.

V. Numerical simulations

In this section, we perform some numerical simulations to assess the discriminatory power of the two models proposed in Section IV, which correct the theoretical limitations of the FRCN classifier. Actually, we would only need to verify that the new algorithms retain the prediction power of the FRCN classifier. The reader is referred to [7], [11], [12], [10] and [23] for further detail on the FRCNs' prediction performance on structured classification problems.

A. Traditional benchmark problems

In our simulations, we use 140 structured (traditional) pattern classification datasets reported in [11]. In these problems, the number of instances ranges from 14 to 12,906, the number of decision classes ranges from 2 to 100 and the number of attributes from 2 to 262. These benchmark problems include 13 noisy and 47 imbalanced datasets, with the imbalance ratio fluctuating between 5:1 and 2160:1.

Numerical attributes have been normalized to avoid potential out-of-range issues when computing the distance function. Furthermore, whenever necessary, we replaced missing values with the mean or the mode depending on whether the attribute was numerical or nominal, respectively.

B. State-of-the-art algorithms

Although the goal of this research is not devoted to claiming the numerical superiority of the FRCN models, we will compare them against other granular and state-of-the-art classifiers for the sake of completeness. The granular models used in our experiments include: k-Nearest Neighbors (kNN) with $k = 3$, the K^* classifier, Fuzzy-Rough k-Nearest Neighbors (FRNN) and Vaguely-quantified k-Nearest Neighbors (VQNN). The traditional classifiers include: Multilayer Perceptron (MLP), Support Vector Machines (SVM), Naïve Bayes (NB), Decision Tree (DT) and Random Forest (RF).

No classification model performed perform hyperparameter tuning. Instead, we used the default parameter settings as provided in the Weka v3.6.11 software tool [24]. Usually, specific parameter settings improves the algorithms' performance, but it can be questioned whether a model that is quite sensitive to its hyperparameters should be considered "better" than another that performs simply well for a wide variety of scenarios. On the other hand, it goes without saying that obtaining such a degree of optimization goes at the expense of increasing the computational burden of training the model.

In the case of our granular classifiers, the default parameter setting is as follows. The *Heterogeneous Manhattan-Overlap Metric* [25] is adopted as the dissimilarity function. This distance function computes the normalized Euclidean distance between numerical attributes and an overlap metric for nominal ones. Moreover, we adopted the well-known Lukasiewicz operator $\mathcal{T}(x, y) = \max\{0, x + y - 1\}$ to implement the fuzzy conjunction and implication operations. Finally, the λ value of the function $f(x)$ is set to 2.0.

C. Statistical analysis and discussion

Aiming at quantifying algorithms' performance, we utilized Cohen's Kappa coefficient [26]. This measure computes the inter-rater agreement for categorical items. It is deemed a more robust measure than the standard accuracy since it takes into account the agreement occurring by chance.

The first experiment is devoted to comparing the prediction ability of the proposed fuzzy-rough classifiers. The Wilcoxon

signed rank test does not reject the null hypothesis for the 95% of the confidence interval (i.e., p-value $= 0.573 > 0.05$), which means that both variants perform comparably when it comes to the number of correctly classified instances. In order to simplify the experiments, FRCR will be adopted to perform the remaining simulations in this section.

Fig. 4 summarizes the average Kappa value attained by each granular classifier after performing a 10-fold cross-validation for each dataset. The reader can observe that FRCR stands as the best-performing algorithm when compared with the other methods across the benchmark problems.

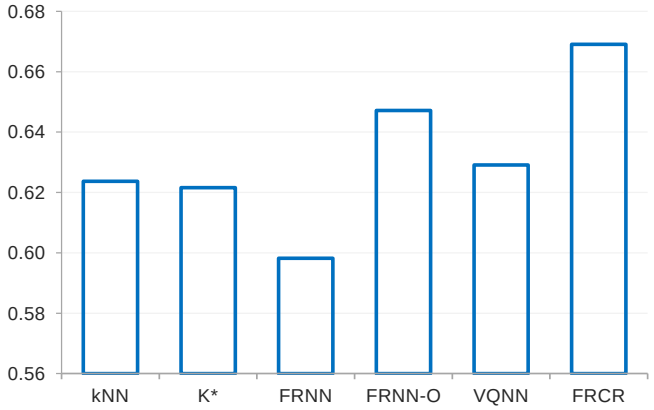


Fig. 4: Performance of granular classifiers.

The Friedman test suggests rejecting the H_0 hypothesis (p-value $=$ 5.27E-10 < 0.05) for a confidence interval of 95%. This suggests that there are significant differences between at least two methods across benchmark.

Table I shows the p-values reported by the Wilcoxon signed rank test, the negative (R^-) and the positive (R^+) ranks, and the corrected p-values according to Holm where FRCR is the control method. This post-hoc attempts to control for type-I errors that might arise when performing pairwise comparisons. The results indicate that all null hypotheses can be rejected for a significance level of 0.05, thus confirming the superiority in performance of the FRCR algorithm.

TABLE I: Pairwise analysis for granular classifiers where FRCR is used as the control method.

Algorithm	p-value	R^-	R^+	Holm	Hypothesis
FRNN	5.73E-9	39	96	2.86E-8	Rejected
IBk	1.02E-7	34	99	4.07E-7	Rejected
K*	3.67E-5	49	85	1.10E-4	Rejected
VQNN	1.21E-3	52	83	2.41E-3	Rejected
FRNN-O	3.52E-2	57	77	3.52E-2	Rejected

Fig. 5 displays the average Kappa value computed by each traditional classifier after performing a 10-fold cross-validation for each benchmark problem. In this experiment, FRCR, RF and MLP reported the highest Kappa values, while NB turned to be the worst-performing model.

The Friedman test suggests rejecting the H_0 hypothesis (p-value $=$ 8.47E-10 < 0.05) for a significance level of 0.05. Table II shows the p-values reported by the Wilcoxon signed rank test, the negative (R^-) and the positive (R^+) ranks, and the corrected p-values according to Holm where FRCR is the control method. The corrected p-values suggest rejecting the null hypotheses for the following pairwise comparisons: FRCR vs. NB, FRCR vs. SVM, FRCR vs. DT. The null hypotheses for the pairs FRCR vs. MLP and FRCR vs. RF are not rejected for the 95% of the confidence interval.

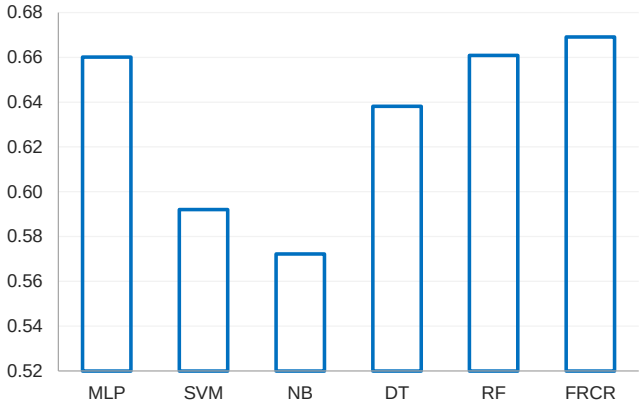


Fig. 5: Performance of traditional classifiers.

TABLE II: Pairwise analysis for traditional classifiers where FRCR is used as the control method.

Algorithm	p-value	R^-	R^+	Holm	Hypothesis
NB	1.06E-10	36	100	5.31E-10	Rejected
SVM	4.69E-6	45	88	1.88E-5	Rejected
DT	2.49E-3	44	88	7.48E-3	Rejected
MLP	5.94E-1	58	76	1.0	Not Rejected
RF	8.46E-1	66	67	1.0	Not Rejected

Long story short, the results have shown that the proposed models perform significantly well when compared against both granular and traditional classifiers. Even more important is the fact that our algorithms retain the predictive power of FRCNs while being more simple and robust.

D. *Image classification*

In this subsection, we will develop a case study illustrating how to use the proposed FRCR method to solve unstructured classification problems such as image classification. The selected dataset is the well-known Street View House Numbers (SVHN) dataset[1], which contains 73257 training images and 10 decision classes (i.e., the digits). Figure 6 shows an excerpt of this dataset. To perform the simulations, we split the dataset such that 80% of the instances are used to either train or build the models while the remaining 20% are used for testing the classifiers' generalization capability.

This case study comes with the problem that FRCN-based classifiers operate on structured datasets with well-defined features (although they might contain noise, missing values, etc.). Hence, if we want to use the FRCR model for image classification, then we would need first to extract relevant features describing the images. This is the idea behind transfer learning [27] in which a pre-trained model extracts the features, which are used to feed a second classifier.

[1]http://ufldl.stanford.edu/housenumbers/

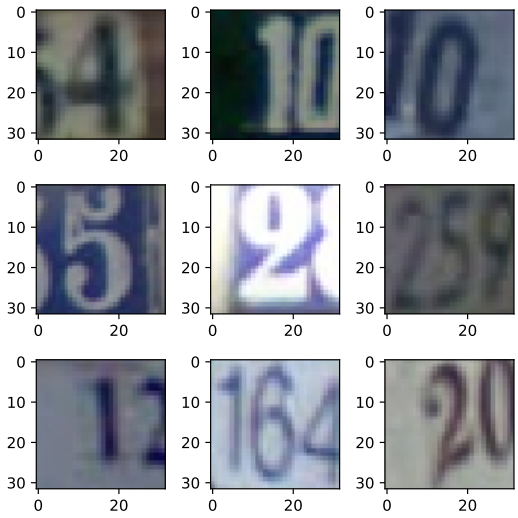


Fig. 6: Excerpt of the SVHN dataset for image classification.

In short, the simulations in this subsections are oriented to comparing the performance of the FRCR method against deep learning models when the features are given.

In order to extract the features, we will use a Convolutional Neural Network (CNN) [28]–[31] comprised of three blocks, each containing two 2D convolution layers and a max pooling layer (see Figure 7). The number of filters attached to convolution layers in each block is 32, 64 and 128, respectively, while the stride size of the max pooling was set to one. To train the model, the latter block is connected with a hidden layer having 128 ReLU neurons, which is connected with the output layer. These layers are not visualized in Figure 7 as they will be removed when coupling the CNN with other classifiers as done in transfer learning architectures.

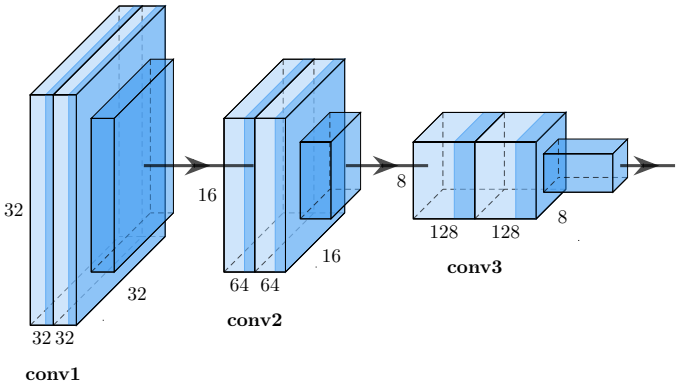


Fig. 7: Architecture of the CNN model used to extract the features from the SVHN dataset.

The CNN involves 551,018 learnable parameters and produces 2048 features that will feed the following deep learning models: a Long short-term memory (LSTM) [32], [33] with 10 cells, a Bidirectional Long short-term memory (BiLSTM) [34], [35] and a Recurrent Neural Network (RNN) [36]. These algorithms used the backpropagation learning algorithm, and the popular ADAM optimization method [37] with learning rate equal to 0.001 and 50 learning epochs. All models (including ours) use the features extracted by the CNN, therefore resulting in four coupled algorithms: CNN-RNN, CNN-LSTM, CNN-BiLSTM and CNN-FRCR. In that way, all models will operate on the same pieces of information.

The Kappa values reported by these classification mdodels on the test set are as follow: CNN-RNN (0.931), CNN-LSTM (0.938), CNN-BiLSTM (0.936) and CNN-FRCR (0.941). The results suggest that, when the features are already extracted, the proposed FRCR method performs as well as most powerful deep learning algorithms. However, it would not be realistic to assume that such appealing results can easily be generalized to other unstructured problems. Even if they were, we will need for other algorithms to extract the features since FRCN-based classifiers were designed to deal with classification problems were explicit features are available.

VI. Concluding Remarks

In this paper, we conducted a theoretical analysis of FRCNs and the contribution of their building blocks to algorithm's performance. FRCNs' dynamical analysis revealed that negative and boundary neurons will always converge to a unique fixed-point attractor, while the ranking of positive neurons will remain invariant during the whole reasoning process. Even more serious is the fact that negative neurons have no influence on algorithm's decision process. These findings motivated the proposal of two simpler fuzzy-rough classifiers that overcome FRCNs' theoretical limitations.

With regard to the simulations, the *Fuzzy-Rough Cognitive Regression* was capable of outperforming most granular and traditional classifiers used for comparison. It is true that such results might change if we optimize algorithms' hyperparameters for each dataset, but this usually comes with a significant increase in the time and effort required to build the model. Beyond the competitive prediction rates, what we consider of utmost relevance is the ability of our models to elucidate their reasoning process at a granular level by means of fuzzy-rough inclusion degree equations. Not many machine learning classifiers allow for such a relevant feature. The future research will be oriented to reducing the processing time when deriving the information granules in large datasets.

Acknowledgment

The authors would like to thank the anonymous reviewers for their valuable and constructive feedback.

References

[1] W. Pedrycz and G. Vukovich, "Granular neural networks," *Neurocomputing*, vol. 36, no. 1, pp. 205–224, 2001.

[2] X. Xu, G. Wang, S. Ding, X. Jiang, and Z. Zhao, "A new method for constructing granular neural networks based on rule extraction and extreme learning machine," *Pattern Recognition Letters*, vol. 67, pp. 138–144, 2015.

[3] X. Hu, W. Pedrycz, and X. Wang, "Optimal allocation of information granularity in system modeling through the maximization of information specificity: A development of granular input space," *Applied Soft Computing*, vol. 42, pp. 410–422, 2016.

[4] Yan-Qing Zhang, M. D. Fraser, R. A. Gagliano, and A. Kandel, "Granular neural networks for numerical-linguistic data fusion and knowledge discovery," *IEEE Transactions on Neural Networks*, vol. 11, no. 3, pp. 658–667, 2000.

[5] S. K. Pal, B. Dasgupta, and P. Mitra, "Rough self organizing map," *Applied Intelligence*, vol. 21, no. 3, pp. 289–299, 2004.

[6] A. Ganivada, S. Dutta, and S. K. Pal, "Fuzzy rough granular neural networks, fuzzy granules, and classification," *Theoretical Computer Science*, vol. 412, no. 42, pp. 5834–5853, 2011.

[7] G. Nápoles, I. Grau, E. Papageorgiou, R. Bello, and K. Vanhoof, "Rough Cognitive Networks," *Knowledge-Based Systems*, vol. 91, pp. 46–61, 2016.

[8] A. Abraham, R. Falcon, and R. Bello, *Rough Set Theory: A True Landmark in Data Analysis*. Springer Verlag, 2009.

[9] Y. Yao, "Three-way decision: an interpretation of rules in rough set theory," in *Rough Sets and Knowledge Technology*, P. Wen, Y. Li, L. Polkowski, Y. Yao, S. Tsumoto, and G. Wang, Eds. Springer Verlag, 2009, pp. 642–649.

[10] G. Nápoles, I. Grau, R. Falcon, R. Bello, and K. Vanhoof, "A Granular Intrusion Detection System using Rough Cognitive Networks," in *Recent Advances in Computational Intelligence in Defense and Security*, R. Abielmona, R. Falcon, N. Zincir-Heywood, and H. Abbass, Eds. Springer Verlag, 2016, ch. 7, pp. 169–191.

[11] G. Nápoles, R. Falcon, E. Papageorgiou, R. Bello, and K. Vanhoof, "Rough cognitive ensembles," *International Journal of Approximate Reasoning*, vol. 85, pp. 79–96, 2017.

[12] G. Nápoles, C. Mosquera, R. Falcon, I. Grau, R. Bello, and K. Vanhoof, "Fuzzy-rough cognitive networks," *Neural Networks*, vol. 97, pp. 19–27, 2018.

[13] C. Cornelis, M. De Cock, and A. M. Radzikowska, "Fuzzy rough sets: from theory into practice," *Handbook of Granular Computing*, pp. 533–552, 2008.

[14] D. Dubois and H. Prade, "Rough fuzzy sets and fuzzy rough sets," *International Journal of General System*, vol. 17, no. 2-3, pp. 191–209, 1990.

[15] K. V. Marnick Vanloffelt, Gonzalo Nápoles, "Fuzzy-rough cognitive networks: Building blocks and their contribution to performance," in *Proceedings of the International Joint Conference on Neural Networks (IJCNN 2020)*. IEEE, 2020.

[16] M. Inuiguchi, W.-Z. Wu, C. Cornelis, and N. Verbiest, "Fuzzy-rough hybridization," in *Springer Handbook of Computational Intelligence*. Springer, 2015, pp. 425–451.

[17] B. Kosko, "Hidden patterns in combined and adaptive knowledge networks," *International Journal of Approximate Reasoning*, vol. 2, no. 4, pp. 377–393, 1988.

[18] I. Á. Harmati, M. F. Hatwágner, and L. T. Kóczy, "On the existence and uniqueness of fixed points of fuzzy cognitive maps," in *Proceedings of the International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*. Springer, 2018, pp. 490–500.

[19] T. Kottas, Y. Boutalis, and M. Christodoulou, "Bi-linear adaptive estimation of fuzzy cognitive networks," *Applied Soft Computing*, vol. 12, no. 12, pp. 3736–3756, 2012.

[20] W. Rudin *et al.*, *Principles of mathematical analysis*. McGraw-hill New York, 1964, vol. 3.

[21] A.-L. Cauchy, *Cours d'analyse de l'École Royale Polytechnique*. Cambridge University Press, 2009, vol. 3, pp. 17–30.

[22] T. M. Apostol, *Calculus, Volume I, One-Variable Calculus, with an Introduction to Linear Algebra*. John Wiley & Sons, 2007, vol. 1.

[23] G. Felix, G. Nápoles, R. Falcon, R. Bello, and K. Vanhoof, "Performance analysis of granular versus traditional neural network classifiers: Preliminary results," in *Proceedings of the International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA 2018)*. IEEE, 2018, pp. 1–6.

[24] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The weka data mining software: an update," *ACM SIGKDD explorations newsletter*, vol. 11, no. 1, pp. 10–18, 2009.

[25] D. R. Wilson and T. R. Martinez, "Improved heterogeneous distance functions," *Journal of Artificial Intelligence Research*, vol. 6, no. 1, pp. 1–34, 1997.

[26] J. Cohen, "A coefficient of agreement for nominal scales," *Educational and Psychological Measurement*, vol. 20, no. 1, pp. 37–46, 1960.

[27] S. J. Pan, J. T. Kwok, Q. Yang *et al.*, "Transfer learning via dimensionality reduction," in *Proceedings of the 23rd AAAI Conference on Artificial Intelligence*, vol. 8, 2008, pp. 677–682.

[28] L. Hertel, E. Barth, T. Käster, and T. Martinetz, "Deep convolutional neural networks as generic feature extractors," in *Proceedings of the International Joint Conference on Neural Networks (IJCNN 2015)*. IEEE, 2015, pp. 1–4.

[29] T. Wiatowski and H. Bölcskei, "A mathematical theory of deep convolutional neural networks for feature extraction," *IEEE Transactions on Information Theory*, vol. 64, no. 3, pp. 1845–1866, 2017.

[30] W. Rawat and Z. Wang, "Deep convolutional neural networks for image classification: A comprehensive review," *Neural Computation*, vol. 29, no. 9, pp. 2352–2449, 2017.

[31] P. Sermanet, S. Chintala, and Y. LeCun, "Convolutional neural networks applied to house numbers digit classification," in *Proceedings of the 21st International Conference on Pattern Recognition (ICPR 2012)*. IEEE, 2012, pp. 3288–3291.

[32] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[33] J. Wang, Y. Yang, J. Mao, Z. Huang, C. Huang, and W. Xu, "CNN-RNN: A unified framework for multi-label image classification," in *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR 2016)*. IEEE, 2016, pp. 2285–2294.

[34] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, 1997.

[35] C. Wang, H. Yang, C. Bartz, and C. Meinel, "Image captioning with deep bidirectional LSTMs," in *Proceedings of the 24th ACM International Conference on Multimedia*. ACM, 2016, pp. 988–997.

[36] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, *Learning Internal Representations by Error Propagation*. MIT Press, 1986, pp. 318–362.

[37] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proceedings of the 3rd International Conference on Learning Representations (ICLR 2015)*, 2015.