Feature and Label Association Based on Granulation Entropy for Deep Neural Networks

Marilyn Bello[1,2(✉)], Gonzalo Nápoles[2,3], Ricardo Sánchez[1], Koen Vanhoof[2], and Rafael Bello[1]

[1] Computer Science Department, Central University of Las Villas, Santa Clara, Cuba
`mbgarcia@uclv.cu`
[2] Faculty of Business Economics, Hasselt University, Hasselt, Belgium
[3] Department of Cognitive Science and Artificial Intelligence, Tilburg University, Tilburg, The Netherlands

Abstract. Pooling layers help reduce redundancy and the number of parameters before building a multilayered neural network that performs the remaining processing operations. Usually, pooling operators in deep learning models use an explicit topological organization, which is not always possible to obtain on multi-label data. In a previous paper, we proposed a pooling architecture based on association to deal with this issue. The association was defined by means of Pearson's correlation. However, features must exhibit a certain degree of correlation with each other, which might not hold in all situations. In this paper, we propose a new method that replaces the correlation measure with another one that computes the entropy in the information granules that are generated from two features or labels. Numerical simulations have shown that our proposal is superior in those datasets with low correlation. This means that it induces a significant reduction in the number of parameters of neural networks, without affecting their accuracy.

Keywords: Granular computing · Rough sets · Association-based pooling · Deep learning · Multi-label classification

1 Introduction

Multi-Label Classification (MLC) is a type of classification where each of the objects in the data has associated a vector of outputs, instead of being associated with a single value [8,20]. Formally speaking, suppose $X = R^d$ denotes the d-dimensional instance space, and $L = \{l_1, l_2, \ldots, l_k\}$ denotes the label space with k being the possible class labels. The task of multi-label learning is to estimate a function $h : X \longrightarrow 2^L$ from the multi-label training set $\{(x_i, L_i) \mid 1 \leq i \leq n\}$. For each multi-label example (x_i, L_i), $x_i \in X$ is a d-dimensional feature vector $(x_{i1}, x_{i2}, \ldots, x_{id})$ and $L_i \subseteq L$ is the set of labels associated with x_i. For any unseen instance $x \in X$, the multi-label classifier $h(\cdot)$ predicts $h(x) \subseteq L$ as the

R. Bello et al. (Eds.): IJCRS 2020, LNAI 12179, pp. 225–235, 2020.
https://doi.org/10.1007/978-3-030-52705-1_17

set of proper labels for x. This particular case of classification requires additional efforts in extracting relevant features describing both input and decision domains, since the boundaries regions of decisions usually overlap with each other. This often causes the decision space to be quite complex.

Deep learning [6,10] is a promising avenue of research into the automated extraction of complex data representations at high levels of abstraction. Such algorithms develop a layered, hierarchical architecture of learning and representing data, where higher-level (more abstract) features are defined in terms of lower-level (less abstract) features. For example, pooling layers [6,11,12] provide an approach to down sampling feature maps by summarizing the presence of features in patches of the feature map. Two common pooling methods are average pooling and max pooling, which compute the average presence of a feature and the most activated presence of a feature, respectively.

In the case of MLC, this must be done for both features and labels. Several authors [5,15,17,21] have proposed MLC solutions inspired on deep learning techniques. All these solutions are associated with application domains in which the data have a topological organization (i.e. recognizing faces, coloring black and white images or classifying objects in photographs). In [1] the authors introduced the *association-based pooling* that exploits the correlation among neurons instead of exploiting the topological information as typically occurs when using standard pooling operators. Despite of the relatively good results reported by this model, the function used to quantify the association between problem variables does not seem to be suitable for datasets having poor correlation among their features or labels. An alternative to deal with this issue consists in replacing the correlation measure with a more flexible association estimator.

In this paper, we compute the entropy of the granules that are generated from two problem features or labels. Several methods based on Granular Computing use granules as basic elements of analysis [7,18], so that from two similar granulations of the universe of discourse, similar results must be achieved. One way to measure this similarity between the granulations is to measure the entropy in the data that they generate [19]. The rationale of our proposal suggests that two features (or labels) can be associated if the generated granulations from them have equal entropy. Therefore, the proposal consists in obtaining a universe granulation, where each feature (or label) defines an indiscernibility relation. In this method, the information granules are the set of indiscernible objects with respect to the feature (or label) under consideration.

The rest of the paper is organized as follows. Section 2 presents the theoretical background related to our proposal. Section 3 introduces the new measure to quantify the association between features and labels, and Sect. 4 is dedicated to evaluating its performance in the model on synthetic datasets. Finally, in Sect. 5 we provide relevant concluding remarks.

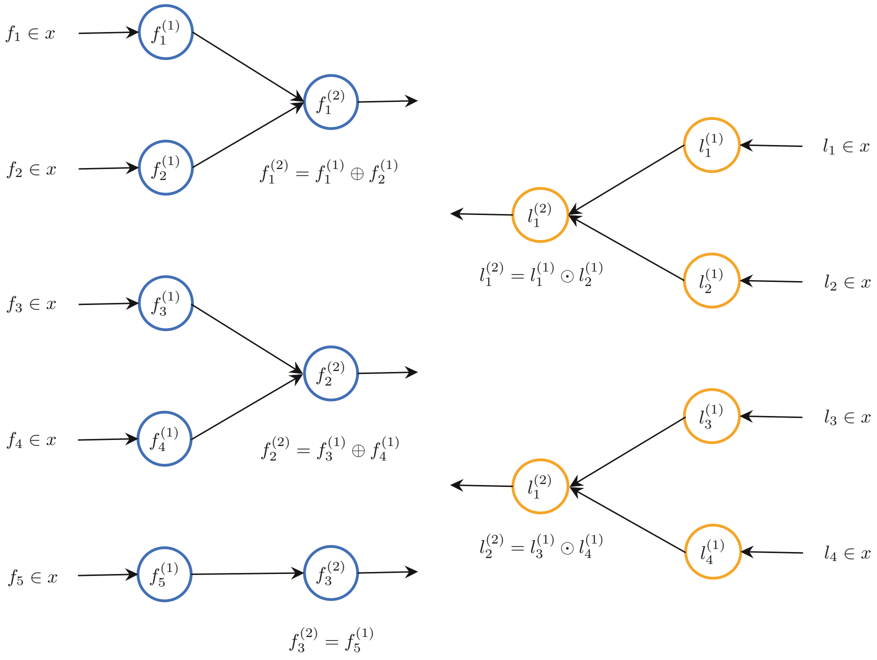
2 Theoretical Background

In this section, we briefly describe the bidirectional neural network to be modified, and the granulation approach used in our proposal.

2.1 Bidirectional Deep Neural Network

Recently, in [1] the authors introduced a new bidirectional network architecture that is composed of stacked association-based pooling layers to extract high-level features and labels in MLC problems. This approach, unlike the classic use of pooling, does not pool pixels but problem features or labels.

The first pooling layer is composed of neurons denoting the problem features and labels (i.e. low-level features and labels), whereas in deeper pooling layers the neurons denote high-level features and labels extracting during the construction process. Each pooling layer uses a function that detects pairs of highly associated neurons, while performing an aggregation operation to derive the pooled neurons. Such neurons are obtained from neurons belonging to the previous layer such that they fulfil a certain association threshold. Figure 1 shows an example where two pooling layers are running for both features (left figure) and labels (right figure). In this example, five high-level neurons were formed from the association of the feature pairs (f_1, f_2) and (f_3, f_4), and the label pairs (l_1, l_2) and (l_3, l_4). The f_5 feature is not associated with any other feature, so it is transferred directly to the $t+1$ pooling layer. In this pooling architecture, $\oplus$ and $\odot$ are the aggregation operators used to conform the pooled neurons.



(a) Feature pooling process (b) Label pooling process

Fig. 1. Bidirectional association-based pooling.

This model uses Pearson's correlation to estimate the association degree between two neurons. Overall, the authors computed the correlation matrix among features and labels, and derive the degree of association of the pooled neurons from the degree of association between each pair of neurons in the previous layer. The pooling process is repeated over aggregated features and labels until a maximum number of pooling layers is reached.

Once the high-level features and labels have been extracted from the dataset, they are connected together with one or several hidden processing layers. Finally, a decoding process [9] is performed to connect the high-level labels to the original ones by means of one or more hidden processing layers. Figure 2 depicts the network architecture resulting five high-level neurons that emerge from the association-based pooling layers. These hidden layers are equipped with either ReLU, sigmoid or hyperbolic tangent transfer functions, therefore conferring the neural system with prediction capabilities.

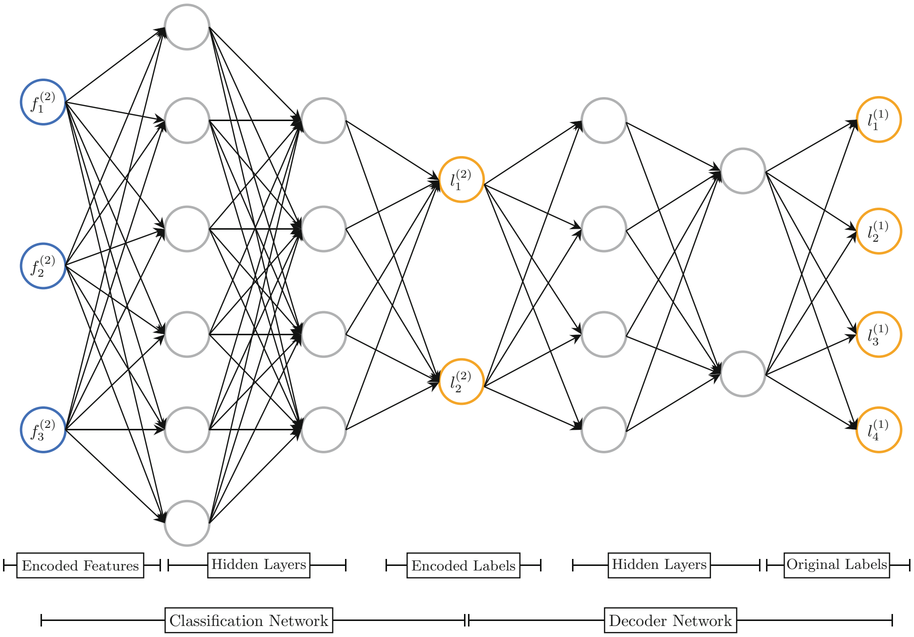


Fig. 2. Neural network architecture using association-based pooling.

It is worth reiterating that this model is aimed at pooling features and labels in traditional MLC problems where neither features nor labels have a topological organization. For example, when using numerical descriptors to encode a protein, it might happen that two distant positions in the sequence are actually close two each other in the tri-dimensional space.

2.2 Universe Granulation

The underlying notion for granulation in classic rough sets [2,13,14] relies on equivalence relations or partitions. Let U be a finite and non-empty universe, and A is a finite non-empty set of features that describe each object. Given a subset of attributes $B \subseteq A$, an indiscernibility relation is defined as $IND = \{(x,y) \in U \times U | \forall b \in B, x(b) = y(b)\}$. This relation is reflexive, symmetric and transitive. The equivalence class $[x]_{IND}$ consists of all elements equivalent to x according to relation IND. The family of equivalence classes $U/IND = \{[x]_{IND} | x \in U\}$ is a partition of the universe.

The indiscernibility relation seems to be excessively restrictive. In presence of numerical attributes, two inseparable objects (according to some similarity relation R [16]) will be gathered together in the same set of non-identical (but reasonably similar) objects. The definition of R may admit that a small difference between features values is considered as unsignificant. This relation delimits whether two objects x and y are inseparable or not, and defines a similarity class where $\bar{R}(x) = \{y \in U | yRx\}$. Equation (1) shows the similarity relation, assuming that $0 \leq \varphi(x,y) \leq 1$ is a similarity function,

$$R : yRx \Leftrightarrow \varphi(x,y) \geq \xi. \tag{1}$$

This weaker binary relation states that objects x and y are deemed inseparable as long as their similarity degree $\varphi(x,y)$ exceeds a similarity threshold $0 \leq \xi \leq 1$. It is worth mentioning that the similarity relation R does not induce a partition of U into a set of equivalence classes but rather a covering [3] of U into multiple similarity classes $\bar{R}(x)$.

3 Feature Association Using the Granulation Entropy

The granular approach in [19] uses the Shannon entropy to characterize partitions of a universe. Two granulations with the same (or similar) entropy value could be considered equivalent. Similarly, the degree of association between two features (or labels) could be determined using the entropy of the granulations they generate. Our method verifies if the coverings (or partitions) generated by two features (or labels) induce similar entropy values.

Let us assume that the problem feature f_1 generates the covering $Cf_1 = \{GF_1, GF_2, \ldots, GF_s\}$ that contains s granules, i.e. the family of similarity classes when only the f_1 feature is considered. Thus, we define $\varphi(x,y) = 1 - |x(f_1) - y(f_1)|$ as a similarity function used in Eq. (1), where $x(f_1)$ and $y(f_1)$ are the values of the feature f_1 in objects x and y. In addition, the l_1 label generates the partition $Pl_1 = \{GL_1, GL_2, \ldots, GL_t\}$ with t granules, i.e. the family of equivalence classes where all objects have exactly the same value on the l_1 label. Since the domain of the label is $\{0,1\}$, this partition will only contain two equivalence classes. Equations (2) and (3) define the probability distributions for the partitions Cf_1 and Pl_1, respectively,

$$D_{Cf_1} = \left\{ \frac{|GF_1|}{|U|}, \frac{|GF_2|}{|U|}, \ldots, \frac{|GF_s|}{|U|} \right\} \tag{2}$$

$$D_{Pl_1} = \left\{ \frac{|GL_1|}{|U|}, \frac{|GL_2|}{|U|}, \ldots, \frac{|GL_t|}{|U|} \right\} \tag{3}$$

where $|\cdot|$ denotes the cardinality of a set. The Shannon entropy function of the probability distributions is defined by Eqs. (4) and (5),

$$H(Cf_1) = -\sum_{i=1}^{s} \left(\frac{|GF_i|}{|U|} \right) log \left(\frac{|GF_i|}{|U|} \right) \tag{4}$$

$$H(Pl_1) = -\sum_{j=1}^{t} \left(\frac{|GL_j|}{|U|} \right) log \left(\frac{|GL_j|}{|U|} \right). \tag{5}$$

The similarities of the granulations generated by two features f_1 and f_2, or two labels l_1 and l_2 can be defined according to the measures $GSE_1(f_1, f_2)$ and $GSE_2(l_1, l_2)$ defined in the Eqs. (6) and (7) respectively,

$$GSE_1(f_1, f_2) = \frac{(1 + E1)}{(1 + E2)} \tag{6}$$

$$GSE_2(l_1, l_2) = \frac{(1 + N1)}{(1 + N2)} \tag{7}$$

where $E1 = min\{H(Cf_1), H(Cf_2)\}$, $E2 = max\{H(Cf_1), H(Cf_2)\}$, $N1 = min\{H(Pl_1), H(Pl_2)\}$, and $N2 = max\{H(Pl_1), H(Pl_2)\}$.

In this way, two features can be associated if $GSE_1(f_1, f_2) \geq \alpha_1$, where α_1 is the association threshold regulating the aggregation of features. In the same way, two labels will be associated if $GSE_2(l_1, l_2) \geq \alpha_2$, where α_2 is the association threshold regulating the aggregation of labels.

In our approach, we estimate the association degree between pairs of pooled neurons from the values of the association matrix calculated for the original features and labels of the problem. Then, the association between two pooled neurons would be performed as the average of the values determined by GSE_1 and GSE_2 for each pair of features (or labels) in these neurons.

Equations (8) and (9) define the association between $f_1^{(v)}$ and $f_2^{(v)}$ (i.e. neurons in the v-th pool of features), and between $l_1^{(w)}$ and $l_2^{(w)}$ (i.e. neurons in the w-th pool of labels), respectively,

$$SP_1(f_1^{(v)}, f_2^{(v)}) = \frac{1}{k_1} \sum_{i=1}^{k_1} GSE_1(p_i^f) \tag{8}$$

$$SP_2(l_1^{(w)}, l_2^{(w)}) = \frac{1}{k_2} \sum_{j=1}^{k_2} GSE_2(p_j^l) \tag{9}$$

where k_1, k_2 are the number of pairs of features and labels that can be formed from the aggregation of $f_1^{(v)}$ and $f_2^{(v)}$, and $l_1^{(w)}$ and $l_2^{(w)}$, respectively. Similarly, p_i^f and p_j^l denote the ith and jth pairs of features and labels. In this way, we say that two pooled neurons $f_1^{(v)}$ and $f_2^{(v)}$, or $l_1^{(w)}$ and $l_2^{(w)}$ can be associated in the current layer if $SP_1 \geq \alpha_1$, or $SP_2 \geq \alpha_2$, respectively.

4 Simulations

In this section, we evaluate the ability of our proposal to estimate the association between problem variables (low-level features and labels), and between pooled neurons (high-level features and labels).

To perform the simulations, we use 10 multi-label datasets taken from the RUMDR repository [4]. In these problems (see Table 1), the number of instances ranges from 207 to 10,491, the number of features goes from 72 to 635, and the number of labels from 6 to 400. Also, the average maximal correlation of Pearson according to both features and labels is reported.

Table 1. Characterization of datasets used for simulations.

Dataset	Name	Instances	Features	Labels	Correlation-F	Correlation-L
D1	Emotions	593	72	6	0.62	0.39
D2	Scene	2,407	294	6	0.74	0.22
D3	Yeast	2,417	103	14	0.49	0.57
D4	Stackex-chemistry	6,961	540	175	0.18	0.13
D5	Stackex-chess	1,675	585	227	0.27	0.24
D6	Stackex-cooking	10,491	577	400	0.14	0.14
D7	Stackex-cs	9,270	635	274	0.18	0.18
D8	GnegativePseAAC	1,392	440	8	0.29	0.22
D9	GpositivePseAAC	519	440	4	0.33	0.34
D10	VirusPseAAC	207	440	6	0.40	0.22

The simulations aim at comparing our approach with the correlation-based method proposed in [1]. In order to make fair comparisons, we will use the same network architecture proposed by the authors. Similarly, as far as the pooling process is concerned, we set the maximum number of pooling layers to 5 for the features and 3 for the labels. The association thresholds α_1 and α_2 will range from 0.0 to 0.8. The operators used to aggregate two neurons (i.e. $\oplus$ and $\odot$) are the average in the feature pooling process, and maximum in the label pooling. In addition, the value of the similarity threshold parameter used in Eq. (1) is fixed to 0.85, although other values are also possible.

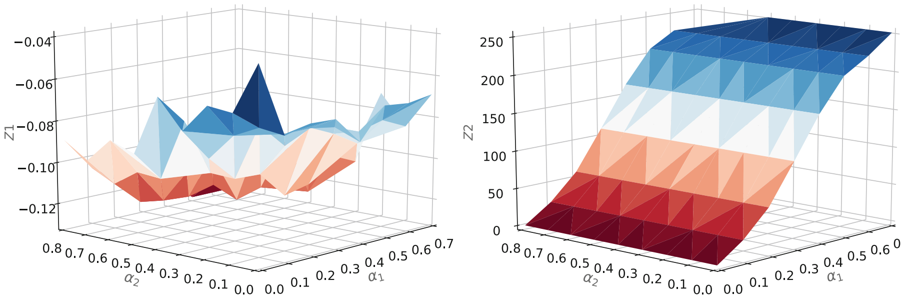
In all experiments conducted in this section, we use 80% of the dataset to build the model and 20% for testing purposes, while the reported results are averaged over 10 trials to draw consistent conclusions.

4.1 Results and Discussion

Table 2 displays the results of our measure in the model proposed by [1]. These tables report the number of high-level features, the reduction percentage those high-level features represent (%Red-Features), the number of high-level labels, the reduction percentage in the number of labels (%Red-Labels), the accuracy obtained when using only the high-level features and labels, and the accuracy loss with respect the model using all features and labels (i.e. neural network model without performing the pooling operations).

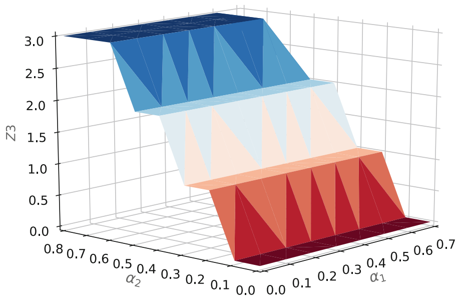
Table 2. Results achieved by the GSE_1 and GSE_2 measures.

Dataset	HL-Features	%Red-Features	HL-Labels	%Red-Labels	Accuracy	Loss
D1	3	95.83%	3	50%	0.515	−0.308
D2	10	96.60%	3	50%	0.771	−0.144
D3	4	96.12%	4	71.43%	0.765	−0.036
D4	17	96.85%	22	87.43%	0.988	0
D5	19	96.75%	29	87.22%	0.99	0
D6	19	96.71%	50	87.5%	0.995	0
D7	20	96.85%	35	87.23%	0.991	0
D8	14	96.82%	4	50%	0.864	−0.054
D9	14	96.82%	4	0%	0.646	−0.22
D10	14	96.82%	3	50%	0.736	−0.058



(a) accuracy

(b) high-level features



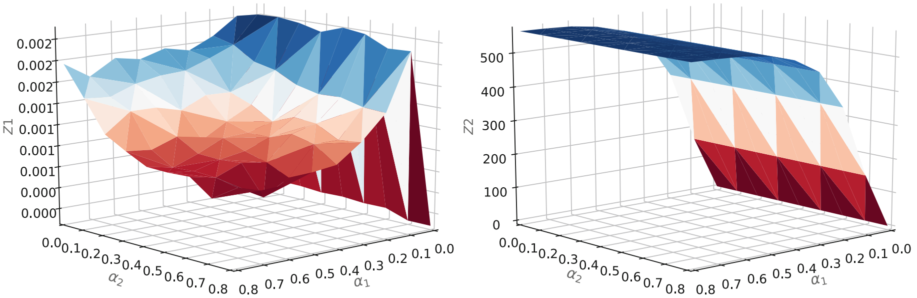
(c) high-level labels

Fig. 3. Average statistics over $G1$ using the GSE_1 and GSE_2 measures.

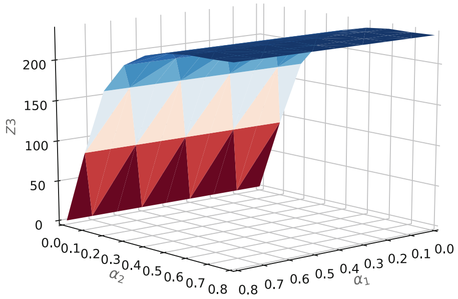
Our proposal (i.e. to compute the association between variables from GSE_1 and GSE_2) obtains a percentage of reduction in the features over 95%, and in the labels over 50% in most cases. On the other hand, the loss of accuracy is significant in those datasets that present a high correlation (e.g., $D1, D2, D9$), which means that this measure is not suitable in this datasets. It is remarkable the accuracy loss for $D1$, which is also the dataset with the lowest number of features in our study. However, the proposal reports a very small loss in those datasets having a lower correlation (e.g., $D4, D6, D7$).

Figures 3 and 4 show the comparison of our proposal against the one using Pearson's correlation (baseline). In these figures, we report the differences in accuracy between our method versus the baseline ($Z1$), the differences in the number of high-level features ($Z2$), and the number of high-level labels ($Z3$), when using different α_1 and α_2 values. Figure 3 summarizes the results for a first group of datasets $G1 = \{D1, D2, D3, D8, D9, D10\}$, while Fig. 4 shows the result of the second group $G2 = \{D4, D5, D6, D7\}$. The first group contains datasets having high correlation between their features and a middle correlation between their labels. Meanwhile, the second group consists of datasets having low correlation between their features and their labels.

For both groups, our proposal obtains a higher reduction rates when it comes to the number of high-level features and labels describing the problem.



(a) accuracy (b) high-level features



(c) high-level labels

Fig. 4. Average statistics over $G2$ using the GSE_1 and GSE_2 measures.

This difference is more significant when using high values for the association thresholds. In terms of accuracy, although the differences are not significant, the greatest differences are obtained in the $G1$ datasets, and that is, when high thresholds of association are used, while for the $G2$ datasets the opposite occurs. Our proposal achieves better results in datasets that have low correlation (i.e, those in $G2$), which confirms the hypothesis of our research.

It is worth mentioning that this model does not aim at increasing the prediction rates but to reduce of features and labels associated with the MLC problem. However, our results cry for the implementation of a convolutional operator to also increase networks' discriminatory power.

5 Concluding Remarks

In this paper, we have presented a method to quantify the association between problem variables (features and labels). This measure detects pairs of features (or labels) that are highly associable, and that will be used to perform an aggregation operation resulting in high-level features and labels. Unlike the pooling approach proposed in [1], our proposal does not require that either the features or labels have a certain degree of correlation with each other. Numerical results have shown that our proposal is able to significantly reduce the number of parameters in deep neural networks. When compared with the correlation-based variant, our model reported higher reduction values in datasets having low correlation values among their features and labels. As a result, we obtained simpler models without significantly affecting networks' discriminatory power.

References

1. Bello, M., Nápoles, G., Sánchez, R., Vanhoof, K., Bello, R.: Deep neural network to extract high-level features and labels in multi-label classification problems. Neurocomputing (Submitted)
2. Bello, R., Falcon, R., Pedrycz, W., Kacprzyk, J.: Granular Computing: At the Junction of Rough Sets and Fuzzy Sets. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-76973-6
3. Bonikowski, Z., Bryniarski, E., Wybraniec-Skardowska, U.: Extensions and intentions in the rough set theory. Inf. Sci. **107**(1–4), 149–167 (1998)
4. Charte, F., Charte, D., Rivera, A., del Jesus, M.J., Herrera, F.: R Ultimate multilabel dataset repository. In: Martínez-Álvarez, F., Troncoso, A., Quintián, H., Corchado, E. (eds.) HAIS 2016. LNCS (LNAI), vol. 9648, pp. 487–499. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-32034-2_41
5. Choi, K., Fazekas, G., Sandler, M.: Automatic tagging using deep convolutional neural networks. arXiv preprint arXiv:1606.00298 (2016)
6. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. The MIT Press, Cambridge (2016)
7. Herbert, J.P., Yao, J.: A granular computing framework for self-organizing maps. Neurocomputing **72**(13–15), 2865–2872 (2009)

8. Herrera, F., Charte, F., Rivera, A.J., del Jesus, M.J.: Multilabel Classification. Problem Analysis, Metrics and Techniques. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-41111-8
9. Hinton, G.E., Salakhutdinov, R.R.: Reducing the dimensionality of data with neural networks. Science **313**(5786), 504–507 (2006)
10. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature **521**(7553), 436–444 (2015)
11. Lee, C.Y., Gallagher, P.W., Tu, Z.: Generalizing pooling functions in convolutional neural networks: mixed, gated, and tree. In: Artificial Intelligence and Statistics, pp. 464–472 (2016)
12. Lin, M., Chen, Q., Yan, S.: Network in network. arXiv preprint arXiv:1312.4400 (2013)
13. Pawlak, Z.: Rough sets. Int. J. Comput. Inf. Sci. **11**(5), 341–356 (1982)
14. Pedrycz, W., Skowron, A., Kreinovich, V.: Handbook of Granular Computing. Wiley, New York (2008)
15. Rios, A., Kavuluru, R.: Convolutional neural networks for biomedical text classification: application in indexing biomedical articles. In: Proceedings of the 6th ACM Conference on Bioinformatics, Computational Biology and Health Informatics, pp. 258–267. ACM (2015)
16. Slowinski, R., Vanderpooten, D.: A generalized definition of rough approximations based on similarity. IEEE Trans. Knowl. Data Eng. **12**(2), 331–336 (2000)
17. Wei, Y., et al.: HCP: a flexible CNN framework for multi-label image classification. IEEE Trans. Pattern Anal. Mach. Intell. **38**(9), 1901–1907 (2015)
18. Yao, J.T., Yao, Y.Y.: Induction of classification rules by granular computing. In: Alpigini, J.J., Peters, J.F., Skowron, A., Zhong, N. (eds.) RSCTC 2002. LNCS (LNAI), vol. 2475, pp. 331–338. Springer, Heidelberg (2002). https://doi.org/10.1007/3-540-45813-1_43
19. Yao, Y.: Probabilistic approaches to rough sets. Expert Syst. **20**(5), 287–297 (2003)
20. Zhang, M.L., Zhou, Z.H.: A review on multi-label learning algorithms. IEEE Trans. Knowl. Data Eng. **26**(8), 1819–1837 (2013)
21. Zhu, J., Liao, S., Lei, Z., Li, S.Z.: Multi-label convolutional neural network based pedestrian attribute classification. Image Vis. Comput. **58**, 224–229 (2017)