WILEY

Community detection-based deep neural network architectures: A fully automated framework based on Likert-scale data

Francisco Javier Pérez-Benito[1,2] | **Juan Miguel García-Gómez**[1] | **Esperanza Navarro-Pardo**[3] | **J. Alberto Conejero**[2]

[1]Biomedical Data Science Lab (BDSLab), Universitat Politècnica de València, València, Spain

[2]Instituto Universitario de Matemática Pura y Aplicada (IUMPA), Universitat Politècnica de València, València, Spain

[3]Departamento de Psicología y de la Educación, Universidad de València, Valencia, Spain

Correspondence
E. Navarro-Pardo, Departamento de Psicología y de la Educación, Universidad de València, Avenida Blasco Ibáñez, 21, 46010 Valencia, Spain.
Email: Esperanza.Navarro@uv.es

Communicated by: J. R. Torregrosa

Deep neural networks (DNNs) have emerged as a state-of-the-art tool in very different research fields due to its adaptive power to the decision space since they do not presuppose any linear relationship between data. Some of the main disadvantages of these trending models are that the choice of the network underlying architecture profoundly influences the performance of the model and that the architecture design requires prior knowledge of the field of study. The use of questionnaires is hugely extended in social/behavioral sciences. The main contribution of this work is to automate the process of a DNN architecture design by using an agglomerative hierarchical algorithm that mimics the conceptual structure of such surveys. Although the train had regression purposes, it is easily convertible to deal with classification tasks. Our proposed methodology will be tested with a database containing socio-demographic data and the responses to five psychometric Likert scales related to the prediction of happiness. These scales have been already used to design a DNN architecture based on the subdimension of the scales. We show that our new network configurations outperform the previous existing DNN architectures.

KEYWORDS

automatic architecture, community detection, community-detection deep neural network (CD-DNN), deep learning, happiness, network science, psychometric scales, regression

MSC CLASSIFICATION

68T07; 90B10

1 | INTRODUCTION

The admissible level of complexity of neural network architectures was constrained until the computational capacity provided by GPUs was unlocked. This new capacity has permitted deep neural networks (DNNs) to become one of the best tools for classification and regression tasks using data from different science fields.[1-3] Under this new paradigm, DNNs can learn from data represented into subsequent levels of abstraction, which permits to increase its predictive performance.[2]

A DNN is composed of several hidden *layers* which, in turn, are divided into neurons. A neuron is composed of a matrix of weights that produces a value after applying an activation function to the entries of the neuron. In general, the design of the network layers structure, the activation function, and the use of bias in the model are human choices that are usually

based on the researcher's knowledge in the problem under consideration. However, sometimes the best architecture that fits a problem can be obtained after a trial-and-error process.

Several attempts of automatic design of DNN architectures have been already considered[4-9] with successful results. Their methods are mainly based on genetic algorithms, hyper-parameter optimization, and reinforcement learning. In all these cases, lots of computational resources are required because of the exponential increment of steps at the training stage.

The study of the network representations of complex processes arisen in physics, biology, and sociology can be represented in terms of interconnected nodes of graphs combining organization and randomness.[10,11] This field is known as *Network Science*. Some well-known examples include the world-wide-web,[12] citation networks,[13] the interactome,[14], diseasome,[15] or P2P networks.[16] For further information, we refer the reader to Barabasi's book.[17] We will refer to these networks as graphs to avoid confusion with neural networks.

Community detection algorithms permit to decompose graphs into highly interconnected subunits.[18] Unlike partitioning, in community detection procedures, the number and size of communities cannot be determined in advance. There is a wide variety of community detection algorithms, covering heuristic,[19] divisive,[20] and agglomerative algorithms.[21,22]

The current research on graph community detection has many branches. On one side, although almost all community detection algorithms require the global information of the graph, a local approach based on fuzzy relations has been proposed.[23] Furthermore, other approaches based on the trend paradigm of deep learning have emerged, this is known as graph neural networks and has many applications, in particular, community detection. From a Deep Graph Kernels framework that can learn latent representations inside a graph [24] or a variant of a convolutional network that encodes the local graph structure and the features of nodes.[25] Up to adversarial networks that make perturbations on the graph to generate constrained ones which are then classified into communities[26] or the extraction of temporal features using local long short-term memory networks to be used to learn spatio-temporal patterns to infer the communities.[27] Finally, the graph attention networks provide a methodology based on convolutional networks, without depending or knowing the graph structure upfront.[28]

In this work, we propose a methodology to design DNN architectures for conceptual-structured data of the answers to items of Likert-type scales. It will rely on the use of community detection algorithms for determining the aggregation of items by similarity in the answers, that is, if some items are answered in a similar way (positive or negative), we assume that they will work fine if we consider them in the same layer. To define a DNN with several layers, we will apply community detection algorithms at different resolutions. The graph modularity will be a figure of merit of the resulting distribution of the graph into communities. Modularity optimization algorithms belong to the class of agglomerative methods that provide a hierarchical clustering fitting the problem nature. Thereby, methods based on modularity optimization will be our starting point.

The rest of the paper is organized as follows: First, the proposed methodology is presented in Section 2. To test the performance of this new method, we apply it to a psychological dataset that has been recently used for predicting happiness.[29] The new proposed DNN architectures enhance the prediction given by the DNN based on the inner structure of psychological scales shown in Pérez-Benito et al,[29] as it is drawn in Section 4. Discussion of the results, including the limitations of the methodology and a proposal for future research lines, are shown in Section 5. Finally, some brief conclusions are presented in Section 6.

2 | PROPOSED METHODOLOGY

Our study aims to present a framework to fully automate the development of DNN architectures for solving supervised regression-type problems based on conceptual-structured data. Our approach that we will call community detection-based deep neural networks (CD-DNN) consists of the following:

- Construction of a graph that quantifies the similarity in the answers to items from different scales.
- Application of a community detection method based on modularity optimization at different resolutions.
- Proposal of a DNN architecture automatically inferred from the community hierarchy of the scales items.
- Implementation, training, and validation of the new proposed architecture.

Let us develop each one of these steps.

2.1 | Construction of a similarity graph from the dataset

Likert scales, or summated rating scales,[30] are one of the most commonly used research tools in behavioral sciences. They consist of a list of items that present a finite ordered list of possible answers. The subject who is evaluated answers showing his agreement or disagreement degree with the statement of the item.[31] A value is assigned to each answer, and later all of them are summed. Depending on that value, the individual is classified into a group.

Let us suppose that each subject answers to n Likert-type questionnaires, denoted by S_i with $1 \leq i \leq n$, where each questionnaire S_i is composed of n_i items. Let us sequentially rename all the items as $\{v_j : 1 \leq i \leq m\}$, where $m = \sum_{i=1}^{n} m_i$.

Before defining our similarity graph, we recall that a *weighted graph* is given by 3-tuple $G = (V, E, w)$, where V is the set of nodes, $E = \{(v_i, v_j) : v_i, v_j \in V\}$ is the set of edges, and $w : E \rightarrow \mathbb{R}_+$ is a function that assign weights to the edges.

Let us consider a dataset in which we have the answers of s_0 subjects to all the scales. Our similarity graph will be a weighted graph $G = (V, E, w)$ defined as follows:

1. We consider $V = \{v_j : 1 \leq i \leq m\}$ as the set of nodes.
2. A pair of nodes is connected by an edge $(v_i, v_j) \in E$ if there exists at least two people answering in the same sense (agreeing or disagreeing) to both items.
3. The weight associated to the edge (v_i, v_j), denoted by w_{ij}, will be the number of subjects answering in the same sense both items v_i and v_j.

2.2 | Communities detection and architecture construction

Now, we will apply to the similarity graph G community detection algorithms at different resolutions to automatically infer the conceptual hierarchy that will provide us the conceptual hierarchy of the DNN to be trained.

Let us suppose that we have split the graph G into k different communities. The notion of *modularity* of a weighted graph G, denoted by $Q(G)$, was introduced by Newman and Girvan[32] as follows:

$$Q(G) = \frac{1}{2W} \sum_{i,j=1}^{m} \left(w_{ij} - \frac{W_i W_j}{2W} \right) \delta(C_i, C_j), \tag{1}$$

where w_{ij} is the weight between nodes i and j, $W_i = \sum_{j=1}^{m} w_{ij}$ is the sum of the weights associated to edges adjacent to v_i, $C_i \in \{1, \ldots, k\}$ is the identifier of the community to which the node v_i belongs to, δ is the Kronecker delta function such that $\delta(u, v)$ is equal to 1 if $u = v$ and 0 otherwise, and $W = \sum_{i,j=1}^{n} w_{ij}$. The value of Q is defined between -1 and 1, and it measures the density of edges inside communities compared with the density of edges between communities.

According to Clauset et al,[33] modularity is a property of a graph and a specific proposed division of that graph into communities. The modularity optimization algorithm[32] belongs to the set of agglomerative hierarchical clustering methods.[20,34] It iterates until a nonnull value of modularity is reached. As a reference, 0 indicates a random division and 1 the best possible division into communities.

It has been demonstrated that a value above 0.3 is a good indicator of significant community structure in a network.[33] One of the most widespread algorithms for modularity optimization is the Louvain algorithm,[35] and it was our choice due to its balance between community detection capability and computation time. Louvain[35] is a 2-phase iterative algorithm that optimizes—in terms of computational time—those proposed by Newman et al[32] and Clauset et al.[33] The aim of the algorithm is to find the graph communities which outperform a predefined value of increment of modularity, namely resolution (ΔQ, see equation (2)). This value may vary between 0 and 1. Let us briefly outline how it works: Starting with a weighted graph $G = (V, E, w)$, the first step is to assign a different community to each node $v \in V$. The set of neighbors of a node v_i can be defined as $N(v_i) = \{v_j : (v_i, v_j) \in E\}$.

Let us now consider the gain of modularity (resolution) by moving a vertex v_i into a community C proposed by Blondel et al,[35] which in terms of graph can be denoted by the following:

$$\Delta Q(G, v_i, C) = \left[\frac{W_C + W_{iC}}{2W} - \left(\frac{W_{\hat{C}} + W_i}{2W} \right)^2 \right] - \left[\frac{W_C}{2W} - \left(\frac{W_{\hat{C}}}{2W} \right)^2 - \left(\frac{W_i}{2W} \right)^2 \right], \tag{2}$$

where W, W_i are already defined, and

$$W_C = \sum_{v_j, v_k \in C} w_{jk}, \quad W_{iC} = \sum_{v_j \in C} w_{ij}, \quad \text{and } W_{\hat{C}} = \sum_{v_j \in C, v_k \notin C} w_{jk}. \tag{3}$$

First, for each node v_i e, compute the gains of modularity $\Delta Q(v_i)$, which is assessed by removing v_i from its community and placing it in the community of each one of its neighbours $v_j \in N(v_i)$. Then, we will place v_i in the community for which the gain is positive and maximum. If all the gains are negative, v_i will stay in its original community.

Second, we construct a new graph $\widetilde{G}$ whose nodes represent the communities found in the previous step. Two nodes of $\widetilde{G}$ will be connected as long as there was a node in each community that was already connected in G. The weights between two nodes of $\widetilde{G}$ will be given by the sum of the weights of edges between the corresponding two communities. This second step has been shown to preserve modularity of G for $\widetilde{G}$.[36]

Finally, the modularity $Q(\widetilde{G})$ is assessed, and the process is repeated until the resolution reached.

3 | PROPOSAL OF DNN ARCHITECTURE

The next step is to define the DNN architecture. We propose optimizing the modularity at different resolutions ranging between 0.6 and 1 to ensure deviations from randomness.

By optimizing the modularity at different resolutions ranging between 0.6 and 1, we were capable of inferring a hierarchy to develop the DNN architecture automatically. Low-resolution levels produce a higher number of communities while the highest resolution, 1, produces the smallest possible number of them.

To create a DNN architecture, we first fix the number of hierarchical levels h_0 in which the layers will be included. We express the range of resolutions, $[0.6, 1]$, in terms of the number of levels according to the next formula, that assigns 0.6 to the first level and 1 to the last one.

$$r_h = 0.6 + \frac{h-1}{h_0 - 1} 0.4, \text{ with } 1 \leq h \leq h_0. \tag{4}$$

The set of layers that belong to the hierarchical level h will be denoted as H_h. Note that the number of layers in each level is, by definition, the number of communities at resolution r_h.

Let $H_1 L_j$ be a layer of the first hierarchical level with $1 \leq j \leq l_1$ and $q(s)$ be one of the inputs of the model with $1 \leq s \leq s_0$, that is, an answer to the question q by one of the subjects. Then, $H_1 L_j$ receives each $q(s)$ as input if, and only if, q belongs to community j at resolution r_1.

Now, let $H_i L_j$ be a layer of the hierarchical level i with $1 \leq j \leq l_i$ and $H_{i+1} L_{j'}$ with $1 \leq j' \leq l_{i+1}$ be a layer of the following level. Then, the outputs of $H_i L_j$ will act as entries for the layer $H_{i+1} L_{j'}$ if, and only if, there exists at least one item in the scale whose community at resolution r_i is j and whose community at resolution r_{i+1} is j'.

Finally, the last hierarchical level only contains one fully connected layer with one neuron, which provides a one-dimensional result which is that optimized using the gold standard in a supervised learning process.

Now, we have explored the performance of this architecture with different subconfigurations in every layer. We have considered the cases of one neuron per layer, and as many neurons as the number of incoming inputs, that is, for the first hierarchical level, it corresponds to the number of incoming items, and for the rest of the levels, it corresponds to the number of incoming layers from the previous one. We have also tested the outcome with and without bias in the hidden layers in both cases.

3.1 | CD-DNN: Training, validation, and test stages

Common practices in machine learning suggest to split the corpus into *Training*, *Test*, and *Validation* sets to monitor the training process and to avoid overfitting problems. In this sense, we split the corpus by taking the 70% as the training set and the remaining 30% as the test set. Half of the test set was extracted as a validation set to monitor the performance between training and validation at each epoch of the training process. So, the training set was composed of the data provided by 578 individuals to the questionnaires. The remaining data was split into the text set (123) and the validation one (122).

We define a heuristic stop rule to avoid overfitting. This rule is taken as follows: *if during five training epochs the training error was reduced and the validation error was incremented then the training process stops*. Sigmoid function was selected as activation function.

The Adam optimizer[37] was chosen as the training algorithm and the performance test the mean-squared error (MSE) with the maximum training epoch set to 100. Finally, the results were assessed taking into account the whole test set (the 30% of the original corpus).

3.2 | Algorithm

The proposed methodology may be summarized in four key points as presented in Section 2. Following the notation on this section, each participant answered m questions.

First, a way to quantify how two questions related was used to define a graph G that represents the similarity between the m questions. Once the graph is built, the number of hierarchical levels needs to be fixed (R). This number lets us choose R thresholds from the resolutions space and apply R times Blondel's community detection algorithm.

For each resolution, each community defines a layer of the final DNN, and two layers of different hierarchical levels are connected if and only if the communities that they represent contain at least one question. The layers of the first hierarchical level receive as input the questions that belong to the communities it represents. By adding the last layer (fully connected) that receives as inputs the outputs of the layers of the last hierarchical level, the DNN is ready to be trained.

The complexity of the graph building is $\mathcal{O}(m^2)$. Due to the definition of the relationship between questions, every pair of nodes has an edge between them. This implies that the complexity of the community detection for each resolution is $\mathcal{O}(m^2)$ but it could be optimized until $\mathcal{O}(m \log(k))$ where k is the average degree.[38] Then, the complexity of the construction of the model is $\mathcal{O}(R \cdot m^2)$ and could become $\mathcal{O}(R \cdot m \log(k))$, where R is the number of hierarchical levels we want to use.

The stages of the procedure may be found in the Algorithm 1. The experimentation for this work was made using *python 3.7* language and the extended libraries *Networkx*[39] and *Tensorflow*.[40]

Algorithm 1 The algorithm of the CD-DNN model construction and training.

---

   procedure CD-DNN(d, R)   ▷ Building the CD-DNN for the database d with R hierarchical levels
      $d\_train, d\_test = split\_corpus\_train\_and\_test()$
      $d\_validation = random\_half(d\_test)$
      for $0 \leq row < rows(d\_train)$ **do**
         for $row < column < rows(d\_train)$ **do**
            $G \leftarrow obtain\_weight(row, column)$ ▷ Definition of the graph edges.
         end for
      end for
      $m = empty\_model()$
      while $i < R$ **do**
         $r \leftarrow 0.6 + \frac{i-1}{R-1}0.4$
         $C \leftarrow comm\_detection(G, r)$ ▷ $comm\_detection(G, r)$ obtain the Blondel's communities of G at resolution r.
         $inf \leftarrow create\_layers(C)$ ▷ Creates layers for current hierarchical level and its associations with previous level.
         $m.update(inf)$
      end while
      $create\_last\_layer()$ ▷ Creates the last layer. It is dense.
      while $j < 100$ or $not(stop\_criterion)$ **do**
         $model.train(d\_train)$
         $stop\_criterion \leftarrow is\_converging(d\_validation)$
      end while
      return m ▷ m is the trained model
   end procedure

---

4 | EXPERIMENTAL RESULTS

We will compare the performance of our new automatically designed architectures with the ones in Pérez-Benito et al[29] that were applied to the dataset described below.

4.1 | Dataset

A cross-sectional survey targeting the noninstitutionalized adult population residing in Spain was completed by 823 cases. The total of 111 elements of the survey is grouped by socio-demographic data and by five psychometric scales. The psychometric scales measure latent variables describing psychological factors through empirical behavioral indicators. Socio-demographic data covers an identifier of the person who recorded the survey and the age, gender, marital status, and level of education for each individual. The psychometric scales enclosed in the study were as follows:

1. *Short-Depression-Happiness Scale* (SDHS)[41] is a 4-point Likert-scale with a total of six items. The total score is a measurement of the happiness degree of the patient (ranging from 0 [*depression*] to 18 [*happiness*]) and was the gold standard of the model.
2. *Brief COPE Inventory*[42] is a 4-point Likert-scale with a total of 28 items regrouped in 14 subscales. The total scale measures coping strategies that are different mental mechanisms regarding manage demands and conflicts and to regulate emotional responses and stress. The 14 subscales represent self-distraction, active coping, denial, substance abuse, use of emotional support, use of instrumental support, behavioral disengagement, venting, positive reframing, planning, humor, religion, and self-blame.
3. *Eysenck Personality Questionnaire Revised-Abbreviated* (EPQR-A)[43] consists of four subscales of six dichotomous items, each that assess neuroticism, extraversion, psychoticism, and sincerity. The total scale measures the personality.
4. *General Health Questionnaire of 28 items* (GHQ-28)[44] is a 5-point Likert-scale assessing the emotional distress which is defined as a feeling that a person or situation is triggering psychological suffering and could be expressed in different degrees not only cognitive or verbally but through mental or physical symptoms. Emotional distress is composed of four subscales to evaluate somatic symptoms, anxiety/insomnia, social dysfunction, and severe depression.
5. *Medical Outcomes Study Social Support Survey* (MOS-SSS)[45] measures social support. It is composed of 20 items consisting of a first question asking for the number of close people that the person has, plus a total of 19 5-point Likert-scale items that covers four functional support subscales: emotional/informational, tangible, affectionate, and positive social interaction.

4.2 | Previous architecture

We will compare the proposed architecture with the one based on the conceptual two-level structure of the scales of the (DS-DNN). We recall that each scale is composed of items. These items are first grouped to measure different psychological subfactors or subscales. At the same time, psychological subfactors are considered altogether to quantify a more general psychological factor. So that, we can consider that the conceptual structure of each scale is composed of two hierarchical levels: the first one to describe psychological subscales and the second one to describe the global psychological factor measured by the scale.

We can transfer this structure into a DNN as follows: In the first level of hidden layers, we represent the subscales. The items of each subscale will be their inputs. For the second hierarchical level, the scales as the hidden layers. Here, the outputs of the subscales are the inputs of the corresponding scale layers. We also add in this second level a new layer where the socio-demographic data is considered. Its output is directly included as input in the last layer. This DNN architecture is the same as the one exposed in Section 3.

Finally, it is worth to mention that for facilitating the validation of our methodology and to obtain comparable results, the exception of the socio-demographic layer is also considered in the construction of the proposed automatic architecture.

4.3 | Experimentation

For testing our method with the results obtained with the DNN described in Section 4.2, we have automatically generated DNN with two hierarchical levels including the socio-demographic layer in the second one. According to equation (4), the resolutions for community detection were 0.6 and 1. The detected communities at these resolutions can be found in Figure 1.

The inferred architecture is summarized in Figure 2, with a total of 22 layers in the first hierarchical level and four layers in the second one, apart from the layer related to socio-demographic information. All the CD-DNN models were trained using the Adam optimizer[37] with random initialization and a learning rate of 0.001 to minimize the MSE. All the layers used the sigmoid function as activation and the last hidden layer is a dense one which receives as inputs the outputs of all of the layers of the last hierarchical level, in this case, all the outputs of the layers from the second hierarchical level.
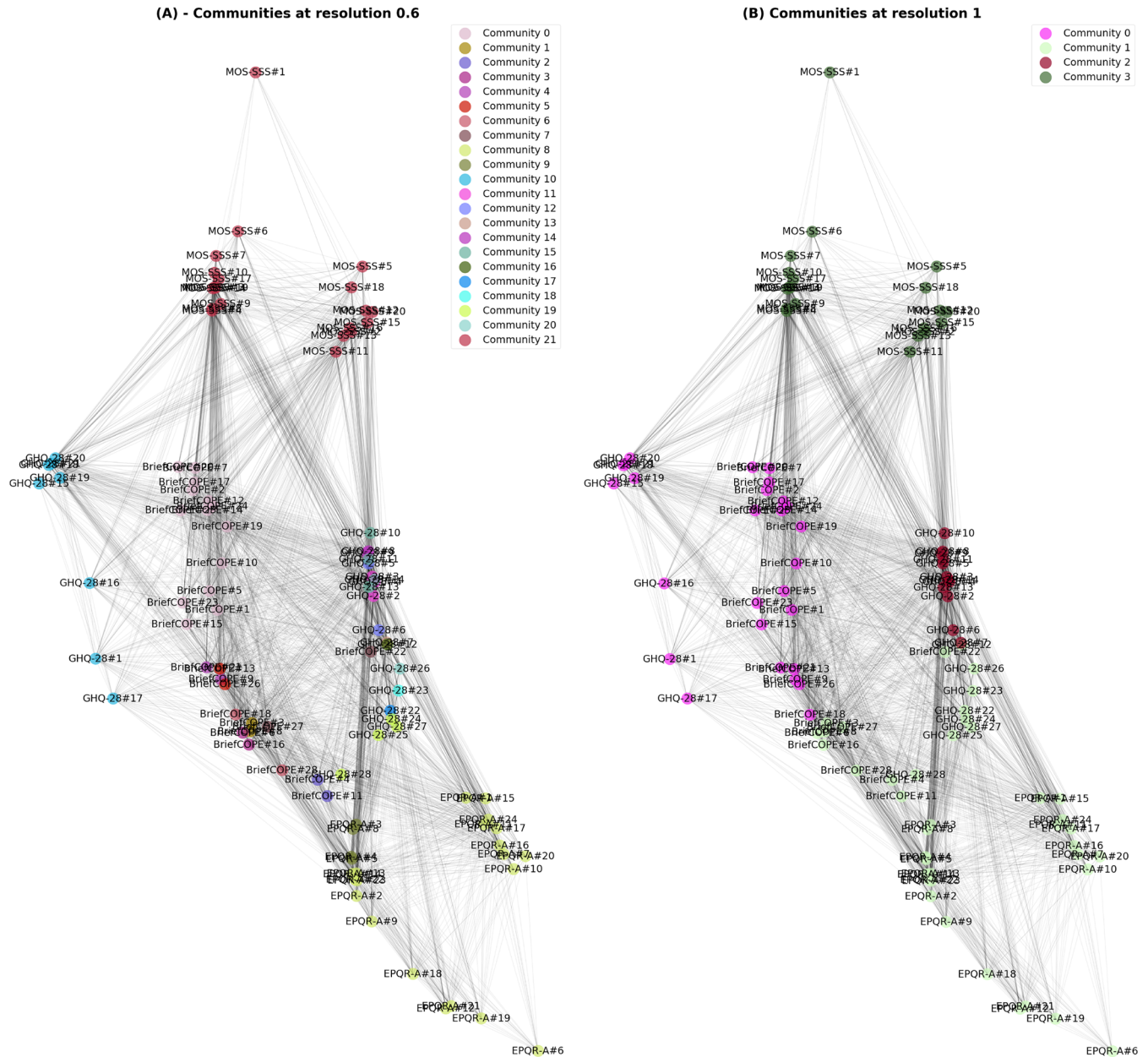
FIGURE 1 Detected communities after applying the Louvain algorithm at different resolutions. Each node represents an item of psychometric scales. Each node is depicted by the acronym of the scale which belongs to and the number of the item. The stronger the relationships are, the thicker the line between nodes is depicted. In A, we show communities at resolution 0.6, and in B, we present communities at resolution 1 [Colour figure can be viewed at wileyonlinelibrary.com]

Results show that the automatic approach, CD-DNN, outperforms the results obtained by the DNN created using the preconceived survey structure (DS-DNN). The comparison between the results obtained using both methods can be found in Figure 3. We highlight that the new proposed methodology tends to be more accurate to predict the less frequent degrees of happiness. This suggests that this methodology has a higher power of abstraction. It is also worth to mention that both approaches reach the best performance using the same layers configuration, namely multineuron layers without using bias.

Finally, a short description of each trained model architecture with some training features are drawn in Table 1. The elapsed time per training epoch barely suffers when neurons are added, meanwhile the needed number of epochs, until the stop criterion is reached, significantly decreases with multineuron configurations. Multineuron-biased network (CD-DNN_N_B) gives a training error of $9 \cdot 10^{-3}$ while the validation error is $1.8 \cdot 10^{-2}$ when the training process ends.
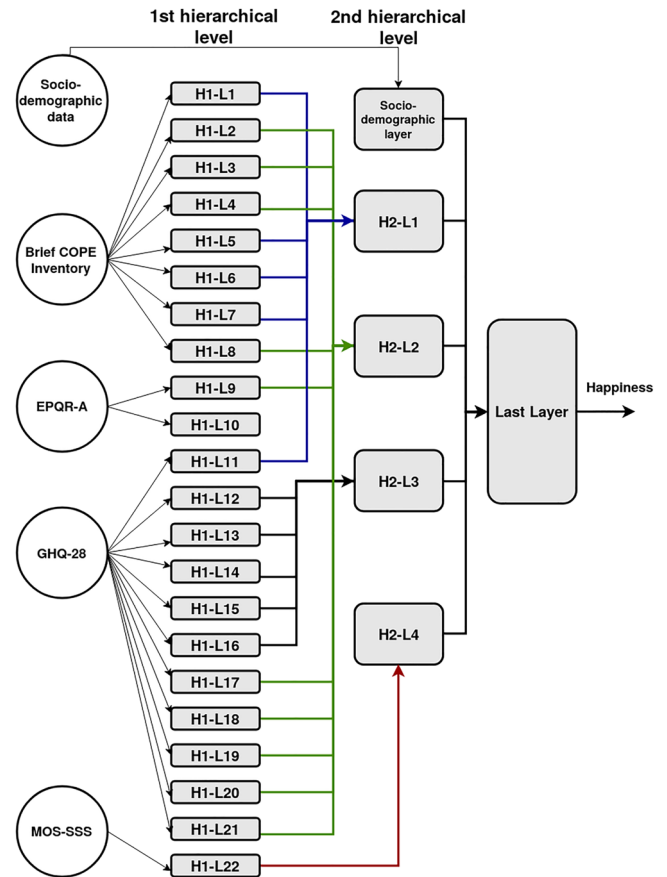
FIGURE 2 Community detection-based deep neural network (CD-DNN) architecture. This architecture is automatically inferred by using the similarity in the responses to the scale items, observed from the responses gathered in the dataset [Colour figure can be viewed at wileyonlinelibrary.com]

TABLE 1 The four DNN configurations applied to test the performance of our CD-DNN

Net	Bias (T/F)[a]	Neurons per layer	Epochs[b]	Time per epoch[c]
CDDNN_1	F	1	42	1.32
CDDNN_1_B	T	1	26	1.32
CDDNN_N	F	multi	19	1.34
CDDNN_N_B	T	multi	10	1.35

Abbreviations: CD-DNN, community detection-based deep neural network; DNN, deep neural network.

[a]Use of bias in layers. T, True; F, False.

[b]Number of training epochs until the stop criterion is reached.

[c]Mean elapsed time during one training and validation epoch (in seconds).

This undesirable behavior is only observed in this configuration. It may imply that this network "learns" so quickly that the stop criterion is not enough to avoid overfitting problems.[46]

Although out of the scope of this work, it is worth to mention that we have noticed latent relationships between the psychometric scales. This can be observed at the inputs of the H_2L_1 and H_2L_2. We observe that the corresponding communities at resolution 1 have grouped altogether layers of different scales in contrast to what can be seen at resolution 0.6 when there were no intersections between scales. It is also highlighted that all the items that belong to the MOS-SSS survey formed one unique community at both resolutions. This suggests that the subscales of the MOS-SSS had similar responses in our dataset and the use of two hierarchical levels may be unnecessary for these items. Finally, 12 of 28 GHQ-28 items were grouped at resolution 1, while these items were grouped into six communities at resolution 0.6.
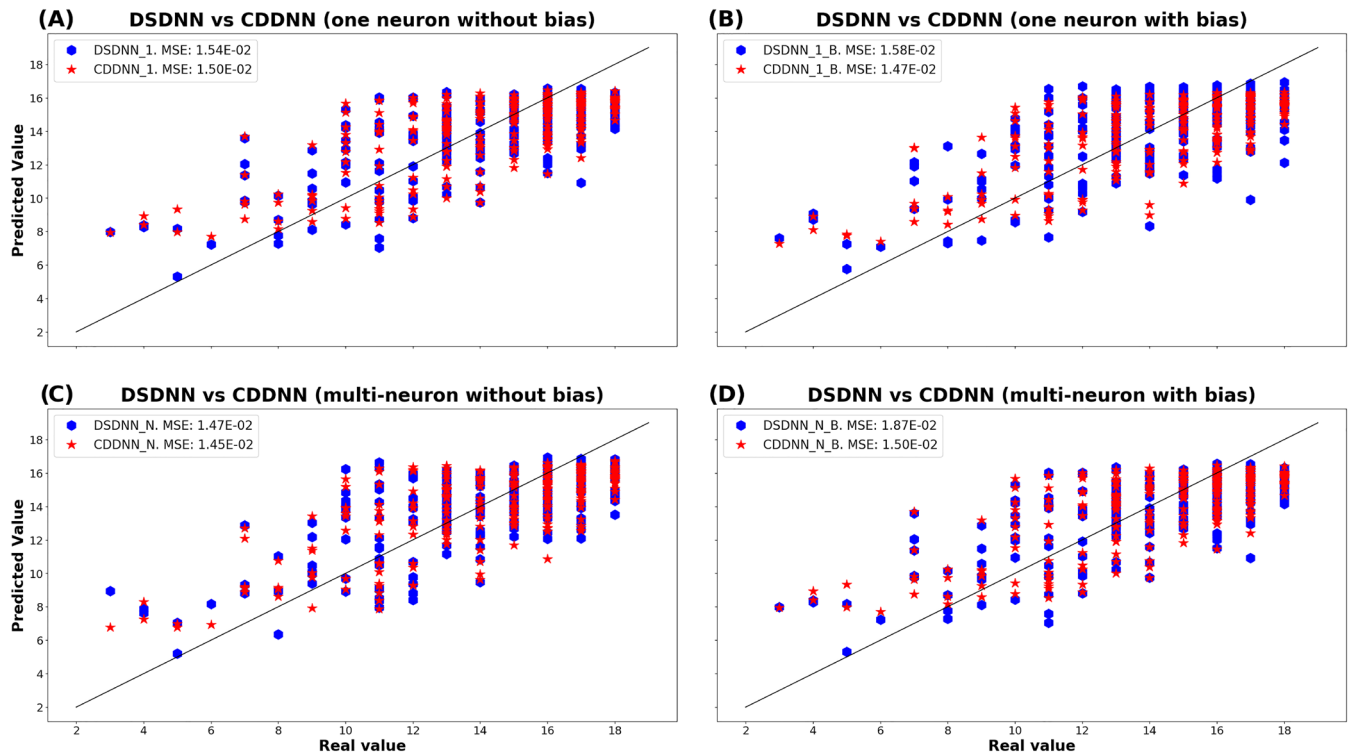
FIGURE 3 Comparison performance results for community detection-based deep neural network (DS-DNN) and CD-DNN of the tested network configurations. (A) presents the comparison of the results of "one-neuron without adding bias" configuration, (B) shows the results considering the case of "one-neuron with bias" configuration, (C) compares the "multineuron without bias" results, and finally, (D) displays the results for "multineuron with bias" configuration [Colour figure can be viewed at wileyonlinelibrary.com]

5 | DISCUSSION

5.1 | Impact and contributions

The aim of the present study has been the definition of a new methodology to automatically construct the architecture of a deep learning-based model to be used with regression purposes. We focused our efforts on the case Likert-scale datasets. The nature of these measurement artifacts provides a simple way to define similarity relationships between items of different scales.

By definition, the proposed methodology is a contribution in the sense that it automates the construction of the model architecture. Besides, the results showed in Section 4 demonstrate that the CD-DNN provides better experimental results on real data when compared with a data-structure architecture drawn using the preconceived subscale structure of the scales.

Congruent with our previous work, CD-DNN with unbiased multiple-neuron layers gives the best results. This may indicate that multiple-neuron configurations provide enough adaptability to changes within hierarchical levels, and the learning rate is controllable enough to avoid falling into overfitting problems.

Although the interpretation of the results shown in this paper is out its scope, the presented approach also opens a debate about the possibility of finding latent relationships between items of different psychological scales. Furthermore, the architecture based on these relationships increases the predictive ability of DNN models when compared with those whose architecture is based on the structure of the scales.

Finally, it would be easy to generalize to classification problems by only modifying the behavior of the last layer, which is independent of the community detection based architecture, and the way of codifying the ground truth for supervised learning.

5.2 | Limitations and future work

Problems based on predictions (either in regression or classification problems) from Likert scales have a very particular field of application. In this sense, the most critical limitation, which is aligned with the most important line for future work, is the capability of extending the automatic construction of the architecture for the DNN for other types of datasets.

If we distinguish between the graph construction and the implementation of the DNN, our efforts should focus on generalizing the graph definition. That is because the DNN is automatically generated when the communities are known, and the communities are automatically detected if the graph is known. So it is enough to find a way to generalize the construction of the graph or to find a set of ways that covered a wide range of data. The use of deep learning in the search of communities could lead to a community detection generalization independent of the data type. It would be necessary to explore how to mimic the hierarchy for these questionnaires. Although the methodology proposed in the present paper uses agglomerative hierarchical algorithms to match the conceptual structure of the questionnaires, it would also be interesting to try other community detection algorithms where an element could belong to two communities.

The experimental results presented in this work are good for the dataset considered, but the validation of the proposed methodology, considering larger datasets, would also be desirable.

6 | CONCLUSION

Deep learning-based models are a trending paradigm in research fields using data science techniques. This technology is increasingly being used for all kinds of tasks and is outperforming results obtained with previous approaches. The main problem of using models based on deep learning is to know the most accurate way of designing them, and many times this process is carried out as trial and error until a quality criterion is reached.

Several studies are based on the analysis of structured data, as is the case with Likert-scales-based studies, for which we propose a framework to construct a deep learning model with regression purposes automatically. The methodology consists of the definition theoretic graph quantifying the relationships between different items of the scales, applying a community detection algorithm to infer the architecture for the DNN (CD-DNN) and the training of the model.

The first experiments using real data have demonstrated better performance when compared with a previous work in which the architecture was drawn by using the conceptual structure of the Likert scales. The natural next step is to get the graph definition abstracted to apply the same methodology to other types of data.

Future steps must explore other approaches for community detection to make the model independent of the data type. Algorithms based on deep learning could deal with community detection generalization.

CONFLICTS OF INTEREST

This work does not have any conflicts of interest.

ORCID

Francisco Javier Pérez-Benito https://orcid.org/0000-0002-6290-5644
Esperanza Navarro-Pardo https://orcid.org/0000-0002-9355-2909
J. Alberto Conejero https://orcid.org/0000-0003-3681-7533

REFERENCES

1. Goodfellow I, Bengio Y, Courville A. *Deep learning*, Vol. 1: MIT press Cambridge; 2016.
2. LeCun Y, Bengio Y, Hinton G. Deep learning. *Nat*. 2015;521(7553):436.

3. Antonov V, Tarkhov D, Vasilyev A. Unified approach to constructing the neural network models of real objects. part 1. *Math Method Appl Sci*. 2018;41(18):9244-9251.

4. Arifovic J, Gencay R. Using genetic algorithms to select architecture of a feedforward artificial neural network. *Physica A: Statistical Mechanics and its Applications*. 2001;289(3-4):574-594.

5. Islam B-U, Baharudin Z, Raza M-Q, Nallagownden P. Optimization of neural network architecture using genetic algorithm for load forecasting. In: 5th International Conference on Intelligent and Advanced Systems (ICIAS), 2014; 2014:1-6.

6. Koutník J, Schmidhuber J, Gomez F. Evolving deep unsupervised convolutional networks for vision-based reinforcement learning. In: Proceedings of the 2014 annual conference on genetic and evolutionary computation ACM; 2014:541-548.

7. Loshchilov I, Hutter F. CMA-ES for hyperparameter optimization of deep neural networks. *arXiv preprint arXiv:1604.07269*. 2016.

8. Salimans T, Ho J, Chen X, Sidor S, Sutskever I. Evolution strategies as a scalable alternative to reinforcement learning. *arXiv preprint arXiv:1703.03864*. 2017.

9. Vidnerová P, Neruda R. Evolving keras architectures for sensor data analysis, Federated Conference on Computer Science and Information Systems (FedCSIS), 2017 IEEE; 2017:109-112.

10. Albert R, Barabási A-L. Statistical mechanics of complex networks. *Rev Mod Phys*. 2002;74(1):47.

11. Newman M, Barabási A-L, Watts D-J. *The structure and dynamics of networks*, Vol. 19: Princeton University Press; 2011.

12. Albert R, Jeong H, Barabási A-L. Internet: diameter of the world-wide web. *Nat*. 1999;401(6749):130.

13. Redner S. How popular is your paper? An empirical study of the citation distribution. *Eur Phys J B*. 1998;4(2):131-134.

14. Ito T, Chiba T, Ozawa R, Yoshida M, Hattori M, Sakaki Y. A comprehensive two-hybrid analysis to explore the yeast protein interactome. *Proc Natl Acad Sci U.S.A*. 2001;98(8):4569-4574.

15. Barabási A-L. Network medicine - from obesity to the diseasom: Mass Medical Soc; 2007.

16. Jian F, Dandan S. Complex network theory and its application research on p2p networks. *Appl Math Nonlinear Sci*. 2016;1(1):45-52. https://content.sciendo.com/view/journals/amns/1/1/article-p45.xml

17. Barabási A-L, Pásfai M. *Network science*. Cambridge: Cambridge University Press; 2016. http://barabasi.com/networksciencebook/

18. Fortunato S, Castellano C. Community structure in graphs. In: Computational complexity; 2012:490-512.

19. Kernighan B-W, Lin S. An efficient heuristic procedure for partitioning graphs. *Bell Syst Tech J*. 1970;49(2):291-307.

20. Scott J. *Social network analysis*: Sage Publications; 2017.

21. Amaral L-A-N, Scala A, Barthelemy M, Stanley H-E. Classes of small-world networks. *Proc Natl Acad Sci. U.S.A*. 2000;97(21):11149-11152.

22. Marchiori M, Latora V. Harmony in the small-world. *Physica A: Statistical Mechanics and its Applications*. 2000;285(3-4):539-546.

23. Luo W, Lu N, Ni L, Zhu W, Ding W. Local community detection by the nearest nodes with greater centrality. *Inform Sci*. 2020.

24. Yanardag P, Vishwanathan S-V-N. Deep graph kernels. In: Proceedings of the 21th acm sigkdd international conference on knowledge discovery and data mining; 2015:1365-1374.

25. Kipf T-N, Welling M. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*. 2016.

26. Li J, Zhang H, Han Z, Rong Y, Cheng H, Huang J. Adversarial attack on community detection by hiding individuals. In: Proceedings of the web conference 2020; 2020:917-927.

27. Khodayar M, Wang J. Spatio-temporal graph deep neural network for short-term wind speed forecasting. *IEEE Trans Sustainable Energy*. 2018;10(2):670-681.

28. Veličković P, Cucurull G, Casanova A, Romero A, Lio P, Bengio Y. Graph attention networks. *arXiv preprint arXiv:1710.10903*. 2017.

29. Pérez-Benito F-J, Villacampa-Fernández P, Conejero J-A, García-Gómez J-M, Navarro-Pardo E. A happiness degree predictor using the conceptual data structure for deep learning architectures. *Comput Methods Programs Biomed*. 2019;168:59-68.

30. Spector P-E. *Summated rating scale construction: An introduction*, Vol. 82: Sage Publications; 1992.

31. Cacioppo J-T, Berntson G-G. Relationship between attitudes and evaluative space: a critical review, with emphasis on the separability of positive and negative substrates. *Psychol Bull*. 1994;115(3):401.

32. Newman M-E-J, Girvan M. Finding and evaluating community structure in networks. *Phys Rev E*. 2004;69(2):026113.

33. Clauset A, Newman M-E-J, Moore C. Finding community structure in very large networks. *Phys Rev E*. 2004;70(6):066111.

34. Wilks D-S. Cluster analysis, *International geophysics*; 2011:603-616.

35. Blondel V-D, Guillaume J-L, Lambiotte R, Lefebvre E. Fast unfolding of communities in large networks. *J Stat Mech Theory Exp*. 2008;2008(10):P10008.

36. Arenas A, Duch J, Fernández A, Gómez S. Size reduction of complex networks preserving modularity. *New J Phys*. 2007;9(6):176.

37. Kingma D-P, Ba J. Adam: a method for stochastic optimization. *arXiv preprint arXiv:1412.6980*. 2014.

38. Traag V-A. Faster unfolding of communities: Speeding up the Louvain algorithm. *Phys Review E*. 2015;92(3):032801.

39. Hagberg Aric, Swart Pieter, S Chult Daniel. Exploring network structure, dynamics, and function using networkx, Los Alamos National Lab.(LANL), Los Alamos, NM (United States); 2008.

40. Abadi M, Barham P, Chen J, et al. Tensorflow: a system for large-scale machine learning. In: 12th USENIX symposium on operating systems design and implementation (OSDI 16); 2016:265-283.

41. Joseph S, Linley P-A, Harwood J, Lewis C-A, McCollam P. Rapid assessment of well-being: the short depression-happiness scale (SDHS). *Psychol Psychother-T*. 2004;77(4):463-478.

42. Carver C-S. You want to measure coping but your protocol's too long: consider the brief cope. *Int J Behav Med*. 1997;4(1):92.

43. Francis L-J, Brown L-B, Philipchalk R. The development of an abbreviated form of the Revised Eysenck Personality Questionnaire (EPQR-A): its use among students in England, Canada, the USA and Australia. *Pers Individ Differ*. 1992;13(4):443-449.

44. Goldberg D. Manual of the ghq. *NFER, Windwor*. 1978.

45. Sherbourne C-D, Stewart A-L. The MOS social support survey. *Soc Sci Med*. 1991;32(6):705-714.

46. Lawrence S, Giles C-L, Tsoi A-C. Lessons in neural network training: overfitting may be harder than expected. In: AAAI/IAAI; 1997:540-545.