Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

Extracting knowledge from Deep Neural Networks through graph analysis

Vitor A.C. Horta [a,*], Ilaria Tiddi [b], Suzanne Little [a], Alessandra Mileo [a,*]

[a] Insight SFI Research Centre for Data Analytics, Dublin City University, Dublin, Ireland
[b] Faculty of Computer Science, Vrije Universiteit Amsterdam, De Boelelaan 1081-1089, 1081 HV Amsterdam, The Netherlands

ARTICLE INFO

Article history:
Received 23 December 2019
Received in revised form 1 February 2021
Accepted 13 February 2021
Available online 2 March 2021

Keywords:
Explainable AI
Deep representation learning
Graph analysis

ABSTRACT

The popularity of deep learning has increased tremendously in recent years due to its ability to efficiently solve complex tasks in challenging areas such as computer vision and language processing. Despite this success, low-level neural activity reproduced by Deep Neural Networks (DNNs) generates extremely rich representations of the data. These representations are difficult to characterise and cannot be directly used to understand the decision process. In this paper we build upon our exploratory work where we introduced the concept of a co-activation graph and investigated the potential of graph analysis for explaining deep representations. The co-activation graph encodes statistical correlations between neurons' activation values and therefore helps to characterise the relationship between pairs of neurons in the hidden layers and output classes. To confirm the validity of our findings, our experimental evaluation is extended to consider datasets and models with different levels of complexity. For each of the considered datasets we explore the co-activation graph and use graph analysis to detect similar classes, find central nodes and use graph visualisation to better interpret the outcomes of the analysis. Our results show that graph analysis can reveal important insights into how DNNs work and enable partial explainability of deep learning models.

1. Introduction

Modern Deep Neural Networks (DNNs) can leverage large amounts of data to be efficiently trained to perform hard tasks such as translating languages and identifying objects in an image [1,2]. DNNs have been shown to achieve good performance in such complex tasks where traditional machine learning methods may fail due to the high dimensionality of the data [3].

Despite this success the lack of interpretability and explainability in understanding why a DNN makes a particular decision is still a open challenge [4]. This can be crucial in situations where the ability to explain decisions and misjudgements is key, like in medical diagnosis, law enforcement, financial analysis and more. Not only that but more insights on the inner working of deep learning models can pave the way for better transfer learning (even with limited amount of available training data), debugging and engineering of complex deep architectures.

The desire to understand the black box inside which Artificial Neural Networks (ANNs) work is not new. There were already attempts to explain ANNs even when the quality and availability of learning data and the lack of computational resources did not yet enable full exploitation of deep learning approaches [4–6].

More recently the number of papers that tackle this problem has increased considerably [7] and strategies with different perspectives have been developed to help understand the decision making process of DNNs. Although existing approaches have made a considerable progress in explaining the high level behaviour of DNNs, there is no general mechanism that provides an interpretation for the internal representation of DNNs while being architecture agnostic. In other words, existing methods either consider the DNN as a black box (e.g Pedagogical Rule Extraction methods [8]) or are limited to specific architectures, such as Decompositional Rule Extraction methods [9], which often work only for fully-connected layers.

We aim to provide a general mechanism to interpret DNNs and extract knowledge acquired by deep learning models. Our method takes inspiration from recent studies in neuroscience [10,11] that use network analysis and graph representations to interpret brain structure. These studies have shown that analysing brain networks, using graph theory methods and concepts like communities in graphs and node centrality, can give important insights on the architecture, development, and evolution of brain networks. More specifically, researchers in neuroscience use functional graphs to study the brain by representing and analysing statistical dependencies among neuronal activities.

Inspired by these approaches the co-activation graph method we introduced in [12] was developed to interpret DNNs by using

[*] Corresponding authors.
E-mail address: vitor.araujocautierohorta2@mail.dcu.ie (V.A.C. Horta).

https://doi.org/10.1016/j.future.2021.02.009

representations similar to functional graphs based on statistical correlations among artificial neural activities. In a co-activation graph *nodes* represent neurons in a DNN and *weighted relationships* indicate a statistical correlation between their activation values. This representation is general enough to represent connections between pairs of neurons in any layer of the neural network, including hidden (convolutional and dense) layers and the output layer. Unlike previous approaches this makes it possible to study relationships between neurons in any layer of the DNN. The key hypothesis of our approach is that the knowledge encoded in the co-activation graph reflects the knowledge acquired by the DNN in the training phase. As a result we can use graph analysis tools on the co-activation graph to gain insights on how the model works.

In our preliminary investigation [12] we introduced the general idea of a co-activation graph and focused on two small experiments to assess its suitability to correctly represent the workings of the inner layers of a DNN. In this paper we extend our early work in three ways. First, we reproduce our previous experiments using a more complex architecture represented by the state-of-the-art model MobileNetV2 [13]. We use MobileNetV2 to show that our approach can be applied to deeper and more complex models with separable convolutions. Second, beyond considering the MNIST-handwritten and MNIST-fashion datasets, we conduct an experiment over a more complex dataset, CIFAR-10 [14], to show that our approach can give interesting insights on non-trivial datasets. Third, we explore the notion of node centrality and study the association between central nodes and their respective neurons in the DNN for all three datasets.

We can summarise the key contributions of this paper as follows: (i) we provide a formal definition of co-activation graphs so that we can identify properties and graph analytics processes for extracting meaningful knowledge from the deep learning model; (ii) we apply graph analysis beyond class similarity, for detecting groups of neurons that work together to predict similar classes on models and datasets at different levels of complexity; (iii) we explore and apply the notion of graph centrality to detect central nodes that represent the most important neurons in hidden layers.

We demonstrate that by querying the co-activation graph it is possible to detect which neurons in the hidden layers are more likely to impact the prediction value of each output class. Graph algorithms are used on the co-activation graph to find groups of classes that are more similar to each other from the point of view of the DNN, and groups of neurons that are constantly being activated at the same time. An interesting insight we obtained from our investigation is that classes with high numbers of shared neurons (overlapping nodes in the graph) are responsible for most of the mistakes in the DNN. This is an evidence that overlapping nodes in the graph might indicate which neurons in the DNN should be considered for a fine-tuning process to better differentiate the overlapping classes. In addition, we show that central nodes in the co-activation graph might indicate important neurons in the DNN. Finally, since explaining DNNs is a known challenge and many different methods are available in the literature, we suggest how our approach can be combined with other techniques and how they can benefit from each other to explain DNNs.

The remainder of the paper is organised as follows. Section 2 discusses related approaches to extract knowledge from neural networks. Section 3 provides a formal definition of the co-activation graph and recalls the technique to generate such graph. In Section 4 we present our experiments on MNIST-handwritten, MNIST-fashion and CIFAR-10 datasets and we discuss how we can explain DNNs analysing community structure, node centrality and graph visualisations on the co-activation graph. Section 5 summarises our discussion and Section 6 presents our conclusions and future steps.

2. Related work

The problem of understanding and explaining the decision making process of neural networks has been explored by researchers since the 1990s when the models were still shallow (models with only a few hidden layers) and consisted of only fully connected layers. Even with these simplifications compared to Deep Neural Networks (DNNs), providing meaningful explanations for such models proved to be a challenging task.

One of the first approaches proposed in [5] measured the contribution of input variables to the output layer to provide explanations for the model. This approach can be extended for measuring the contribution of neurons in the hidden layer [6] but one limitation of the strategy used is that it cannot be applied to modern architectures such as convolutional and recurrent neural networks.

After the development of more modern deep architectures, different strategies for explaining DNNs were also developed. One such strategy is based on rule extraction and the goal is to extract sets of rules that explain the decisions of complex black box models. A comprehensive review of these methods can be found in [15]. There are three main types of rule extraction methods: decompositional, pedagogical and hybrid or eclectic strategy. Methods based on a decompositional strategy extract rules by examining activation and weights in the neural networks [9,16,17] but they cannot be applied to convolutional layers. The pedagogical strategy uses the decisions made by DNNs to extract rules without exploring the model architecture [8]. Unlike methods based on decompositional strategy, pure pedagogical methods can be applied to any neural network because they do not rely on the architecture. As a drawback, however, these methods are not able to explain the inner working of hidden layers as they use the DNN as a black box. It is also possible to use a hybrid of these two approaches, which is the eclectic strategy [18,19], but current works apply the technique only on top of fully-connected layers.

Another approach that has been proposed more recently to explain DNNs is to use visualisation techniques [20–22]. This approach is based on providing rich visualisations that can help understanding, for example, which parts of the input are more relevant for the model's predictions. With these methods it is also possible to understand the role of convolutional filters or entire convolutional layers by looking at the visual representations they provide when activated with some specific data. Extensive surveys of visualisation methods are provided in the literature [23–25]. Despite visualisation techniques playing an important role in understanding the behaviour of DNNs, most visual analyses require human interpretation, which limits their scalability.

The approach proposed in this paper is based on the notion of a co-activation graph and it brings a novel perspective in addressing the explainability problem in DNNs. The methodology takes inspiration from recent studies in neuroscience [10,11] that use graph representations and concepts like communities in graphs and node centrality to analyse brain activity.

The notion of a co-activation graph introduced in [12] and used in this paper differs from existing works in two ways. First, it can represent both convolutional and fully connected layers in any depth of DNNs. Second, it allows the use of automatic methods such as graph analysis tools to discover interesting patterns, interactions and properties of the internal workings of DNNs.

In this sense a recent work [26] that proposes a graph representation for embedding vectors is the most similar to the co-activation graph idea. This approach, however, is more concerned about connecting input instances to neurons in the hidden layers, while our approach is focused on exploring relationships between pairs of neurons in the hidden layers and the output layer. We believe there are some interesting synergies between the two approaches that is worth exploring further. In Section 5 we briefly discuss how the two approaches could be potentially combined in future work.

3. Co-activation graph: Definition and construction

The key idea behind the notion of the co-activation graph is that it can be used to extract and represent knowledge from trained Deep Neural Networks (DNNs) and use it to better understand how the hidden part of the model works. Remember that the co-activation graph connects every pair of neurons of any type (fully connected or convolutional) and located in any layer of the neural network. This section provides a formal definition of the co-activation graph and recalls the general idea on how it is constructed.

Nodes of the co-activation graph correspond to neurons in the DNN and weighted edges represent a statistical correlation between them based on their activation values. We refer to this graph as a *co-activation graph*, since the relationships between its nodes represent how their activation values are correlated. The main idea of the co-activation graph is to create a relation between pairs of neurons in any depth of the hidden layers and neurons in the output classes, since the latter are more comprehensible for humans.

Definition 3.1. A co-activation graph can be represented as an undirected graph $G = (V, E)$ where $V = \{v_0, v_1, \dots, v_n\}$ is the set of n nodes that represent the neurons in the DNN and E is the set of weighted relationships (edges) $e_{ij} = (v_i, v_j, w)$ between pairs of neurons v_i and v_j with weight w_{ij}, considering $0 \leq i < n$ and $0 \leq j < n$. The weight w_{ij} represents a statistical correlation between the activation values of v_i and v_j.

Given a data sample S, the first step to calculate w_{ij} is to feed the DNN with S. Then extract the sets of activation values $A(v_i, S) = \{a_{i0}, a_{i1}, \dots, a_{in}\}$ and $A(v_j, S) = \{a_{j0}, a_{j1}, \dots, a_{jn}\}$, where a_{i0} is a single activation value of neuron v_i for a single data input $s_0 \in S$. The weight w_{ij} is then obtained by applying a statistical correlation on $A(v_i, S)$ and $A(v_j, S)$, as shown in Eq. (1):

$$w_{ij} = Spearman\_corr(A(v_i, S), A(v_j, S)). \quad (1)$$

We chose to use Spearman coefficient since we do not expect linear relationships between neurons' activation values. Therefore, edge weights vary in the range of [−1,1]. To clarify this process we recall the three steps below on how to generate a co-activation graph for a given trained DNN. For this, consider a DNN with n neurons and a data sample $S = \{s_0, s_1, \dots, s_m\}$.

1. **Extract activation values**: The first step to build a co-activation graph is to feed the given DNN with S. Then, for each neuron v_i and each data input $s_h \in S$ where $0 \leq h < m$, extract a single activation value a_{ih}. The result is the set $\{A(v_0, S), A(v_1, S), \dots, A(v_n, S)\}$, where $A(v_i, S)$ represent all activation values of each neuron v_i for the whole data sample S.
   For dense layers this process is straightforward because each neuron outputs a single activation value. Filters in convolutional layers will output multiple values since they have different activation values for each region in the input. To overcome this and make our approach work for convolutional layers, the average pooling technique is applied to extract a single value for convolutional filters. Although some spatial information is lost in this process, it allows the extraction of a single activation value for each filter while keeping the dimensionality low [27].
2. **Define and calculate edge weights**: After collecting the activation values $A(v_i, S)$ for each neuron v_i in the DNN, the next step is to define the relationships between pairs of neurons. For each pair of neurons v_i and v_j, Eq. (1) is applied using the activation values $A(v_i, S)$ and $A(v_j, S)$ to calculate a statistical correlation that will define the relationship weights w_{ij} between each pair of neurons. The result of this step is a matrix containing weights w_{ij} for every neuron pair v_i and v_j that can be used to construct the set of edges E.
3. **Build and analyse the co-activation graph**: Given the set of edges E that describes the relationship between every pair of neurons, the co-activation graph can be built and analysed using any suitable computational tool for graph structures. In this paper, we chose to represent the co-activation graphs in the graph database Neo4j[1] to facilitate data persistence and enhance reproducibility. The result of this final step is a graph where nodes represent neurons of any layer in the DNN and weighted edges indicate the correlation between their activation values. Our evaluation in Section 4 demonstrates that this graph correctly encodes the knowledge contained in the hidden layers of the DNN and we can then explore the graph structure and use graph analysis tools to understand relationships between neuron pairs in hidden layers and output classes.

It is easy to imagine how the result of this process is a very dense graph where each possible pair of nodes is connected. Analysing and visualising such a graph would be both difficult and computationally expensive. However, it is possible to define a threshold to remove relationships with small weights. Although we do not provide a systematic way for determining the threshold, it is feasible to define it empirically given that the relationship weights vary in a constrained range of [−1,1] and can be interpreted as statistical correlations. By doing this, the resulting process will be a less dense graph where only relationships with significant positive correlations are kept.

In Section 4 we validate our approach by conducting experiments on models and datasets with different levels of complexity.

4. Evaluation: Explaining DNNs through co-activation graphs

The first step in our evaluation is to demonstrate whether the co-activation graph encodes the same knowledge as the Deep Neural Network (DNN) model. In order to do that we have conducted experiments with community analysis and centrality analysis over three different models and datasets. We extend our preliminary experiments from [12] in two ways. First, we perform a community analysis over a deeper model on a more complex dataset to see if the results are consistent in a more complex environment. Second, a novel centrality analysis is conducted over the three models to study the presence of central nodes in co-activation graphs and understand the association between node centrality and neuron importance in the DNN.

In what follows, we start by introducing the datasets, model architectures and how to build a co-activation graph for each model. Then, the community and centrality analysis are performed to show how a graph algorithm applied to the co-activation graph can help understand the underlying DNN model.

[1] https://neo4j.com

4.1. Building the co-activation graph: Datasets and neural architectures

Three well known datasets were used for the experiments: *MNIST handwritten digits* [28], *MNIST fashion* [29] and *CIFAR-10* [14]. For the MNIST variants, two shallow DNNs were trained from scratch. For CIFAR-10, a state-of-the-art model was used, as detailed below.

4.1.1. MNIST-handwritten and MNIST-fashion

The classes in *handwritten digits* dataset refer to digits from 0 to 9 and classes in *fashion* dataset are related to clothes. Both datasets contain ten classes, with 60,000 training images and 10,000 testing images. The DNN used for *handwritten digits* contains two convolutional layers and three fully connected layers and the DNN used for the *fashion* dataset has three convolutional layers and two fully connected layers. The Adam optimiser was used with learning rate 0.002. After training for 10 epochs, these models achieved an accuracy greater than 97%. Although it is possible to obtain higher accuracy with different architectures, we chose to use these models since we also want to analyse the reason behind mispredictions.

The co-activation graph was built for each DNN following the three steps discussed in Section 3. We first fed the DNNs with data samples from the testing set to extract activation values for each neuron. Then, we calculated the Spearman correlations between those neurons and built a co-activation graph for each DNN. In order to keep only relevant relationships in our graph, we applied a threshold of 0.3 so that only neurons with some significant correlation are connected. This threshold value was chosen empirically aiming to maintain only connections with a significant positive correlation and without partitioning the graph into multiple components.

4.1.2. CIFAR-10

The *CIFAR-10* dataset contains ten classes: *Dog; Cat; Horse; Frog; Bird; Deer; Airplane; Ship; Truck; Automobile*. There are 50,000 training images and 10,000 testing images. The DNN used for *CIFAR-10* was the *MobileNetV2* [13] that was designed to achieve state-of-the-art results on the challenging *ImageNet* dataset.

The *MobileNetV2* architecture contains an initial convolutional layer, 19 residual bottleneck layers followed by another convolutional layer and a fully connected layer. The model used in this experiment achieved 94.43% accuracy on the *CIFAR-10* dataset.

To build a co-activation graph for this model the first step was to extract the activation values from each neuron in every layer. For the convolutional and fully connected layers we extracted the activation values using the same approach described in Section 3. For the bottleneck layers we had to adapt our strategy as these layers are formed by depthwise and pointwise convolutions. In this case we collected the RELU activation values from the last pointwise convolution, since this is the last transformation that is performed on each bottleneck layer.

After collecting the activation values we calculated the Spearman correlation between each pair of neurons and connected them to generate the co-activation graph. Finally, we applied the same threshold of 0.3 to keep only relevant relationships in our graph. With these settings the *MobileNetV2* co-activation graph for *CIFAR-10* contains 4012 nodes and 184,144 edges.

4.2. Community structure analysis

The presence of community structure in graphs and networks is a key aspect worth investigating to understand interesting properties of the graph itself and the knowledge it represents. The results of our preliminary investigation in [12] already demonstrated that community analysis over the co-activation graph can help identify classes that are similar from the point of view of the DNN. One interesting finding was the semantic alignment across similar communities for the *MNIST-fashion* dataset. It is possible to see from Table 1 that classes like *shirt* and *t-shirt* were put in the same community while *sandals* and *sneakers* are grouped in a different community, reflecting the semantic alignment among classes grouped in the same community structure for that dataset.

Table 1
Communities on MNIST-fashion.

Community	Classes
C1	T-shirt/Top; Pullover; Coat; Shirt
C2	Trouser; Dress;
C3	Sandal; Sneaker; Bag; Ankle Boot

Table 2
Communities on MobileNetV2 for CIFAR-10 with default resolution.

Community	Classes
C1	Deer; Dog; Horse
C2	Frog; Bird; Cat
C3	Airplane; Ship; Truck; Automobile
Modularity	0.489

In this section we extend and elaborate on the validity of this result by analysing the community structure of a deeper model (*MobileNetV2*) on a more complex dataset (*CIFAR-10*). As described before the *CIFAR-10* dataset contains ten classes: *Dog; Cat; Horse; Frog; Bird; Deer; Airplane; Ship; Truck; Automobile*. As humans, it would be reasonable to separate these classes in two groups: Animals and Vehicles. To understand if *MobileNetV2* also organises these classes into different, semantically aligned groups, a community detection algorithm was run on the co-activation graph generated from MobileNetV2 trained on *CIFAR-10* in the same way as in [12] for the *MNIST* datasets.

The Louvain community detection algorithm [30] was chosen for this analysis because, besides being a well established algorithm, this method also outputs a modularity coefficient that can be used to check how the community structure differs from random graphs. The value of modularity varies in the range of [−1,1], where higher values indicate that connections between nodes in the same community are stronger than nodes in different communities. The only required parameter for the Louvain algorithm is the resolution, which can be adjusted to detect a higher number of smaller communities or a lower number of bigger communities. For this experiment the resolution was set to the default value 1.0. Since the Louvain algorithm is not deterministic, ten multiple executions were conducted to check the stability of found communities. For this experiment, communities proved to be stable, meaning that all executions resulted in the same community structure. Table 2 shows the detected communities.

As expected Animal and Vehicle classes were put in different communities showing once more that classes in the same communities are semantically aligned and confirming previous experiments. However, this first result also shows that the model separated Animal classes into two different groups, one containing {Deer; Dog; Horse} and the other containing {Frog; Bird; Cat}.

To further study these communities we analysed the similarity between them. In this case we can expect that $C1$ and $C2$ are more similar between each other than they are with $C3$, since the former contains only animals and the latter only vehicles. To verify this, we increased the resolution parameter to achieve fewer communities (i.e. bigger ones). We observed that when we set the resolution parameter to 1.2, $C1$ and $C2$ are merged by the

Table 3
Communities on MobileNetV2 for CIFAR-10 with increased resolution.

Community	Classes
C1	Deer; Dog; Horse; Frog; Bird; Cat
C2	Airplane; Ship; Truck; Automobile
Modularity	0.489

Table 4
Communities for MNIST-fashion with increased resolution.

Community	Classes
C1	T-shirt/Top; Pullover; Coat; Shirt; Trouser; Dress;
C2	Sandal; Sneaker; Bag; Ankle Boot
Modularity	0.474

Louvain algorithm into one single community, indicating that not only classes in the same community have a semantic similarity but also there is a semantic hierarchy reflected by the analysis as these merged communities are semantically aligned, as shown in Table 3. Interestingly, a similar behaviour happens when we apply the Louvain algorithm with increased resolution for the *MNIST-fashion* dataset. As shown in Table 4, the community containing classes *Trouser; Dress* was merged with *C1*, which contains *T-shirt/Top; Pullover; Coat; Shirt*.

Detecting these groups and having the evidence that they carry a semantic meaning is important for different reasons. First, mistakes are more likely to occur between classes that are semantically similar and therefore belong to the same community, as we will further discuss in Section 4.3. This is also intuitive, since from the human's perspective misclassifications between two animals or two vehicles should be more frequent than mistakes between *Frog* and *Truck* for example. Second, if we consider that neurons in convolutional layers work as feature extractors, it is plausible to expect that neurons for extracting animal parts are more likely to be found in $C1$, while vehicle parts are in $C2$. This claim however would require a more rigorous evaluation that is beyond the scope of this paper and will be considered in future work.

As a result of our investigation, we can say that this third community analysis has shown that results for the co-activation graph generated on deeper and more complex models are consistent with those in [12].

4.3. Centrality analysis

In graph analysis the centrality of a node provides valuable insights on how important this node is in a graph. The notion of importance depends on both the graph domain and the chosen centrality measure. In this experiment the graph domain refers to neurons in a DNN and the objective is to discover whether centrality measures can help reveal neurons that are more important for the DNN. Therefore the third contribution of this paper is to analyse the notion of node centrality in co-activation graphs and determine the association between central nodes and their respective neurons in the DNN. For this consider that a specific neuron is redundant or less important if we can remove it from the DNN without having a significant accuracy drop.

Two different centralities are used for this analysis: degree centrality and PageRank centrality [31]. The degree centrality of a node n_i, which can be calculated using Eq. (2), indicates the number of relationships n_i has with other nodes. Since our graph contains weighted relationships, to calculate the degree centrality we simply sum the weights of the edges that connect n_i with other nodes. In Eq. (2), A is an adjacent matrix of size N where A_{ij} represents the weighted edge between nodes n_i and n_j, considering $0 \leq i < N$ and $0 \leq j < N$. In the case of a co-activation graph, a node with a high degree corresponds to a neuron that correlates strongly with many other neurons.

$$D(n_i) = \sum_{j=0}^{N-1} A_{ij} \tag{2}$$

The PageRank centrality of a node indicates its importance by leveraging not only its direct relationships but also the importance of its neighbours. The PageRank of each node is first initialised with value 1. Then it is iteratively updated using Eq. (3), where N is the total number of nodes in the graph, S_{n_i} is a set containing the neighbours of n_i, A_{ij} is the edge weight between the two nodes, d is a damping factor that controls how often one randomly jumps to another node, and $D(n_j)$ is the degree of node n_j as shown in Eq. (2). This iterative process ends when the PageRank of each node from the previous iteration differs from the current iteration by less than a predefined epsilon, which was set to 0.001 in our experiments.

$$PR(n_i) = \frac{1-d}{N} + d \sum_{n_j \in S_{n_i}} A_{ij} \frac{PR(n_j)}{D(n_j)} \tag{3}$$

To check if there is an association between node centrality and neuron importance, the following steps were performed:

1. Calculate node centrality in the co-activation graph
2. Pick the node with the highest (or lowest) centrality
3. Remove the respective neuron from the DNN and from the co-activation graph
4. Check the accuracy loss
5. Repeat

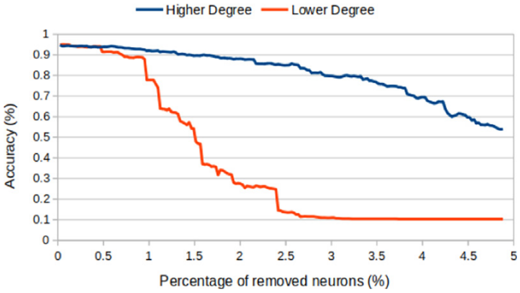
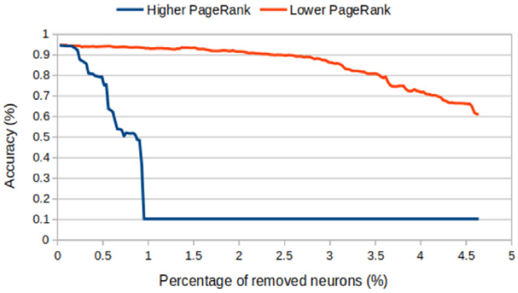
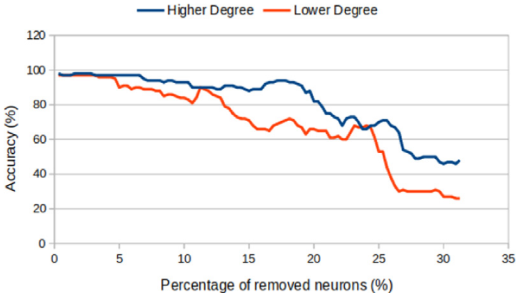
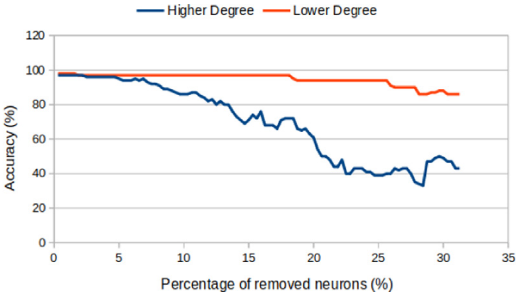
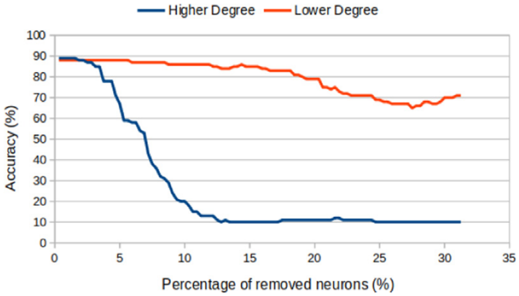
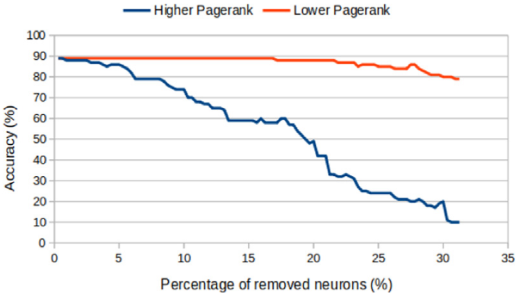
Table 5 shows the result of the above procedure for both degree and PageRank centralities for the three models: *CIFAR-10*, *MNIST-handwritten* and *MNIST-fashion*. In the case of PageRank centrality it is possible to see that there is an interesting correlation between the PageRank centrality in the co-activation graph and the neuron importance in the DNN. This can be observed from the third column of Table 5 that shows when we remove neurons ordered by higher PageRank there is a strong accuracy loss in the DNN. In contrast if neurons with lower PageRank are removed, the accuracy is stable for a longer period. This behaviour is consistent through all three datasets and it indicates that analysing PageRank centrality over co-activation graphs has the potential to reveal important neurons in the respective DNN.

The same consistency cannot be observed for the degree centrality. The second column of Table 5 shows that for *CIFAR-10* nodes with high degree have a high impact when removed from the DNN. However, for *MNIST-fashion*, the opposite behaviour is observed. In addition, when analysing the impact of degree centrality for the *MNIST-handwritten* dataset, removing neurons for either lower or higher degree shows a similar behaviour, indicating that degree centrality is not very informative for this model.

It is important to note that a node with a low degree might still have a high PageRank and thus these results are not contradictory. Instead this can be considered as initial evidence that a co-activation graph might be used as a strategy for pruning algorithms on pre-trained models since removing a considerable amount of nodes with low PageRank caused only a minor accuracy drop in all three DNNs in this experiment.

The centrality analysis shows that there is an association between node centrality in the co-activation graph and neuron importance in the DNN. This represents further evidence that the knowledge contained in the co-activation graph reflects the knowledge encoded in the DNN. However, one current challenge is to understand the role of central nodes. In Section 5 we discuss possible strategies to investigate this issue as well as possible limitations of the proposed method.

Table 5
Model accuracy after removing neurons based on their node centrality in the co-activation graph.

Dataset	Pruning by Degree	Pruning by PageRank
CIFAR-10		
Handwritten		
Fashion		

4.4. Visualising co-activation graphs

Graph visualisations are useful to derive new insights from data and to formulate new hypotheses based on how the graph is visually distributed. This section shows how graph visualisations can help understand the results achieved by each of the graph theory methods applied previously and provides a visual way to exploit co-activation graphs to interpret the knowledge contained in the DNN. For this the *MNIST-fashion* dataset is used as an example and visual representations are presented for the communities and node centralities that were obtained and analysed in Sections 4.2 and 4.3, respectively.

When visualising a co-activation graph it is useful to distinguish nodes in the hidden layers from nodes in the output layers (target classes), since the latter have a clear interpretation for humans. In Fig. 1 blue nodes represent neurons in the hidden layers while yellow and labelled ones represent the output classes. After distinguishing these nodes it is easier to visualise each step of the previous analyses.

From Fig. 1, it is possible to see the impact of applying a specific threshold to the co-activation graph. After applying a threshold of 0.3 to the *MNIST-fashion* dataset, the density of the graph is visibly reduced if compared to a fully-connected graph but the graph still consists of a single component. It can also be seen from this figure that the visualisation algorithm, *ForceAtlas 2* [32], has placed some classes close to each other (e.g *Sandal, Sneaker* and *Ankle Boot*) intuitively indicating that the graph has a community-like structure, which is shown in more detail in Fig. 2.

In Fig. 2 the communities discovered in Section 4.2 are represented by different colours. From this community structure a first observation is that the node representing class *Bag* is quite close to the red community although it belongs to the green one. This could be an indication that the *Bag* class is at the intersection of these two communities and probably should not be considered as part of a single community. This behaviour could be captured by community detection algorithms that consider overlapping nodes, such as [33,34], and we plan to investigate this further. In addition, classes *Dress* and *Trouser*, although they belong to the pink community were placed close to the red community by the *ForceAtlas2* visualisation algorithm. This means that they are probably more similar to classes *Coat, Pullover, Shirt* and *T-shirt* than to *Sneaker, Ankle Boot, Sandal* and *Bag*. In fact this observation corroborates with the results presented in Section 4.2 which showed that the community containing *Dress* and *Trouser* was merged with the red community when we increased the resolution parameters for the Louvain algorithm.

As a third step in the visual analysis Fig. 3 illustrates nodes according to their PageRank centrality represented by the size of the nodes. We have already seen from Section 4.3 that central nodes tend to be important neurons in the DNN. Now, according to the visualisation, the nodes with high centrality seem to belong to dense regions in the graph while less central nodes are either peripheral nodes or nodes in less dense regions. This supports the intuition that, by removing the most central nodes, the model will suffer a considerable accuracy loss since large number of neurons (dense region) can be impacted.

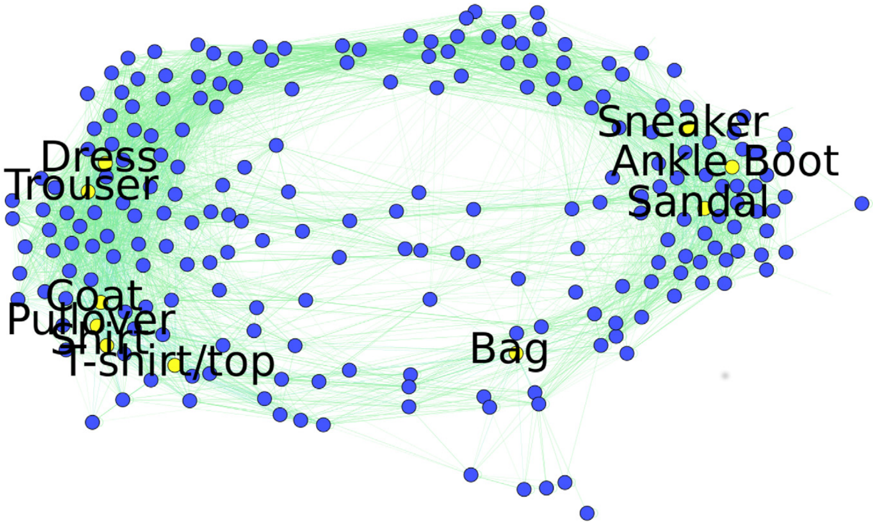


Fig. 1. Visualisation for *MNIST-fashion* co-activation graph made on Gephi. Blue nodes represent neurons in hidden layers and yellow and labelled nodes represent neurons in the last layer (output classes).

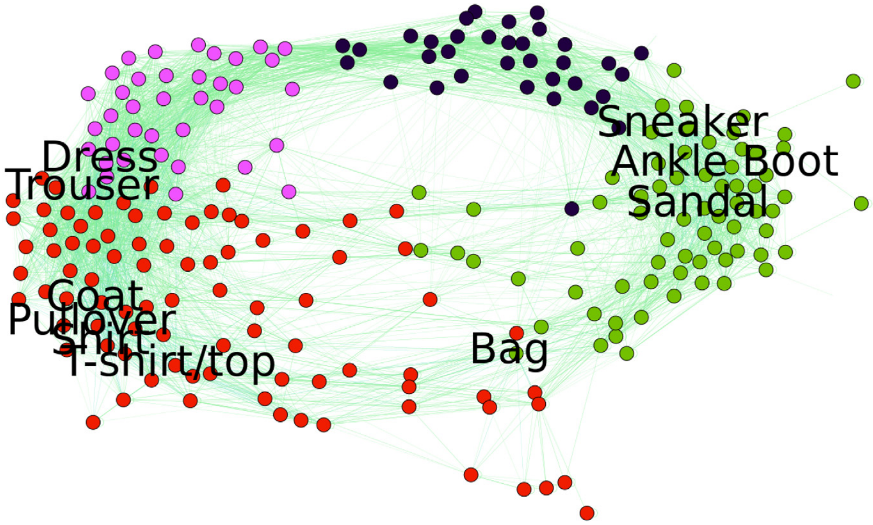


Fig. 2. Visualising the communities detected by the Louvain algorithm. Communities are represented by different colours.

The visualisations for the communities and central nodes also suggest that if we combine the community structure with the centrality analysis we might be able to automatically detect which neurons are more important for the prediction of classes within each of the communities. For example, if we remove central nodes related to the green community (big green nodes), this should result in higher impact on accuracy for classes in the green community (e.g *Sandal*) than classes in the red community (e.g. *Coat*). This is an intuition that should be further investigated.

We can see that by using visualisation techniques we can improve our understanding of the results obtained by the different graph analyses, like the community structure and the positioning of central nodes. This can be considered as another advantage of extracting knowledge from DNNs and representing it in co-activation graphs since each analysis performed in the latter can be evaluated and visualised in a transparent manner. In the next section, we present the limitations we have identified for the current approach and discuss possible strategies to overcome them.

5. Discussion and next steps

The evaluation conducted in Section 4 demonstrated that results obtained by our approach are consistent across datasets and neural architectures of different complexity. However, in this section we discuss some limitations of our approach that should be taken into account and indicate possible ways to overcome them. We will also discuss other interesting open issues worth considering in the next steps of this research.

One limitation is that both the community and centrality analysis are biased by the chosen threshold. Choosing this threshold is not a trivial task. A possible way around this would be to use a deeper understanding of the graph distribution to help define it.

A second limitation is the strategy used to extract activation values from convolutional layers. In our experiments we used spatial pooling. This makes it easier to extract and represent convolutional filters but it causes some information loss. One possible strategy to overcome this is to use multiple nodes to represent all possible slices covered by each convolutional filter.

When comparing with alternative approaches it is important to note that the method proposed in this paper aims to explain Deep Neural Networks (DNNs) from a novel perspective

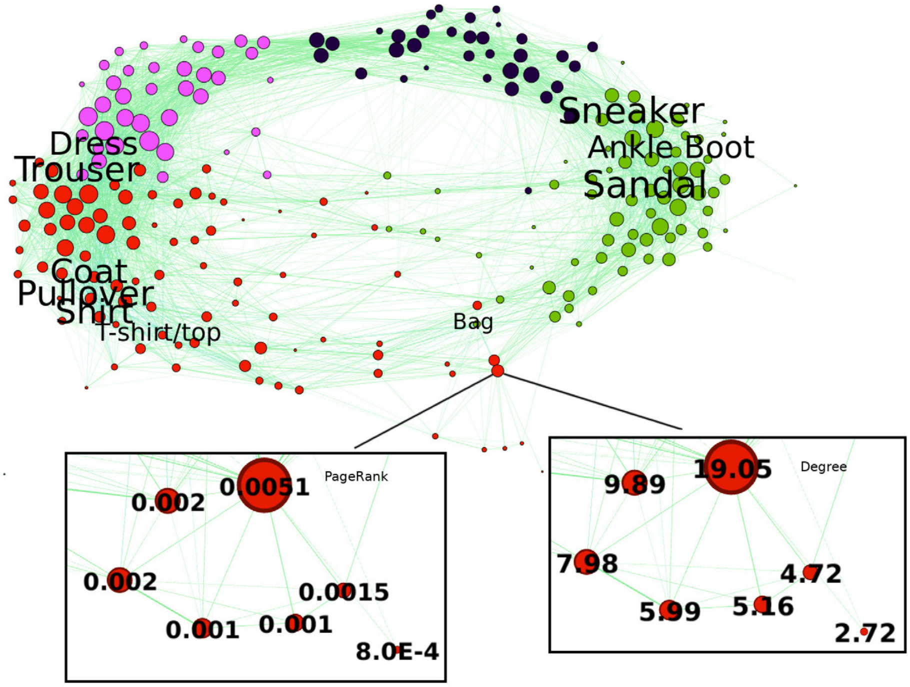


Fig. 3. Visualising nodes according to their PageRank centrality. Bigger nodes represent the most central ones.

and thus a direct and quantitative comparison is not possible at the moment. However, we believe that both community and centrality analysis could be combined with other explanation techniques to generate mutual benefits for better understanding of DNNs. For example, visualisation techniques might use information about communities and node centrality to decide which neurons are more important and therefore should be analysed. Visualisation techniques might also help in understanding why communities were formed in this particular way as well as the role of central nodes. By combining the method proposed in this work with existing approaches to enhancing explainability (such as visualisation techniques) we expect to achieve richer and complementary explanations that may improve understanding of how our approach compares to existing ones such as [22,35].

The observation that the detected communities contain classes with a similar semantic meaning is also first evidence that our approach might be used to integrate the knowledge acquired by DNNs with external knowledge. In this regard an interesting avenue for investigation is to use information from knowledge bases such as ConceptNet or DBpedia to identify properties shared among classes in a community. In our experiments, for example, even though our datasets contain a small number of classes, it would be possible to formalise that C1 is related to *dbo:Animal* while C2 is related to *dbc:Transport*. While this is a trivial example, we plan to look into using richer datasets such as CIFAR-100 [36] or ImageNet [37] to explore this knowledge linkage as these might lead to a more valuable community structure with potential for a richer semantic interpretation.

Finally, to include more semantic information in our approach we are considering the addition of connections between the input layer and the co-activation graph. This is important particularly due to the existence of datasets with contextualised images, such as CoCo [38] and ImageNet, and could potentially be done based on recent work in feature extraction [26,27].

6. Conclusion

In this paper we formalise and experiment on a novel approach to analyse and explain the inner workings of deep learning models. The proposed methodology relies on the notion of co-activation graph introduced in [12] and formalised in Section 3 to extract and represent knowledge from a trained Deep Neural Network (DNN). In the co-activation graph nodes represent neurons in a DNN and weighted relationships indicate a statistical correlation between their activation values. This representation connects neurons in any layer of the neural network, including hidden (convolutional and dense) layers, with the output layer.

The extensive evaluation conducted across models and datasets at different levels of complexity corroborates initial results showing that the co-activation graph representation is compatible with the knowledge encoded in the DNN. We started the evaluation process by reproducing the community analysis, previously performed in [12], with a more complex dataset (CIFAR-10) using the state-of-the-art model MobileNetV2. The new results are compatible with the ones in [12], since the communities detected for MobileNetV2 reflect groups of semantically aligned classes. An important aspect of this result is that the DNN not only has a good accuracy for the dataset but it also seems to be distinguishing groups of classes in an intelligent manner. In this case, for the CIFAR-10 dataset, the community analysis showed that the model learned how to separate classes related to animals from vehicle classes.

A step further in our analysis focuses on exploring the notion of centrality. Our evaluation showed that there is an association between central nodes in a co-activation graph and their respective neurons in the DNN. It was demonstrated that the DNN drastically loses accuracy if neurons with high PageRank centrality are removed from the model, while the accuracy remains stable if neurons with low PageRank centrality are removed. This indicates that central nodes in the co-activation graph may represent important neurons in the DNN and that our approach might be considered for pruning the DNN. This is of interest, for example, in applications where dimensionality reduction is crucial.

It was also shown that graph visualisation techniques can be used to clarify the result of each analysis performed over co-activation graphs adding a level of transparency. Unlike black box models we showed that by visualising co-activation graphs it is possible to better understand why a certain algorithm produced the resulting community structure and what it is expected when changing certain parameters in the algorithm. Graph visualisations can also help interpret why central nodes in co-activation graphs tend to be important nodes in DNNs since they are visually connected to dense regions and thus can impact many other neurons.

Our next steps include an investigation into how co-activation graphs can be combined with other sources of knowledge. As a first step, since communities detected in co-activation graphs already proved to group together classes with high semantic similarity, it would be interesting to conduct this analysis over richer datasets. On CIFAR-100, for example, it is reasonable to expect that communities would match the given superclasses. For other datasets, like ImageNet, we might be able to use information from WordNet to see if there is any relation between communities and the WordNet hierarchy. By doing this, we expect to have a semantic interpretation for which parts of the DNN are responsible for different patterns in the dataset.

Finally, we want to explore the potential of combining our approach with methods such as visualisation techniques to achieve more robust explanations for DNNs. Such combinations would also potentially help discover the meaning of different node centrality notions as well as when and why central nodes are more important in the DNN.

CRediT authorship contribution statement

Vitor A.C. Horta: Conceptualization, Methodology, Writing - original draft. **Ilaria Tiddi:** Conceptualization, Reviewing and editing. **Suzanne Little:** Supervision, Conceptualization, Reviewing and editing, Visualization. **Alessandra Mileo:** Supervision, Conceptualization, Methodology, Writing - review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This publication has emanated from research supported by Science Foundation Ireland (SFI) under Grant Number 17/RC-PhD/3483 and Grant Number SFI/12/RC/2289_P2, co-funded by the European Regional Development Fund.

References

[1] A.B. Nassif, I. Shahin, I. Attili, M. Azzeh, K. Shaalan, Speech recognition using deep neural networks: A systematic review, IEEE Access (ISSN: 2169-3536) 7 (2019) 19143–19165, https://doi.org/10.1109/ACCESS.2019.2896880.

[2] A. Voulodimos, N. Doulamis, A. Doulamis, E. Protopapadakis, Deep learning for computer vision: A brief review, Comput. Intell. Neurosci. 2018 (2018) 1–13, https://doi.org/10.1155/2018/7068349.

[3] B. Liu, Y. Wei, Y. Zhang, Q. Yang, Deep Neural Networks for High Dimension, Low Sample Size Data, 2017, pp. 2287–2293, https://doi.org/10.24963/ijcai.2017/318.

[4] G.G. Towell, J.W. Shavlik, Extracting refined rules from knowledge-based neural networks, Mach. Learn. (ISSN: 0885-6125) 13 (1) (1993) 71–101.

[5] E.B. Bartlett, Self determination of input variable importance using neural networks, Neural Parallel Sci. Comput. 2 (1994) 103–114.

[6] B. Mak, R.W. Blanning, An empirical measure of element contribution in neural networks, IEEE Trans. Syst. Man Cybern. C (ISSN: 1094-6977) 28 (4) (1998) 561–564.

[7] L.H. Gilpin, D. Bau, B.Z. Yuan, A. Bajwa, M. Specter, L. Kagal, Explaining explanations: An overview of interpretability of machine learning, 2018, arXiv:1806.00069.

[8] E. Junque de Fortuny, D. Martens, Active learning-based pedagogical rule extraction, IEEE Trans. Neural Netw. Learn. Syst. 26 (2015).

[9] V. Chan, C.W. Chan, Development and application of an algorithm for extracting multiple linear regression equations from artificial neural networks for nonlinear regression problems, in: 2016 IEEE 15th International Conference on Cognitive Informatics Cognitive Computing (ICCI*CC), 2016, pp.479–488.

[10] J.O. Garcia, A. Ashourvan, S. Muldoon, J.M. Vettel, D.S. Bassett, Applications of community detection techniques to brain graphs: Algorithmic considerations and implications for neural function, Proc. IEEE 106 (5) (2018) 846–867.

[11] J. Liu, M. Li, Y. Pan, W. Lan, R. Zheng, F.-X. Wu, J. Wang, Complex brain network analysis and its applications to brain disorders: A survey, Complexity 2017 (2017) 1–27, https://doi.org/10.1155/2017/8362741.

[12] V.A.C. Horta, A. Mileo, Towards explaining deep neural networks through graph analysis, in: G. Anderst-Kotsis, A.M. Tjoa, I. Khalil, M. Elloumi, A. Mashkoor, J. Sametinger, X. Larrucea, A. Fensel, J. Martinez-Gil, B. Moser, C. Seifert, B. Stein, M. Granitzer (Eds.), Database and Expert Systems Applications, Springer International Publishing, Cham, ISBN: 978-3-030-27684-3, 2019, pp. 155–165.

[13] M. Sandler, A.G. Howard, M. Zhu, A. Zhmoginov, L.-C. Chen, MobileNetV2: Inverted residuals and linear bottlenecks, in: CVPR, IEEE Computer Society, 2018, pp. 4510–4520, URL http://dblp.uni-trier.de/db/conf/cvpr/cvpr2018.html#SandlerHZZC18.

[14] A. Krizhevsky, Learning Multiple Layers of Features from Tiny Images, Technical Report, 2009.

[15] T. Hailesilassie, Rule extraction algorithm for deep neural networks: A review, 2016, arXiv:1610.05267.

[16] D. Kim, J. Lee, Handling continuous-valued attributes in decision tree with neural Network Modelling, in: ECML, 2000.

[17] R. Krishnan, G. Sivakumar, P. Bhattacharya, A search technique for rule extraction from trained neural networks, Pattern Recognit. Lett. (ISSN: 0167-8655) 20 (3) (1999) 273–280.

[18] M.H. Mohamed, Rules extraction from constructively trained neural networks based on genetic algorithms, Neurocomputing (ISSN: 0925-2312) 74 (17) (2011) 3180–3192.

[19] G. Bologna, A simple convolutional neural network with rule extraction, Appl. Sci. 9 (12) (2019) 2411, https://doi.org/10.3390/app9122411.

[20] K. Simonyan, A. Vedaldi, A. Zisserman, Deep inside convolutional networks: Visualising image classification models and saliency maps, 2013, CoRR abs/1312.6034, URL http://arxiv.org/abs/1312.6034.

[21] D. Erhan, Y. Bengio, A. Courville, P. Vincent, Visualizing Higher-Layer Features of a Deep Network, Technical Report, Univeristé de Montréal, 2009.

[22] S. Carter, Z. Armstrong, L. Schubert, I. Johnson, C. Olah, Activation atlas, Distill (2019) https://doi.org/10.23915/distill.00015, https://distill.pub/2019/activation-atlas.

[23] F. Grün, C. Rupprecht, N. Navab, F. Tombari, A taxonomy and library for visualizing learned features in convolutional neural networks, 2016, arXiv preprint arXiv:1606.07757.

[24] Z. Qin, F. Yu, C. Liu, X. Chen, How convolutional neural network see the world - A survey of convolutional neural network visualization methods, 2018, arXiv:1804.11191.

[25] A. Chatzimparmpas, R.M. Martins, I. Jusufi, A. Kerren, A survey of surveys on the use of visualization for interpreting machine learning models, Inf. Vis. (2020) 147387162090467, https://doi.org/10.1177/1473871620904671.

[26] D. Garcia-Gasulla, A. Vilalta, F. Parés, J. Moreno, E. Ayguadé, J. Labarta, U. Cortés, T. Suzumura, Building graph representations of deep vector embeddings, 2017, CoRR abs/1707.07465, URL http://arxiv.org/abs/1707.07465.

[27] D. Garcia-Gasulla, A. Vilalta, F. Parés, J. Moreno, E. Ayguadé, J. Labarta, U. Cortés, T. Suzumura, An out-of-the-box full-network embedding for convolutional neural networks, in: 2018 IEEE International Conference on Big Knowledge (ICBK), 2018, pp. 168–175.

[28] Y. LeCun, C. Cortes, MNIST handwritten digit database, 2010, URL http://yann.lecun.com/exdb/mnist/.

[29] H. Xiao, K. Rasul, R. Vollgraf, Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms, 2017, arXiv:cs.LG/1708.07747.

[30] V. Blondel, J.-L. Guillaume, R. Lambiotte, E. Lefebvre, Fast unfolding of communities in large networks, J. Stat. Mech. Theory Exp. 2008 (2008) https://doi.org/10.1088/1742-5468/2008/10/P10008.

[31] L. Page, S. Brin, R. Motwani, T. Winograd, The PageRank citation ranking: Bringing order to the Web, in: Proceedings of the 7th International World Wide Web Conference, Brisbane, Australia, 1998, pp.161–172, URL http://ilpubs.stanford.edu:8090/422/.

[32] M. Jacomy, T. Venturini, S. Heymann, M. Bastian, ForceAtlas2, a continuous graph layout algorithm for handy network visualization designed for the gephi software, in: M.R. Muldoon (Ed.), PLoS One 9 (6) (2014) e98679, https://doi.org/10.1371/journal.pone.0098679.

[33] Z. Ding, X. Zhang, D. Sun, B. Luo, Overlapping community detection based on network decomposition, Sci. Rep. 6 (1) (2016) https://doi.org/10.1038/srep24115.

[34] V. Horta, V. Ströele, R.M.M. Braga, J.M.N. David, F. Campos, Analyzing scientific context of researchers and communities by using complex network and semantic technologies, Future Gener. Comput. Syst. 89 (2018) 584–605, https://doi.org/10.1016/j.future.2018.07.012.

[35] C. Olah, A. Satyanarayan, I. Johnson, S. Carter, L. Schubert, K. Ye, A. Mordvintsev, The building blocks of interpretability, Distill (2018) https://doi.org/10.23915/distill.00010, https://distill.pub/2018/building-blocks.

[36] A. Krizhevsky, V. Nair, G. Hinton, CIFAR-100 (Canadian Institute for Advanced Research), URL http://www.cs.toronto.edu/~kriz/cifar.html.

[37] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, ImageNet: A large-scale hierarchical image database, in: CVPR09, 2009.

[38] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C.L. Zitnick, P. Dollár, Microsoft COCO: Common objects in context, 2014, arXiv:1405.0312.

Mr Vitor A. C. Horta is a full-time Ph.D. student at Insight Centre for Data Analytics at Dublin City University in Ireland. His research is focused on Knowledge Representation and Deep Learning and one of the main goals is to explain the whole decision making process used by Deep Learning models. Previously, he got his B.Sc. and M.Sc. from the Federal University of Juiz de Fora, Brazil. During the M.Sc. he conducted research in the fields of Social Network Analysis and Semantic Web and Applications.

Ms Ilaria Tiddi is a Research Associate at the Vrije Universiteit Amsterdam, working for the Knowledge Representation and Reasoning group as well as the Amsterdam Cooperation Lab.

Her research focuses on creating transparent, intelligent systems that generate explanations through a combination of machine learning, semantic technologies, open data and cognitive theories, mostly applied in e-Science and robotics scenarios.

Dr Suzanne Little is an Associate Professor in the School of Computing at Dublin City University, Ireland and an SFI Principal Investigator at the Insight SFI Research Centre for Data Analytics working in the area of media analytics, information access and retrieval across a variety of application domains. She completed her Ph.D. at the University of Queensland, Australia in 2006 developing tools for analysing and managing scientific multimedia data and has since worked on a number of EU projects in the areas of multimedia, technology enhanced education, security, autonomous vehicles and big data in Italy, Germany and the UK. She is currently involved in projects using multimedia analytics for instrumented vehicles, smart cities and biomedical devices and, as co-director of the SFI Centre for Research Training in Artificial Intelligence, works with Ph.D. students across a diverse range of AI applications.

Dr. Alessandra Mileo is a tenured Assistant Professor in the School of Computing and Funded Investigator at the Insight Centre for Data Analytics and at the Advanced Manufacturing Research Centre (I-FORM), Dublin City University. In the past few years, Dr. Mileo has developed a research programme in Knowledge Representation and Stream Reasoning, leveraging Semantic Technologies, expressive reasoning and statistical relational learning to design new approaches for scalable complex reasoning in dynamic environments. More recently she has developed a new research programme focused on neural-symbolic computation and knowledge-enhanced Deep Representation Learning. Dr. Mileo has secured almost 1 million euros in funding including national (SFI, IRC), international (EU, NSF) and industry-funded projects, publishing 60+ papers in high impact conferences and journals and being an active PC member of over 20 top-ranked conferences and high-impact journals. She is also a member of the W3C, the AAAI, the Association for Logic Programming, the Italian Association for Artificial Intelligence (AI*IA), the Web Reasoning and Rule Systems Association (RRA) among others.