Neural-Symbolic Integration of Knowledge Extraction and Reasoning on Graph Data*

Luisa Werner, Nabil Layaïda, and Pierre Genèves

Tyrex team, Univ. Grenoble Alpes, CNRS, Inria, Grenoble INP, LIG, 38000 Grenoble, France
https://tyrex.inria.fr

Abstract. In order to extend the Neural-Symbolic approach Knowledge Enhanced Neural Networks, we investigate its replicability and present a re-implementation of Knowledge Enhanced Neural Networks based on the Graph Neural Network framework PyTorch Geometric. Knowledge Enhanced Neural Networks integrate prior knowledge in the form of logical formulas into an Artificial Neural Network by adding additional Knowledge Enhancement layers. The obtained results show that the model outperforms pure neural models as well as Neural-Symbolic models. To ensure that our implementation produces the same results, we replicated the original transductive experiments and explained the various challenges and the steps that we went through to reach that goal. Our long term goal is to be able to address more complex and large-scale knowledge graphs and to benefit from the wide range of functionalities available in PyTorch Geometric.

Keywords: Knowledge Enhanced Neural Networks · Neural-Symbolic Integration · Fuzzy Logic · Relational Learning · Reproducibility

1 Neural-Symbolic Integration

Over the last few years, much progress has been made in the field of Deep Learning. Many remarkable results in several research domains (Natural Language Processing, Computer vision, Game playing etc.) could be achieved thanks to the application of neural network (NNs). Their success comes from ability to discover features in high-dimensional data without human intervention or expert knowledge. At the same time deep learning has some major drawbacks. Neural networks are often called *black box models* since they store all knowledge in the network parameters which are often difficult to interpret. Besides, the training of NNs is often time and data intensive. Deep learning methods are also exposed to a high risk of overfitting and often fail to generalise. A lot of concerns about interpretability, reliability and accountability exist, particularly in safety-sensitive domains. With regard to *zero short learning* which describes the ability to recognise objects not previously seen at training, NNs have a strong

* Supported by MIAI Knowledge Communication and Evalution Chair

need for improvement. Besides deep learning, also called sub-symbolic learning, a paradigm called symbolic AI coexists. Symbolic AI carries out a series of logic-like reasoning steps over language-like representations. Data efficiency, , transfer learning and generalisation capacity are the strength of symbolic AI approaches. However, they are often not robust to exceptions, errors or noise. Symbolic AI aims at tackling the shortcomings of Deep Learning. Neural-symbolic approaches aim to bring together the best of both worlds and obtain models that are interpretable, data efficient, generalise well and transfer knowledge. Neural-Symbolic systems seek to benefit from knowledge representations and reasoning capacities of logical symbolic representations and the robust learning capacities of neural networks, reconciling the logical nature of reasoning and the statistical nature of learning [3].

2 Knowledge Enhanced Neural Networks

Knowledge Enhanced Neural Network (KENN) [2] is a Neural-Symbolic approach that stacks Knowledge Enhancers (KEs) as additional layers on top of a NN. These layers modify the outputs of the so-called Base Neural Network (Base NN) with respect to some prior knowledge in the form of first-order logic formulas. Both the KEs as well as the Base NN are designed to be fully differentiable and are optimized jointly with backpropagation. The KEs contain learnable *clause weights* that are modified during training and allow the model to be robust to wrong knowledge by setting the respective weights to zero. While KENN was initially developed for Multi-Label Classification [2], an extension of the model to relational domains was proposed [1]. In order to evaluate the performance of KENN on relational data, the model is applied to a Collective Classification task on the Citeseer data set. The experiments show that the proposed model is not only able to outperform the standalone Base NN but also Neural-Symbolic models. Furthermore, when limited training data is available, the use of prior knowledge leads to considerable performance gains. KENN [1] was recently extended to relational domains.

In a classification setting the Base NN takes features $x \in R^n$ as input and calculates outputs $y \in R^m$ for m classes. The KEs are stacked on top of the Base NN and modify the predictions with respect to the satisfaction of the prior knowledge. The prior knowledge is specified as a set of clauses in first-order logic that are formulated as disjunction of literals and refer to the output classes. The logical language contains constants, variables and predicates. Unary predicates express properties of constants, while binary predicates describe relations between two constants. To give an example from the Citeseer data set, let the unary predicate $AI(x)$ describe that a scientific paper belongs to the document class AI and let the binary predicate $Cite(x, y)$ indicate a citation between two papers x and y. The clause

$$\forall x, y : AI(x) \wedge Cite(x, y) \longrightarrow AI(y)$$

describes that two papers belong to the same class AI if they cite each other. A KE revises the predictions y of the Base NN with respect to the satisfaction of the clauses in the prior knowledge and returns updated predictions y'. The KE contains *clause weights* which are modified during training together with the trainable parameters of the Base NN. The clause weight w_c of a clause c quantifies the importance of a specific clause on the input and can be robust to wrong knowledge by being set to zero.

To be incorporated in a neural architecture, the knowledge has to be interpreted in a real-valued domain. Fuzzy Logic is applied to obtain values in a continuous interval of $[0, 1]$. The predictions of the Base NN are used as interpretation of unary predicates in the numeric domain. T-conorm functions map the truth values of grounded atoms to the truth value of a clause and are defined as follows:

$$\perp : [0, 1] \times [0, 1] \longrightarrow [0, 1] \tag{1}$$

To quantify the improvement of the satisfaction of a clause, the concept of a *t-conorm boost function (TBF)* that updates initial predictions, is specified as:

$$\delta : [0, 1]^n \rightarrow [0, 1]^n \tag{2}$$

In KENN, the following function is chosen as TBF and applied to the preactivation $\mathbf{v}_i$ of the Base NN.

$$\delta_s^{w_c}(\mathbf{v})_i = w_c \cdot softmax(\mathbf{v})_i \tag{3}$$

A module called *Clause Enhancer (CE)* implements the TBF for each clause. The CE selects the literals concerned by a clause from the preactivations of the Base NN and then applies the TBF of Equation (3) to obtain the changes. The updates introduced by all CEs are aggregated, added to the preactivations and fed into a logistic function. Various groundings of a clause to constants are stored in a matrix format where columns represent the predicates and rows the objects. Once the CE is instantiated, it works on several groundings of a clause.

When relational data is considered, not only unary predicates but also binary predicates are taken into account. In order to be applicable to relational data, some architectural modifications of KENN are required. When a clause contains multiple variables, the same grounded atom may occur in multiple groundings of the clause. Considering the clause $c : \neg AI(x) \vee \neg Cite(x, y) \vee AI(y)$ of the previous section and the two groundings $c[x/a, y/b]$ and $c[x/b, y/c]$ for example, the same grounded $AI(b)$ has to be enhanced. This leads to complications because changes proposed by the same CE are not aggregated in non-relational KENN. Consequently, adaptions of the data representation are required. In the relational version of KENN [1], the knowledge is split into a set of *unary clauses* that contain only unary predicates and a set of *binary clauses* that contain binary or lower arity predicates. The KE is applied on unary and binary clauses separately before the changes are added to the preactivations. While the groundings of unary predicates can be represented as a matrix $\mathbf{U}$ that contains the objects as rows and predicates as columns, the keys of the binary predicates are two-dimensional. Consequently, binary predicates are represented as a matrix $\mathbf{B}$ that has as many rows as number of edges in a graph and columns as binary predicates. To enhance binary clauses, all predicates of the clause have to be presented together in one matrix on which the CE can be applied. Hence, unary predicates are extended to binary predicates by ignoring one component of the input. Given a unary predicate $P(x)$ for example, it can be extended to two binary predicates $P^X(x,y)$ and $P^Y(x,y)$. Consequently, the relational KENN contains a Join Layer that puts binary predicates and the binarized unary predicates in one matrix on which the CE can operate. After obtaining the changes, a Group-By Layer extracts the changes that apply to the same grounded atom and aggregates them. By representing unary and binary predicates in a single matrix the Knowledge Enhancement on relational domains is made possible.

3 On the Replicability of Knowledge Enhanced Neural Networks in a GNN Framework

As new results in Artificial Intelligence (AI) are increasingly derived through experiments on data, the reproducibility of experiments is crucial to obtain more transparent, independent and reliable research results. According to the definitions in [5], we define a result as *reproducible* if one can take the original code, re-execute it and obtain the reported results. Results are called *replicable* if one can create a new implementation that matches the conceptual and algorithmic descriptions given the article and obtain (or approximately obtain) the communicated results. The replication of results is a way to ensure that the conceptual descriptions and results of the literature are well interpreted. It is also a possibility to guarantee that the results are robust and transparent.

The first experiments with KENN show that it represents a promising Neural-Symbolic approach that has a enormous potential for extension. The current scope of the results is however limited. In the future, we intend to extensively investigate the applicability of KENN to larger knowledge graphs, the injection of more complex prior knowledge and the interaction with Graph Neural Networks (GNNs).

In our paper under review 'On the Replicability of Knowledge Enhanced Neural Networks in a Graph Neural Network Framework' [4] we represented a new interpretation and implementation of relational KENN based on the GNN framework PyTorch Geometric [7]. We use the conceptual description of relational KENN [1] and the original code written using Tensorflow 2 and Keras. In order to ensure that the conceptual descriptions are well understood, we replicated the transductive experiments on the Citeseer data set. The main contributions of this work were (1) a new interpretation and implementation of relational KENN based on PyG and (2) the replication of the experiments reported in [1]. The replication of results turned out to be a more challenging task than initially thought. In contrast to reproduction, replication requires a full understanding of the proposed concepts together with a detailed inspection of all components of the accompanying experiment. Beyond this, replication can reveal conceptual limitations, bugs, ambiguities. It can also give more insights on potential improvements and extensions. With our work, we demonstrated that the results with relational KENN on Citeseer can be achieved in two different frameworks. The future enhancements of relational KENN with our implementation thus have a solid and reliable foundation.

4 Extension of KENN to Large-scale Graphs

Based on our implementation of relational KENN in PyTorch based on PyTorch Geometric, we aim to apply the model to large-scale graphs. The Open Graph Benchmark (OGB) [6] is a well-known collection of large-scale datasets for graph learning. In this context we want to investigate the applicability and feasibility of the concepts of KENN on large-scale graphs. Furthermore, the data sets in OGB contain not only homogeneous graphs but also more informative heterogeneous graphs. A *homogeneous* graph contains only one node and edge type, while a *heterogeneous* graphs contains more than one node or edge type. With heterogeneous graphs, more complex prior knowledge can be formulated. The application of KENN to large-scale and heterogeneous graphs in OGB is ongoing work. For our studies, the data sets for node classification are particularly interesting. OGB contains five data sets of three different domains that are suitable for node classification

- OGBN-products: Amazon purchasing network
- OGBN-arxiv: citation network (extracted from Microsoft Academic Graph (MAG))
- OGBN-mag: citation network (extracted from MAG)
- OGBN-papers100m: citation network (extracted from MAG)
- OGBN-proteins: protein-protein association network

On top of that, in prior experiments KENN was only used with a simple base NN that neglected the edges between nodes. We investigate the usage of a GNN as base NN that incorporates edges. Since a GNN as standalone model already performs the standalone base NN used in [1], the enhancement of the Graph Neural Network might lead to further gains in performance.

References

1. Daniele, A., Serafini, L.: Neural Networks Enhancement with Logical Knowledge. In: arXiv, (2020). https://doi.org/10.48550/ARXIV.2009.06087
2. Daniele, A. and Serafini, L.: Knowledge Enhanced Neural Networks. In: PRICAI 2019: Trends in Artificial Intelligence: 16th Pacific Rim International Conference on Artificial Intelligence, (2019). https://doi.org/= 10.1007/978-3-030-29908-$8_4$3
3. Susskind, Zachary and Arden, Bryce and John, Lizy K. and Stockton, Patrick and John, Eugene B, L.: Neuro-Symbolic AI: An Emerging Class of AI Workloads and their Characterization. In: arXiv, (2021). https://doi.org/https://arxiv.org/abs/2109.06133
4. Werner, L. and Layaïda, N. and Genèves, P.: On the Replicability of Knowledge Enhanced Neural Networks in a Graph Neural Network Framework. https://hal.inria.fr/hal-03692362 (2022).

5. Weihua Hu and Matthias Fey and Marinka Zitnik and Yuxiao Dong and Hongyu Ren and Bowen Liu and Michele Catasta and Jure Leskovec: Open Graph Benchmark: Datasets for Machine Learning on Graphs. In: CoRR, (2020). https://doi.org/https://arxiv.org/abs/2005.00687
6. Fey, Matthias and Lenssen, Jan E.: Fast Graph Representation Learning with PyTorch Geometric. In: ICLR Workshop on Representation Learning on Graphs and Manifolds (2019).