

Connectionist Models For Formal Knowledge Adaptation

Ilianna Kollia, Nikolaos Simou,
Giorgos Stamou and Andreas Stafylopatis

Department of Electrical and Computer Engineering,
National Technical University of Athens,
Zographou 15780, Greece
nsimou@image.ntua.gr

Abstract. Both symbolic knowledge representation systems and artificial neural networks play a significant role in Artificial Intelligence. A recent trend in the field aims at interweaving these techniques, in order to improve robustness and performance of classification and clustering systems. In this paper, we present a novel architecture based on the connectionist adaptation of ontological knowledge. The proposed architecture was used effectively to improve image segment classification within a multimedia application scenario.

1 Introduction

Intelligent systems based on symbolic knowledge processing, on the one hand, and artificial neural networks, on the other, differ substantially. Nevertheless, they are both standard approaches to artificial intelligence and it is very desirable to combine the robustness of neural networks with the expressivity of symbolic knowledge representation. This is the reason why the importance of the efforts to bridge the gap between the connectionist and symbolic paradigms of artificial intelligence has been widely recognised. As the amount of hybrid data containing symbolic and statistical elements, as well as noise, increases, in diverse areas, such as bioinformatics, or text and web mining, including multimodal application scenarios, neural-symbolic learning and reasoning becomes of particular practical importance. Notwithstanding the progress in this area, this is not an easy task. The merging of theory (background knowledge) and data learning (learning from examples) in neural networks has been indicated to provide learning systems that are more effective than purely symbolic and purely connectionist systems, especially when data are noisy. This has contributed decisively to the growing interest in developing neural-symbolic systems [9, 5, 6, 4].

While significant theoretical progress has recently been made on knowledge representation and reasoning using neural networks, and on direct processing of symbolic and structured data using neural methods, the integration of neural computation and expressive logics is still in its early stages of methodological development [6].

Adaptation of symbolic ontological knowledge from raw data is an ideal use-case for further development and exploitation of neural-symbolic systems. Since the pioneering work of McCulloch and Pitts, a number of systems have been developed in the 80s and 90s, including Towell and Shavlik’s KBAN, Shastri’s SHRUTI, the work by Pinkas [9], Holldobler [6] and Artur S. d’Avila Garcez et al[5][4]. These systems, however, have been developed for the study of general principles, and are in general not suitable for real data or application scenarios that go beyond propositional logic. Only very recently, the theory has advanced far permitting the implementation of systems which can deal with logics beyond the propositional case [6].

This integration can be realised by an incremental workflow for knowledge adaptation. Symbolic knowledge bases can be embedded into a connectionist representation, where the knowledge can be adapted and enhanced from raw data. This knowledge may in turn be extracted into symbolic form, where it can be further used. This workflow is generally known as the neural-symbolic learning cycle, as depicted in the following diagram.

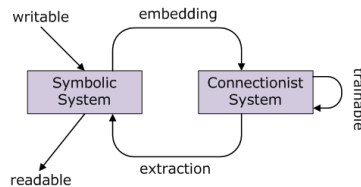


Fig. 1. The neural-symbolic learning cycle

In this paper we focus on developing connectionist adaptation of ontological knowledge, in particular of knowledge represented using expressive description logics. We then show that neural-symbolic methods can be used effectively to enhance knowledge adaptation within a multimedia application scenario. The rest of the paper is organized as follows. Section 2 presents the proposed architecture that mainly consists of the formal knowledge and the knowledge adaptation components, which are described in sections 3 and 4 respectively. Section 5 presents a multimedia analysis experimental study illustrating the theoretical developments. Conclusions and planned future activities are presented in section 6.

2 The proposed Architecture

Capitalizing these experiences our system is designated as a learning, evolving and adapting cognitive model. Starting with basic knowledge about the nature of the problem and by using powerful reasoning mechanisms the proposed system gradually attempts to evolve its knowledge. In that way it incorporates its observations along with its own or the user’s evaluation.

Figure 2 summarizes the proposed system architecture, consisting of two main components: the *Formal Knowledge* and the *Knowledge Adaptation*. The Formal Knowledge stores, the terminology and assertions, constraints that describe the problem under analysis in the appropriate knowledge representation formalism. More specifically, the *Ontologies module* formally represents the general knowledge about the problem.

It is actually a formal ontological description representing the concepts and relationships of the field, providing formal definitions and axioms that hold in every similar environment. This forms the system’s knowledge which generated during the Development Phase by knowledge engineers and experts.

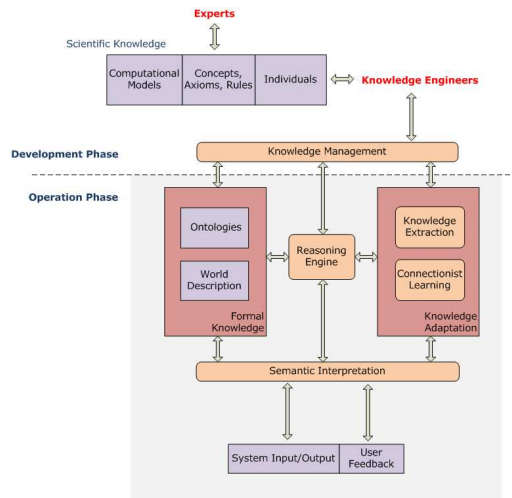


Fig. 2. The semantic adaptation architecture

Moreover, the *Formal Knowledge* contains the *World Description* that is actually a representation of all objects and individuals of the world, as well as their properties and relationships in terms of the Ontology. It is evident that most of the above data involve different types of uncertain information and, thus, they can be represented as formal (fuzzy) description logic assertions connecting the objects and individuals of the world with the concepts and relationships of the Ontology. This operation is performed by the *Semantic Interpretation* module.

In real environments however, this is a rather optimistic claim. Unfortunately, there may be lot of reasons that cause inconsistencies in the *Formal Knowledge*. For example, it is impossible to model all specific environments and thus, in some cases, conflicting assertions can arise. As a more abstract example (and more difficult to handle), the personality and expressivity of a specific user makes some of the axioms and constraints of the Ontology non-applicable or even wrong, according to logical entailments or user feedback. These inconsistencies make the formal use of knowledge that the *Reasoner* provides rather problematic. In

such cases, the *Knowledge Adaptation* component of the system tries to resolve the inconsistency through a recursive learning process.

The knowledge adaptation improves the knowledge of the system by changing the world description and to some degree the axioms of the terminology of the system. The new information as represented in a connectionist model and, with the aid of learning algorithms, is adapted and then re-inserted in the knowledge base through the *Knowledge Extraction* and the *Semantic Interpretation* module for adaptation purposes.

3 The Formal Knowledge Component

3.1 Formal (Ontological) knowledge and Connectionist models

The focus of the proposed system architecture in Figure 2 is the adaptation of the knowledge base, so as to deal with contextual information and raw data peculiarities obtained from multimodal inputs. In this paper we adopt recent results in formal knowledge representation and neural-symbolic integration. In this framework, formal knowledge is transferred to a connectionist system and is adapted by means of machine learning algorithms. Knowledge extraction from trained networks is another important issue, which is included in the neural-symbolic loop, although not studied analytically in this paper.

3.2 Kernel Definition for Description Logics

In this section recent work that extracts parameter kernel functions for individuals within ontologies is presented [3, 2, 1]. Exploitation of these kernels permits inducing classifiers for individuals in Semantic Web (OWL) ontologies. In this paper, extraction of kernel functions is the main outcome of the *Formal Knowledge* component - assisted by the reasoning engine - for feeding the connectionist-based *Knowledge Adaptation* module.

The basis for developing these functions in the framework of the formal knowledge is the encoding of similarity between individuals, as they are presented to the knowledge base of the system, by exploiting semantic aspects of the reference representations.

The family of kernel functions $k_p^F : Ind(A) \times Ind(A) \rightarrow [0, 1]$, for a knowledge base $K = \langle T, A \rangle$ consisting of the TBox T (set of terminological axioms of concept descriptions-Ontology) and the ABox A (assertions on the world state-World Description); $Ind(A)$ indicates the set of individuals appearing in A , and $F = \{F_1, F_2, \dots, F_m\}$ is a set of concept descriptions. These functions are defined as the L_p mean of the, say m , simple concept kernel functions κ_i , $i = 1, \dots, m$, where, for every two individuals a, b , and $p > 0$,

$$\kappa_i(a, b) = \begin{cases} 1 & (F_i(a) \in A \wedge F_i(b) \in A) \vee (\neg F_i(a) \in A \wedge \neg F_i(b) \in A) \\ 0 & (F_i(a) \in A \wedge \neg F_i(b) \in A) \vee (\neg F_i(a) \in A \wedge F_i(b) \in A) \\ \frac{1}{2} & \text{otherwise} \end{cases} \quad (1)$$

$$\forall a, b \in \text{Ind}(A) \quad k_p^F(a, b) := \left[\sum_{i=1}^m \left| \frac{\kappa_i(a, b)^p}{m} \right| \right]^{1/p} \quad (2)$$

The rationale of these kernels is that similarity between individuals is determined by their similarity with respect to each concept F_i , i.e, if they both are instances of the concept or of its negation. Because of the Open World Assumption for the underlying semantics, a possible uncertainty in concept membership is represented by an intermediate value of the kernel. A value of $p = 1$ has generally been used for implementing (2) in [3]. In our case, we have used the mean value of the above kernel, which is computed through high level feature relations and a normalized linear kernel which is computed through low level feature values.

3.3 The Reasoning Engine

It should be stressed that the reasoning engine, included in Figure 2, is of major importance for the whole procedure, because it assists the operation of all knowledge related components. First, during the knowledge development phase, it is responsible for enriching manual generation of concepts and relations, so that computation of the kernels in (1), (2) includes the fewest ambiguities possible, and any inconsistencies are removed from the knowledge representation. In fact (1), (2) are computed, by relating every two individuals w.r.t each concept in the knowledge base, by using the reasoning engine. In the operation phase, it interacts with the semantic interpretation layer and the connectionist system for achieving knowledge adaptation to real life environments. Both crisp and fuzzy reasoners can form this engine. In our case, we have been using the FIRE engine [12].

The FIRE system is based on Description Logic f-*SHIN* [11] that is a fuzzy extension of the DL *SHIN* [7] and it similarly consists of an alphabet of distinct concept names (**C**), role names (**R**) and individual names (**I**). The main difference of the fuzzy extended Description Logics (DL) is their assertional component. Hence, in fuzzy DLs *ABox* is a finite set of fuzzy assertions of the form $\langle a : C \bowtie n \rangle$, $\langle (a, b) : R \bowtie n \rangle$, where \bowtie stands for $\geq, >, \leq, <$, for $a, b \in \mathbf{I}$. Fuzzy representation enriches expressiveness, so a fuzzy assertion of the form $\langle a : C \geq n \rangle$ means that a participates in the concept C with a membership degree that is at least equal to n . In this case a contradiction is formed when an individual participates in a concept with a membership degree at least equal to n and at the same time with a membership degree at-most equal to l , with $l < n$.

The main reasoning services supported by crisp reasoners are *ABox consistency*, *entailment* and *subsumption*. These services are also available by FIRE together with greatest lower bound queries which incorporate the fuzzy element. Since a fuzzy *ABox* might contain many positive assertions for the same individual, without forming a contradiction, it is of interest to compute what is the best lower and upper truth-value bounds of a fuzzy assertion. For that purpose

the term of *greatest lower bound* (GLB) of a fuzzy assertion with respect to a knowledge base is defined.

The reason why we use fuzzy reasoning is that fuzzy assertional component permits more detailed descriptions of a domain. In order to compute (1), (2) the GLB reasoning service of FiRE is used, but the resulting greatest lower bound is treated crisply. In other words, if GLB for $F_i(a) > 0$, then $F_i(a) \in A$, while if GLB for $F_i(a) = 0$, then $\neg F_i(a) \in A$. As a future extension, we intend to incorporate the fuzzy element in the estimation of kernel functions using fuzzy operations like fuzzy aggregation and fuzzy weighted norms for the evaluation of the individuals.

4 The Knowledge Adaptation Mechanism

4.1 The System Operation Phase

In the proposed architecture of Figure 2, let us assume that the set of individuals (with their corresponding features and kernel functions), that have been used to generate the formal knowledge representation in the development phase, is provided, by the *Semantic Interpretation Layer*, to the *Knowledge Adaptation* component.

Support Vector Machines constitute a well known method which can be based on kernel functions to efficiently induce classifiers that work by mapping the instances into an embedding feature space, where they can be discriminated by means of a linear classifier. As such, they can be used for effectively exploiting the knowledge-driven kernel functions in (1), (2), and be trained to classify the available individuals in different concept categories included in the formal knowledge. In [3] it is shown that SVMs can exploit such kernels, so that they can classify the (same) individuals - used for extracting the kernels - accurately; this is validated by several test cases. A Kernel Perceptron is another connectionist method that can be trained using the set of individuals and applied to this linearly separable classification problem.

Let us assume that the system is in its - real life - operation phase. Then, the system deals with new individuals, with their corresponding - multimodal - input data and low level features being captured by the system and being provided through the semantic interpretation layer to the connectionist subsystem for classification to a specific concept. It is well known that due to local or user oriented characteristics, these data can be quite different from those of the individuals used in the training phase; thus they may be not well represented by the existing formal knowledge. In the following we discuss adaptation phase of the system to this local information, taking place through the connectionist architecture.

4.2 Adaptation of the Connectionist Architecture

Whenever a new individual is presented to the system, it should be related, through the kernel function to each individual of the knowledge base w.r.t a specific concept - category; the input data domain is, thus, transformed to another

domain - taking into account the semantics that have been inserted to the kernel function.

There are some issues that should be solved in this procedure. The first is that the number of individuals can be quite large, so that transporting them in different user environments is quite difficult. A Principal Component Analysis (PCA), or a clustering procedure can reduce the number of individuals so as to be capable of effectively performing approximate reasoning. Consequently, it is assumed that through clustering, individuals become the centers of clusters, to which a new individual will be related through (1), (2).

The second issue is that the kernel function in (1), (2) is not continuous w.r.t individuals. Consequently, the values of the kernel functions when relating a new individual to any existing one should be computed. To cope with this problem, it is assumed that the semantic relations, that are expressed through the above kernel functions, also hold for the syntactic relations of the individuals, as expressed by their corresponding low level features, estimated and presented at the system input. Under this assumption, a feature based matching criterion using a k-means algorithm, is used to relate the new individual to each one of the cluster centers w.r.t the low level feature vector. Various techniques can be adopted for defining the value of the kernel functions at the resulting instances. A vector quantization type of approach, where each new individual is replaced by its closest neighbor, when computing the kernel value, is a straightforward choice. To extend the approach to a fuzzy framework, weighted averages and Gaussian functions around the cluster centers are used to compute the new instances' kernel values.

In cases that classification - of the new individual - in the specific (local) environment and the specific individual characteristics or behaviour, remains linearly separable, the SVM or Kernel Perceptron are retrained - including the new individuals in the training data set, while getting the corresponding desired responses by the *User* or by the *Semantic Interpretation Layer* - thus, adapting its architecture / knowledge to the specific context and use.

In case the problem doesn't remain linearly separable, we propose to use an hierarchical, multilayer kernel perceptron, the input layer of which is identical to the trained kernel perceptron, and which is - constructively - created, by adding hidden neurons, and learning the resulting additional weights through a tractable adaptation procedure [10]. The latter is achieved through linearization of the added neurons' activation function, while taking into account both the new input/desired output data, as well as the previous knowledge and individuals. To stress, however, the importance of current training data, a constraint that the actual network outputs are equal to the desired ones, for the new individuals, is used. As a result of this network adaptation, the system will be able to operate satisfactorily within the user's environment

The problem will, in parallel, be reported back to formal knowledge and reasoning mechanism, for updating system's knowledge for the specific context, and then (off-line) providing again the connectionist module of the user with

a new, knowledge-updated, version of the system. This case is discussed in the following subsection.

4.3 Adaptation of the Knowledge Base

Knowledge extraction from trained neural networks, e.g. perceptrons, or neuro-fuzzy systems, has been a topic of extensive research [8]. Such methods can be used to transfer locally extracted knowledge to the central knowledge base. Nevertheless, the - most characteristic - new individuals obtained in the local environment, together with the corresponding desired outputs - concepts of the knowledge base, can be transferred to the knowledge development module of the main system (in Figure 2), so that with the assistance of the reasoning engine, the system’s formal knowledge, i.e., both the TBox and the ABox, can be updated, w.r.t the specific context or user.

More specifically the new individuals obtained in the local environment form an ABox A' . In order to adapt a knowledge base $K = \langle T, A \rangle$ for a defined concept F_i using atomic concepts denoted as C , we check all related concepts denoted as $R_{F_i}C_1 \dots R_{F_i}C_n$ under the specific context, i.e. in A' . Let $|R_{F_i}C_n|$ denote the occurrences of $R_{F_i}C_n \in A$, t denote a threshold defined according to the data size and $Axiom(F_i)$ denote the axiom defined for the concept F_i in the knowledge base (i.e. $Axiom(F_i) \in T$). Furthermore, we write $R_{F_i}C_n \in Axiom(F_i)$ when the concept $R_{F_i}C_n$ is used in $Axiom(F_i)$ and $R_{F_i}C_n \notin Axiom(F_i)$ when it is not used. Knowledge adaptation is made according to the following criteria:

$$|R_{F_i}C_n| = \begin{cases} 0 - t/4 & \text{If } R_{F_i}C_n \in Axiom(F_i) \rightarrow \text{Remove } R_{F_i}C_n \text{ from } Axiom(F_i) \\ t/4 - t & \text{No adaptation in } K \\ > t & \text{If } R_{F_i}C_n \notin Axiom(F_i) \rightarrow Axiom(F_i) \cup R_{F_i}C_n \end{cases} \quad (3)$$

Equation (3) implies that the related concepts with the most occurrences in A' are selected for the adaptation of the terminology, while those that are not significant are removed. Currently, the DL constructor that is used for the incorporation of the related concept, in order to adapt the knowledge base, is specified by the domain expert. Future work includes a semi-automatic selection of constructors, that will be based on the inconsistencies formed by the use of specific DL constructors for the update of the knowledge base.

5 A Multimedia Analysis Experimental Study

The proposed architecture was evaluated in solving segment classification in images and video frames from the summer holiday domain. Such images typically include persons swimming or playing sports in the beach and therefore we selected as concepts of interest for this domain the following: Natural-Person, Sand, Building, Pavement, Sea, Sky, Wave, Dried-Plant, Grass, Tree, Trunk and Ground.

Following a region-based segmentation procedure, we let each individual correspond to an image segment. The low level features used as input to the system for each individual are the MPEG-7 Color Structure Descriptor, Scalable Color and Homogeneous Texture together with the dominant color of each segment. The colours used in this case are White, Blue, Green, Red, Yellow, Brown, Grey and Black.

We used equations (1)-(2) to compute the kernel functions and transferred them to the connectionist subsystem. In that way we trained threshold (and multilayer) perceptrons to classify more than 3000 individuals (i.e., regions extracted from 500 images), regions to the above-mentioned concepts. We tested the classification performance with new segments, with results reported in Table 1. The next step was to use the improved performance of the connectionist model which forms a new ABox, in order to adapt the knowledge base. The roles used in our knowledge base are *above - of*, *below - of*, *left - of* and *right - of* that indicate the neighboring segments, and are extracted by a segmentation algorithm, included in the semantic interpretation layer. The new axioms referred to concepts *Sea*, *Sand*, *Sky*, *Tree* and *Building* using a neighbor criterion, that is the related concept in the specific context. For example, the concept *Sea* was defined as $Sea \equiv Blue \sqcap \exists below - of.Blue$. Assuming *Sea* as F_1 , then the concepts formed by the combination of spatial relations with the other concepts i.e. $\exists below - of.Blue, \exists below - of.Brown, \dots, \exists above - of.Green$, form the set of the related concepts $R_{F_1}C_1 \dots R_{F_1}C_n$.

Using the technique described in section 4.3, the relative concepts that play a significant role, according to the ABox that is formed by the connectionist model, were defined. An adapted axiom was

$$Sea \equiv Blue \sqcap (\exists below - of.Blue \sqcup \exists above - of.Brown \sqcup \exists above - of.White \sqcup \exists right - of.White \sqcup \exists left - of.White \sqcup \exists left - of.Blue \sqcup \exists right - of.Blue \sqcup \exists above - of.Blue \sqcup \exists below - of.Blue).$$

The adapted knowledge was again transferred, through (1) and (2) to the connectionist system, which was then able to improve its classification performance, w.r.t the five concepts, as shown in third column of Table 1. It is important to note that the performance obtained is similar to that provided by adaptation of the (kernel) multilayer perceptron presented in 4.2.

6 Conclusion

In this paper we presented a novel architecture based on connectionist adaptation of ontological knowledge. The proposed architecture was evaluated using a multimedia analysis experimental study presenting very promising results. Future work, includes the incorporation of fuzzy set theory in the kernel evaluation. Additionally, we intend to further examine the adaptation of a knowledge base using the connectionist architecture, mainly focusing on the selection of the appropriate DL constructors and on inconsistency handling.

Label	Regions	NN Performance		Adapted KB	
		Precision	Recall	Precision	Recall
Person	76	56.25%	47.3	56.25%	47.3%
Sand	116	75%	51.7%	83.1%	72.1%
Building	108	58.8%	37	72.7%	53.6%
Pavement	64	25%	18%	25%	18%
Sea	80	68.1%	75%	88%	79.2%
Sky	88	64.7%	50%	75.3%	64%
Wave	36	33.3%	66.6%	33.3%	66.6%
Dried Plant	64	50%	37.5%	50%	37.5%
Grass	80	52.3%	55%	52.3%	55%
Tree	92	63.1%	52.1%	71.2%	63.1%
Trunk	72	57.1%	22.2%	57.1%	22.2%
Ground	112	24.5%	53.5%	24.5%	53.5%

Table 1. Performance after the adaptation of the knowledge base

References

1. S. Bloehdorn and Y. Sure. Kernel methods for mining instance data in ontologies. In *Proceedings of the 6th International Semantic Web Conference (ISWC)*, 2007.
2. N. Fanizzi, C. d Amato, and F. Esposito. Randomised metric induction and evolutionary conceptual clustering for semantic knowledge bases. In *CIKM 07*, 2007.
3. N. Fanizzi, C. d Amato, and F. Esposito. Statistical learning for inductive query answering on owl ontologies. In *Proceedings of the 7th International Semantic Web Conference (ISWC)*, pages 195–212, 2008.
4. Artur S. Avila Garcez, K. Broda, and D. Gabbay. Symbolic knowledge extraction from trained neural networks: A sound approach. *Artificial Intelligence*, 125:155–207, 2001.
5. Artur S. Avila Garcez and G. Zaverucha. The connectionist inductive learning and logic programming system. *Applied Intelligence, Special Issue on Neural networks and Structured Knowledge*, 11:59–77, 1999.
6. P. Hitzler, S. Holldobler, and A. Seda. Logic programs and connectionist networks. *Journal of Applied Logic*, page 245272, 2004.
7. I. Horrocks, U. Sattler, and S. Tobies. Reasoning with Individuals for the Description Logic *SHIQ*. In David MacAllester, editor, *CADE-2000*, number 1831 in LNAI, pages 482–496. Springer-Verlag, 2000.
8. E. Kolman and M. Margaliot. Are artificial neural networks white boxes? *IEEE Trans. on Neural Networks*, 16(4):844–852, 2005.
9. G. Pinkas. Propositional non-monotonic reasoning and inconsistency in symmetric neural networks. In *Proceedings of the 12th International Joint Conference on Artificial Intelligence*, page 525530, 1991.
10. N. Simou, Th. Athanasiadis, S. Kollias, G. Stamou, and A. Stafylopatis. Semantic adaptation of neural network classifiers in image segmentation. pages 907–916. 18th International Conference on Artificial Neural Networks, 2008.
11. G. Stoilos, G. Stamou, J.Z. Pan, V. Tzouvaras, and I. Horrocks. Reasoning with very expressive fuzzy description logics. *Journal of Artificial Intelligence Research*, 30(8):273–320, 2007.
12. Giorgos Stoilos, Nikos Simou, Giorgos Stamou, and Stefanos Kollias. Uncertainty and the semantic web. *IEEE Intelligent Systems*, 21(5):84–87, 2006.