Logic for knowledge representation, learning, and reasoning

Luciano Serafini

Contents

1. Artificial Agents

Artificial intelligence has the main objective of developing artificial agents that are capable of (supporting humans in) taking decisions about what to achive some or more (human) goal in a given environment.

An agent can directly operate in the environment, e.g. robots, the stock marked autonomous agents, or they can suggest actions to be performed by other agents, usually humans. e.g., in recommended systems, or they can provide the necessary information for another agent, usually humans, to take decisions, e.g., agents that perform simulations about some environment.

We use the term "environment" in a very broad sense. It indicates all the relevant aspects of the context in which an agent makes decisions and operates. Examples of environments are: the ambient in which a physical robot is operating, or the worldwide Covid pandemic, a social network, or a smart city.

The environment where the agent operates is dynamic in the sense that it can change both by the effect of the actions executed by the agent or because of some factor external and independent from the agent. The environment is unpredictable in the sense that, at the moment in which we (as engineers) are designing and implementing the agent doesn't know all the possible situations in which the agent will operate.

According to the most famous book on AI Russell and Norvig 2010

> *An agent is anything that can be viewed as perceiving its environment through sensors and acting upon that environment through actuators.*

To make effective decisions in a specific situation, the agent needs sufficient knowledge about the the current situation and it has to accumulate sufficient experience on the effects of its actions, and the general laws obeyed by the environments. All this knowledge is not available when the agent is created, but it is the agent itself that should be capable to *learn knowledge by interacting with the environment*, *represent the learned knowledge in some internal structure (model)*, *perform inference (reasoning) on these models to support its decisions.*

Example 1.1. *Consider the example of a robotic agent that has to operate in a simple environment that contains four coloured blocks. With its arm, the agent can move around the blocks, stacking them one on top of the other; it can move around the environment to get close to some blocks. Furthermore it perceives the environment through an RGB camera. This simple scenario is represented in the Figure 1.*

In the following chapters, we will consider a set of methods for knowledge representation and the different types of inference/reasoning and learning methods in each type of model.

2. Knowledge Representation

Different models for representing knowledge. We concentrate on three models, Logical models, probabilistic models, and neural models, and the integration of the three.

2.1. Logical models. A logical model of an agent's knowledge of the environment is a *set of sentences* of a logical language that encodes a set of propositions

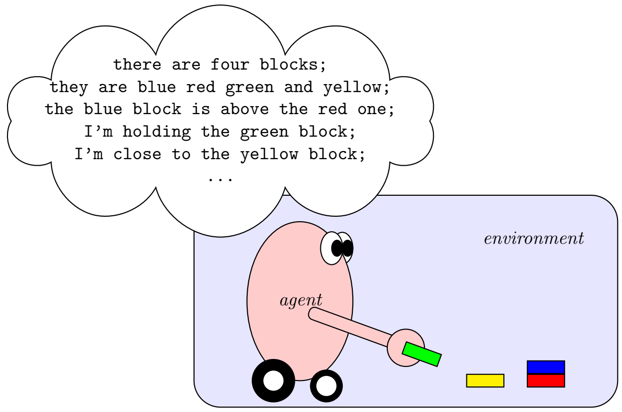

FIGURE 1.

about the environment that the agent is supposed to be true. There are many types of logical languages, e.g., propositional language, first-order language, modal language, temporal language, logic programs,... The father of logical model for AI agents is John McCarty, who, in McCarthy 1959, proposes to uses declarative logical language *"[...] to make programs*[1] *that learn from their experience as effectively as humans do"*. Successively he observed that *"In order for a program to be capable of learning something it must first be capable of being told it."*

A good summary of many different logical models is provided in the Handbook of Knowledge Representation Lifschitz, Porter, and Van Harmelen 2008. Logical models provide information of what is true, what is false and what logically follows from some premises. Minker Minker 2012 offers an overview of how logical models can be built and how they can be used for inference, decision making and planning.

2.2. Probabilistic models. Probabilistic models are particularly designed to represent uncertainty about the state of a domain. They allow associating a measure of the likelihood of a state of the world. A probabilistic model describes a domain in terms of *random variables* that can take values is some abstract domain, e.g. a finite set $\{1.2.3, \dots, n\}$, the infinite set of natural or real numbers. The different configurations of the world are represented by the assignments of the random variables with a value in their domains. Probabilistic models associate a measure of the probability to encounter such a situation (joint probability distribution). A probabilistic model provides a formal language that allows specifying the joint probability distribution. Examples of probabilistic models are explicit probabilistic models, e.g., discrete models (Uniform, Binomial, Bernoulli, Poisson) and continuous models (Normal (or Gaussian), Exponential, Dirichlet, ...), directed/undirected) graphical models Maathuis et al. 2018, (e.g, Markov random fields, Bayesian networks,...).

Inference in probabilistic models has the objective of calculating the probability that a (set of) random variables takes a (set of) values possible conditioned from some facts about the domain (observations or priors) that are known to be

[1]One can read "program" as "agents".

true. These algorithms are strictly connected to the language used to specify the probabilistic model

Learning in probabilistic models has the objective of inducing some (or all) parameters of the probabilistic model starting from a set of observations of states of the world. As for the case of inference, learning algorithms strongly depend on the formalism used to specify the probabilistic model.

2.3. Artificial Neural Networks. [From wikipedia] Artificial neural networks (ANNs), usually simply called neural networks (NNs) or neural nets are computing systems inspired by the biological neural networks that constitute animal brains.

An ANN is based on a collection of connected units or nodes called artificial neurons, which loosely model the neurons in a biological brain. Each connection, like the synapses in a biological brain, can transmit a signal to other neurons. An artificial neuron receives signals then processes them and can signal neurons connected to it. The "signal" at a connection is a real number, and the output of each neuron is computed by some non-linear function of the sum of its inputs. The connections are called edges. Neurons and edges typically have a weight that adjusts as learning proceeds. The weight increases or decreases the strength of the signal at a connection. Neurons may have a threshold such that a signal is sent only if the aggregate signal crosses that threshold.

Typically, neurons are aggregated into layers. Different layers may perform different transformations on their inputs. Signals travel from the first layer (the input layer) to the last layer (the output layer), possibly after traversing the layers multiple times.

3. Integrating Logical models with other models

In this course, we are interested also in how logical models can be integrated with probabilistic models and neural networks. There are many reasons why one wants to combine logical models with neural networks and/or probabilities.

3.1. Integrating logic and probabilistic models. Logical models provide a compact representation of a world of objects which have properties and are in relation. In other words, logical languages are very good to express structured worlds. Logical language allows the imposition of restrictions on the structure of the world. I.e., logical language can express the fact that a certain state of the world is possible, or impossible, with no intermediate degree. In probabilistic models, instead, the world is described by a flat (unstructured) set of random variables. The possible states of the world are described by an assignment to the variables, and therefore there is no structure in such a world. On the other hand, probabilistic models allow expressing that a certain configuration of the world is more likely than another.

The integration of logic and probabilistic models has the main the objective of taking the goods from both models, namely allowing probabilistic inference on structured domains. There is a large set of attempts to combine logic and probability in artificial intelligence. In this course we will introduce two of them: Probabilistic Logic Programming **.gutmann2011learning**; Lukasiewicz 1998; Kimmig et al. 2011 and Markov Logic Networks Richardson and Domingos 2006; Lowd and Domingos 2007

3.2. Integrating logic and neural networks. Logical language allows them to provide precise and crisp definitions of concepts. For instance the first-order logic formula $\forall x(\mathit{MasterStudent}(x) \leftrightarrow \mathit{Student}(x) \wedge \exists y(\mathit{MasterCourse}(y) \wedge \mathit{Enrolled}(x, y)))$ provides a precise definition of the concept of master student as a student who is enrolled in some master course. This definition is very useful when we want to find all the master students of a school by querying the database of such a school. However, the definition is not usable if we want to recognize if a person is a master student from his/her social media profile, or from some pictures of him/her. For this second task, it is much more effective to train a neural network with positive and negative examples. Combining the two types of models would offer the possibility to combine the two advantages described above.

There are many other strengths and weaknesses of neural models and logical models that justify a combination of the two approaches. They are summarized in Table 1. Let us describe them shortly.

- Neural networks are very effective to learn, from a set of positive and negative examples, how to classify an object in a certain category starting from its features from positive and negative examples. Logical models instead have some difficulties in learning general definitions from examples.
- Neural networks are good at processing objects which are described in terms of high-dimensional features. For instance, images, while in logic objects are described as atomic (abstract) entities and it is not easy to take into account numeric features associated with the objects.
- Neural network is robust to exceptions and outlier data. In logic instead, if you state that a property holds *for all* objects, then a single exception makes the model inconsistent. So it is not very adaptable to deal with exceptions.
- Neural networks are data-hungry. I.e., to train a neural network you need to provide a sufficiently large amount of training examples usually manually annotated. The process of manual annotation is very costly. In logic instead one can express a general property with just one simple formula, and even if this requires the intervention of human knowledge this does not constitute a long process.
- Inference in a neural network is a black box. It is very difficult to find an explanation of why a neural network reaches a certain conclusion. In

Strength and Weakness of	**neural network**	**logical models**
Automatic learning from raw data	⋆⋆⋆⋆⋆	⋆⋆
Dealing with hight througput data	⋆⋆⋆⋆⋆	⋆⋆
Robust to exception	⋆⋆⋆⋆	⋆
Size of training data	⋆⋆	⋆⋆⋆⋆
Explanability of inference	⋆	⋆⋆⋆
Adding commonsense knowledge	⋆	⋆⋆⋆⋆⋆
Complexity of inference	⋆⋆⋆⋆⋆	⋆⋆

TABLE 1. Comparing logical and neural models

logic instead, the inference is obtained by applying inference rules and every conclusion is associated with a *deduction or proof* (i.e., a sequence of application of inference rules) that provide the explanation of the conclusion.

- Adding commonsense knowledge, such as for instance "a cat has one tail which is opposite to his/her head") is very difficult, since it needs to be provided either through examples or by changing the architecture of the network. In logical models, instead, commonsense knowledge can be added by using a single formula that encodes such knowledge.
- Inference in neural networks is usually very fast (especially on GPUs) since it is just a matter of computing compositions of linear and non-linear functions. Inference in logic instead might be exponentially complex since it involves the search for a proof or a deduction or the desired conclusion.

In the last few years, there has been a number of proposals for combining neural networks and logical models. In this course we will present two of them: Logic Tensor Networks Badreddine et al. 2022 and Knowledge Enhanced Neural networks Daniele and Serafini 2019.

3.3. Integrating logic, neural networks and probabilities. Finally, there are methods that combine all the three types of models, neural networks, probabilistic models, and logical models. One examples that will be described in this course is DeepProbLog Manhaeve et al. 2018 that combines probabilistic logic programming with neural networks, However, there are many other approaches De Raedt et al. 2020 provides a survey.

Bibliography

Badreddine, Samy et al. (2022). "Logic tensor networks". In: *Artificial Intelligence* 303, p. 103649.

Daniele, Alessandro and Luciano Serafini (2019). "Knowledge enhanced neural networks". In: *PRICAI 2019: Trends in Artificial Intelligence: 16th Pacific Rim International Conference on Artificial Intelligence, Cuvu, Yanuca Island, Fiji, August 26–30, 2019, Proceedings, Part I 16*. Springer, pp. 542–554.

De Raedt, Luc et al. (2020). "From statistical relational to neuro-symbolic artificial intelligence". In: *arXiv preprint arXiv:2003.08316*.

Kimmig, Angelika et al. (2011). "On the implementation of the probabilistic logic programming language ProbLog". In: *Theory and Practice of Logic Programming* 11.2-3, pp. 235–262.

Lifschitz, Vladimir, Bruce Porter, and Frank Van Harmelen (2008). *Handbook of Knowledge Representation*. Elsevier.

Lowd, Daniel and Pedro Domingos (2007). "Efficient weight learning for Markov logic networks". In: *Knowledge Discovery in Databases: PKDD 2007: 11th European Conference on Principles and Practice of Knowledge Discovery in Databases, Warsaw, Poland, September 17-21, 2007. Proceedings 11*. Springer, pp. 200–211.

Lukasiewicz, Thomas (1998). "Probabilistic Logic Programming." In: *ECAI*, pp. 388–392.

Maathuis, Marloes et al. (2018). *Handbook of graphical models*. CRC Press.

Manhaeve, Robin et al. (2018). "Deepproblog: Neural probabilistic logic programming". In: *advances in neural information processing systems* 31.

McCarthy, John (1959). "Programs with Common Sense". In: pp. 77–84.

Minker, Jack (2012). *Logic-based artificial intelligence*. Vol. 597. Springer Science & Business Media.

Richardson, Matthew and Pedro Domingos (2006). "Markov logic networks". In: *Machine learning* 62, pp. 107–136.

Russell, Stuart and Peter Norvig (2010). *Artificial Intelligence: A Modern Approach*. 3rd ed. Prentice Hall.