Chapter 1

Introduction

The history of "machines that learn" is almost as old as the history of modern Computer Science. Of course, they figure prominently in Alan Turing's famous 1950 paper on Computing Machinery and Intelligence [Tur50], but proposals for connectionist-based learning appear even earlier with the development of theories emulating biological learning by McCulloch and Pitts [MP43], and parameter update by Donald Hebb [Heb49] and implementation of perceptron [Ros57], enabling the 'training' of a single-layered model. Later, the method of back-propagating errors [RHW86] in a multi-layered connectionist architecture led to its dramatic usage in recognising handwritten ZIP codes [LBD+89]. The reincarnated term, used for connectionist architectures, is *Neural Networks* or *Deep Neural Networks* or *DNNs*, in short. This field has witnessed several ground-breaking discoveries such as neural network models for learning from sequential data [HS97], deep generative model [HOT06], greedy layer-wise pre-training of deep neural networks [BLPL07]. In 2012, ImageNet classification [DDS+09] by training a deep neural network by efficiently using graphics processing units (GPUs) created a new wave in the field [KSH12], and the rest is history.

At the time of writing this dissertation, deep neural networks are undergoing an unprecedented resurgence in interest as the tools of choice in machine learning. Although many of the techniques are not new, there are at least 3 different threads that are driving recent activity "deep learning", a term that was first introduced to machine learning community by [Dec86], and to neural networks community by [Aiz99], however, the presently prevailing usage of this term is due to the work by [HOT06]: First, deep learning has become more useful as the amount of data has increased; Secondly, the significant improvements of software and hardware architectures have allowed more complex deep learning models to be trained in less time and with reasonable efficiency; Thirdly, deep learning has solved complicated applications from all domains of science and engineering with increasing predictive accuracy over time. These three threads have enabled machine-learning "in-the-large", allowing us to consider its application to problems about which we know little, but for which we are able to obtain large amounts of data (or at

least, we have large amounts of data for problems related to the one we want to solve).

This dissertation is, however, about the use of modern-day deep learning in a different setting involving knowledge-rich problems. An important example is the area of Artificial Intelligence (AI) for Science. This is concerned with the use of AI methods to accelerate our understanding of the natural world, and to assist the application of this understanding to the development of areas of engineering, medicine, healthcare, agriculture, environment and so on. An example of this is the development of Robot Scientists [KWJ+04]. In this, advanced AI machinery is coupled with robotic wet-lab hardware to execute the classic scientific "hypothesize-and-test" cycle characteristic of scientific experimentation. The Robot Scientist is able to achieve, in some measure, the following: (1) identifying the best explanation for a prediction based on what is known; (2) suggesting hidden variables or mechanisms which could improve the prediction; and (3) proposing experiments to test the hypotheses. While further ambitious plans exist for Robot Scientists for completely automating scientific discoveries [Kit16], the current use of machine learning for scientific discoveries remains as providing assistance to the scientist(s) in the loop. An example of such a collaborative system is in Figure 1.1, which intends to describe a setting where a scientist is attempting to understand some (natural) phenomena. They conduct experiments and obtain a set of observations, which are provided as data to a machine learning (ML) engine. Additionally, the scientist also provides the machine learning engine with domain-expertise that maybe relevant to understanding the phenomenon being studied.

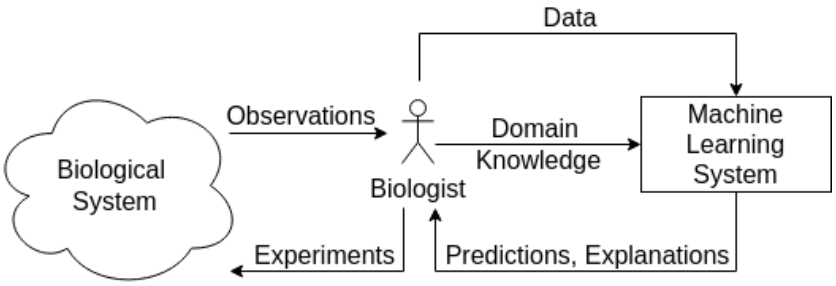

Figure 1.1: An example of using present day machine learning systems as assistance for scientific discoveries. The scientist-in-the-loop is a biologist. The biologist conducts experiments in a biological system, obtains experimental observations. The biologist then extracts data that can be used to construct machine learning model(s). Additionally, the machine learning system has access to domain-knowledge that can be obtained from the biologist. The machine learning system then conveys its predictions and explanations to the biologist.

In the diagram shown above, it is unclear how the domain information from an expert be encoded and provided while building a collaborative system. It is imperative, however, for such collaborative Human-and-AI systems to work effectively, we need at least the following: (1) We have to be able to tell the learner what we know, in a suitably precise

form; and (2) The machine has to be able to tell us what it has found, in a suitably understandable form. Point (1) concerns with some form of encoding of the available domain expertise, and point (2) concerns with some form of human-understandable explanation of the machine's prediction(s). While the remarkable recent successes of deep neural networks [Sch15] on a wide variety of tasks makes a substantial case for their use in model construction, it is not immediately obvious how either (1) or (2) should be done with deep neural networks. The research carried out in this dissertation is focused on providing some answers to (1), that is, incorporating domain information into deep neural networks. However, understanding models constructed by deep neural networks are a different area of intense research activity (see, for example, some recent surveys in this area: [Lip16, ADRDS+20]). This dissertation does not answer this aspect of research.

In the last few years, incorporating some form of domain-knowledge into learning has been emphasised strongly in the deep learning and reasoning community. Experts are stressing that incorporating domain-knowledge into deep neural networks could result in: (a) achieving highly general models and therefore could result in higher predictive performance than the models built only with available data [Mar18, BDR+20]; (b) resulting in robust reasoning systems [Mar20]. Some of these points have been investigated recently in a research report on the inclusion of domain-knowledge into deep learning [Stå21]. The report also highlights that providing domain information may not always be provided as input to a model directly but can be represented as some form of internal logic or external constraint. Furthermore, the inclusion of domain-knowledge in learning could solve some challenges in those scientific areas that require understanding data using human-machine collaboration, such as in medical diagnosis and drug discovery. All the above benefits could be achieved even if available data is scarce and the machine is provided with human-knowledge of the domain encoded in a sufficiently precise form. It is also unsurprising that a recent report on AI for Science [STN+20] identifies the incorporation of domain-knowledge as the first of the 3 Grand Challenges in developing AI systems:

> "ML and AI are generally domain-agnostic... Off-the-shelf practice treats [each of these] datasets in the same way and ignores domain knowledge that extends far beyond the raw data itself—such as physical laws, available forward simulations, and established invariances and symmetries—that is readily available... Improving our ability to systematically incorporate diverse forms of domain knowledge can impact every aspect of AI..."

Motivated by the above-discussed points, we aim to pursue this research direction, and propose new and effective methods that could help to construct robust and accurate scientific assistants. Specifically, our interest is in problems where both data and domain-knowledge are uniformly represented in first-order logic. In what follows, we first describe the importance of domain-knowledge and provide a brief overview of the approaches in

this area. We then outline some major difficulties in inclusion of domain-knowledge into deep neural networks. Next, we state the main contributions of this dissertation and provide detail on how the chapters in this dissertation are organised.

1.1 The Importance of Domain-Knowledge

Even though the idea of incorporating domain-knowledge into learning seems more pronounced recently; this is not entirely a new direction in machine learning. The earliest work in incorporating domain-knowledge into various AI methods, both symbolic and connectionist dates back to the late 1960s or early 1970s. Probably, the oldest research in this direction is by Plotkin in his doctoral thesis on "Automatic Methods of Inductive Inference" [Plo72] in which he defined generalisation from experience, relative to a body of knowledge. Michalski attempted to develop learning systems in which concepts were expressed in augmented predicate calculus. For example, Michaski's INDUCE program [Mic73, Mic80] augments the data rules input by the user with the inference rules in the domain-knowledge resulting in the construction of new rules. CONFUCIUS is a program that works with the principle that descriptions of inputs could be improved by learning domain-knowledge [CS82]. In the implementation of Marvin [SB86], a successor of CONFUCIUS, Sammut and Banerji discuss how existing (domain) concepts can be learned and re-used to learn new (domain) concepts. Here the domain concept is provided by a domain-expert (human) in an indirect fashion: the domain-expert shows the learner (Marvin) a positive example of the concept (the *target* concept) to be learned. The description of the example represents the concept which contains only one object (the shown example), and Marvin's task is to generalise the initial example in a manner that it describes all the positive examples of the target concept and none of the negative examples. Duce [Mug87] is a machine learning system that uses six transformation operators to construct high-level domain-features for a set of examples objects from their descriptions. Duce has its successor in CIGOL [MB88] where, given examples of a high-level predicate, CIGOL generates related sub-concepts to be named by a human-expert. In the process, CIGOL uses existing clauses in the database to be treated as domain-knowledge while constructing new sub-concepts. Such ideas of learning from examples and domain-knowledge comes under the umbrella term coined in 1991, called "Inductive Logic Programming (ILP)" [Mug91]. ILP provides a systematic learning method to induce hypotheses from data and background knowledge. Here data, background knowledge and hypotheses are uniformly represented in first-order logic. In the area of neural networks, knowledge-based artificial neural network (KBANN) [TSN90, TS94] is, to the best of our knowledge, the oldest method that incorporates domain-knowledge encoded as simple propositional rules.

In the next chapter, we conduct a comprehensive survey of a wide range of studies

that incorporate some form of domain-knowledge into deep neural networks. In there, we have studied that any form of domain-knowledge constrains either the structure or the parameters of a deep network. Therefore, we refer to these as "constraints". We will restrict ourselves to domain-knowledge that can be represented either as logical or as numerical constraints. Under logical constraints, we consider domain-knowledge that is represented in propositional logic, predicate logic, including binary or more generally n-ary relations, canonical normal forms, and program primitives. The numerical constraints are represented by priors on the model structure and parameters, leading to the introduction of additional terms in the loss function of the model. So, we restrict the review here to research that involves these forms of background knowledge. We note that there is a class of hybrid systems combining neural and logical systems (see for example, [GBG12, RDMM20]) that attempts to emulate logical inference or represent logical concepts. Although this hybrid approach is relevant to our present research conducted in this dissertation, incorporation of logically encoded domain-knowledge into deep neural networks is more specific than just constructing a hybrid system. We also categorise the approaches of inclusion of domain-knowledge into a deep neural network as follows: (1) transforming the data representation; (2) transforming the loss function; and (3) transforming the model (either its structure or parameters). In a sense, this progression reflects a graded increase in the complexity of changes involved. More detailed discussions on the aspects discussed here are provided in Chapter 2.

1.2 Difficulties in Inclusion of Domain-Knowledge into Deep Neural Networks

There are many difficulties associated with the inclusion of domain-knowledge into deep neural networks. While these difficulties are elaborated in our next chapter, we provide a non-exhaustive list of some of the purely implementational aspects, as follows:

- There is no standard framework for translating logical constraints to neural networks.
- Any form of logical constraint is not differentiable.
- Logical constraints can introduce cyclic dependencies if they are directly used to construct a deep neural network structure.
- The process of introducing a loss term often results in a hard optimisation problem (sometimes constrained) to be solved. Furthermore, it may require additional mathematical tools for a solution that can be implemented practically.

- Deep neural network structures constrained via logical domain-knowledge may not always be scalable to large datasets.

- While numerical constraints are introduced into the loss function or as regularisation terms, it is not straightforward and sometimes not very intuitive.

- A vast majority of the studies on incorporating some form of domain-knowledge into a deep neural network, the data considered is often represented as a numeric feature vectors or, in general, tensors. Adopting the underlying technique of learning from data that has relational structures is not straightforward. One example of a problem with high scientific value where data is relational is drug discovery, where each data instance is a molecular compound [KMSS96].

None of the difficulties listed above is concerned with the broader conceptual question of how domain-knowledge is to be acquired and represented in a machine-friendly form in the first place. In this dissertation, we will not be addressing this conceptual difficulty. Instead, we will focus on resolving some of the implementational issues for problems for which domain-knowledge is already available in some machine-readable form. In particular, we will focus our study on problems where domain-knowledge is encoded as statements in a subset of first-order logic and is about data with some relational structures.

1.3 Contributions of this Dissertation

In this dissertation, we are primarily concerned with real-world scientific problems with the following characteristics: (a) Data are naturally graph-structured (relational), (b) The amount of data available is typically small, and (c) There is significant domain-knowledge, usually expressed in some logical form (rules, taxonomies, constraints and the like). Below we outline the principal contributions made by this dissertation.

Conceptual. An approach to the stochastic selection of "relational features" as a mechanism of inclusion of domain-knowledge into multilayer perceptrons; An approach for simplified inclusion of domain-knowledge into graph-based neural networks for domain-knowledge that is represented in the form of hyperedges in a graph; An approach for complete inclusion of domain-knowledge into graph-based neural networks, through the use of ideas from mode-directed inverse entailment;

Implementational. Techniques that combine deep neural networks and symbolic representations resulting in the implementation of neuro-symbolic learners such as: Deep Relational Machines (DRMs), Vertex-Enriched Graph Neural Networks (VEGNNs),

Bottom-Graph Neural Networks (BotGNNs); and a modular end-to-end neuro-symbolic system that uses a BotGNN as a system component for generation of novel molecules for drug-design;

Applications. Investigating the applications of implementations mentioned above on large-scale carcinogenicity problems and lead-discovery problems relevant to drug design.

A more detailed breakup of the contributions is as follows:

1. We construct multilayer perceptrons (MLPs) from relational data and background knowledge using propositionalisation [LDG91], a technique in ILP that transforms relations into a simpler format, typically a feature-vector or attribute-value representation. Here, we propose a utility-based stochastic sampling method to draw relational features from a large and countable discrete feature space.

2. We propose a simplified technique called 'vertex-enrichment' for incorporating symbolic domain-knowledge into a class of deep neural networks that deal with graph-structured data, known as graph neural networks. We also demonstrate how incorporating higher-order n-ary relations discovered by ILP can further improve the predictive performance of vertex-enriched graph neural networks.

3. We propose a systematic technique to incorporate symbolic domain-knowledge into graph neural networks using the method of inverse entailment available in ILP [Mug95].

4. We construct a conditional deep generative model via the inclusion of domain-knowledge by proposing a methodology that consists of a collaborative combination of two generators and a discriminator proposed in the contribution 3 above. We study the application of this collaborative model for the problem of early-stage lead discovery in drug design.

1.4 Organisation of the Dissertation

The remainder of this dissertation is organised as follows:

In Chapter 2, we provide a review of the categories and subcategories of domain-knowledge, and various methods of their inclusion into deep neural networks (DNNs). We further examine some challenges in adopting the underlying techniques to deal with relational data and symbolic domain-knowledge.

In Chapter 3, we construct DNNs by the standard method of propositionalisation of relational features, constructed using ILP using data and background knowledge. Each

relational feature encodes some logical description of the data instance. The propositionalisation technique transforms a set of relational features into a simple format such as numeric feature vectors that can be used as inputs for a deep neural network. We propose a guided sampling strategy to sample relational features from a large countable discrete feature space. The class of DNNs investigated in this chapter are called Multilayer Perceptrons (MLPs).

Chapter 4 proposes a simplification method called "vertex-enrichment" to incorporate symbolic domain-knowledge into deep neural networks suitable for graph-structured (relational) data, called graph neural networks (GNNs). In this work, each domain relation is treated as a hyperedge. The proposed method of vertex-enrichment allows enriching the feature associated with every vertex of the graph that appears in a hyperedge. Here we will show that the vertex-enrichment technique results in simplified inclusion of domain-knowledge into GNNs.

In Chapter 5, we propose a general technique for the inclusion of multi-relational domain-knowledge into GNNs, using the method of inverse entailment [Mug95] developed within the area of Inductive Logic Programming (ILP). ILP constructs a bottom-clause (the most specific explanation about a data instance) from data and background knowledge. Here, we construct an equivalent graph representation corresponding to a bottom-clause, and show how a GNN can operate on this graph. The technique proposed here provides a framework for the complete inclusion of relational information into a GNN.

In Chapter 6 we extend our study in Chapter 5 to construct a deep generative model for molecule generation. That is, we propose a modular system that uses two deep generative models and a GNN-based discriminator constructed in Chapter 5. The technique proposed here allows indirect inclusion of relational information in the domain-knowledge into the conditional generation of novel molecules.

In all the chapters, we would aim to investigate our primary hypothesis that the inclusion of domain-knowledge improves the performance of deep neural networks. To this end, our investigation would be in the broad of drug discovery. Specifically, we conduct large-scale empirical testing of the resulting DNN models in the Chapter 3 to Chapter 5, using nearly 75 datasets that consists of over 200,000 relational data instances and with domain-knowledge containing about 100 relations. In Chapter 6, we evaluate our constructed system for conditional generation of novel small molecules to act as inhibitors for a well-studied target protein. We outline the main findings of this dissertation in Chapter 7 and provide details of future research directions that could spin out from our present work.

Some additional details concerning the technical background, such as DNNs, ILP, are provided in Appendix A. In Appendix B we provide details of the additional experiments that are conducted to evaluate some secondary research questions in our chapters.