Deep Relational Machines

Huma Lodhi

School of Engineering, University of Warwick, Coventry, CV4 7AL, UK
huma.m.lodhi@gmail.com

Abstract. Deep learning methods that comprise a new class of learning algorithms give state-of-the-art performance. We propose a novel methodology to learn deep architectures and refer to it as a deep relational machine (DRM). A DRM learns the first layer of representation by inducing first order Horn clauses and the successive layers are generated by utilizing restricted Boltzmann machines. It is characterised by its ability to capture structural and relational information contained in data. To evaluate our approach, we apply it to challenging problems including protein fold recognition and detection of toxic and mutagenic compounds. The experimental results demonstrate that our technique substantially outperforms all other approaches in the study.

Keywords: Deep Relational Learning, Restricted Boltzmann Machines, Inductive Logic Programming, Multi-class classification

1 Introduction

Deep learning (DL) is an emerging state-of-that-art class of algorithms, and whose well-known examples are deep belief networks (DBNs) [1] and stacked auto encoders (SAEs) [2]. In DL, a machine learns many layers of representation to capture structure and variations in data, and this process generally improves its generalisation performance.

The standard DL techniques operate on input features (derived from raw input data) that are in binary or numeric form. The data generated in many real-world problems, and especially in biological and chemical domains, is naturally relational. The conversion of data into a form that is amenable to such techniques may loose important information during this process. This establishes a need to propose a novel DL methodology that can handle arbitrary forms of data. First order logic (FOL) provides a useful means to represent data and is well known for its expressive powers. McCarthy who has pioneered the application of logic to construct intelligent machines has observed "If one wants a machine to be able to discover an abstraction, it seems most likely that the machine must be able to represent this abstraction in some relatively simple way", and FOL has been viewed as a means to increasing generality, and expressing knowledge and reasoning [3]. In FOL we can efficiently express both complex information that is needed for solving complex tasks like chess problems and

commonsense knowledge that is required to do a simple task, hence in this paper we take a logicist approach for learning deep relational architectures, namely, deep relational machines (DRMs).

A DRM learns a deep architecture by stacking a set of induced logical rules and any number of generative models. It learns the first layer of representation by using an Inductive Logic Programming (ILP) system. A set of hypothesised rules are used as an input for the second layer, where a restricted Boltzmann machine (RBM) is used for learning the second and successive layers. Once a suitable representation is obtained, classification or regression problems are solved by using support vector machines (SVMs) in conjunctions with the features learned at the highest layer. In DRM the use of first order logic provides the flexibility of using arbitrary forms of structured and non-structured data, and it exploits semantic and syntactic relational structures and variations to train a machine with low error probability.

ILP has been used to construct features and this process is termed as propositionalisation [4]. Recently, features (rules) generated by ILP algorithms have been used to design logic based kernel methods [5, 6]. The design of a DRM and its objectives are fundamentally different from these apparently related approaches. It is based on the belief that the representation learned by an ILP system does not capture all the structure and variations in data, and much needed refinement is achieved by performing deep learning.

We evaluate the proposed methodology by applying it to biological and chemical domains, namely protein fold recognition and identification of toxic and mutagenic compounds. The experimental results show that our approach provides accurate solutions to complex problems and generally outperforms related approaches.

2 Generative Models and Inductive Logic Programming

Generative Models: RBM [7] is a well known example of generative models. It displays a bipartite structure and consists of a single hidden layer where there are connections between the units of visible and hidden layers but there are no connections between the units of any individual layer. The connections between the units are weighted and there is also a bias associated with each unit. In its standard form, the hidden and visible units of an RBM are of the form $(0, 1)$. The probability over the joint configuration of hidden and visible units is given by $p(\mathbf{d}, \mathbf{h}) = \frac{exp\Big(-\mathcal{E}\big(\mathbf{d},\mathbf{h}\big)\Big)}{Z}$, where $\mathbf{d}$ and $\mathbf{h}$ are input and hidden vectors. Z is a normalization constant described by the expression: $Z = \sum_{\mathbf{d},\mathbf{h}} exp\Big(- \mathcal{E}\big(\mathbf{d}, \mathbf{h}\big)\Big)$. $\mathcal{E}$ is an energy function and is defined by $\mathcal{E}(\mathbf{d}, \mathbf{h}) = -\mathbf{c}^\intercal\mathbf{h} - \mathbf{b}^\intercal\mathbf{d} - \mathbf{h}^\intercal\mathbf{W}\mathbf{d}$. In this expression the vector $\mathbf{h}$ gives the hidden layer activations and the activations of visible units (input data) is supplied by $\mathbf{d}$. The entries of vector $\mathbf{b}$ are indexed by the biases of visible units and $\mathbf{c}$ is a bias vector for hidden layer. $\mathbf{W}$ is $N * M$ matrix where N represent numbers of visible units

and the number of hidden units is denoted by M. The training of an RBM can be performed by maximizing the log-likelihood using a gradient ascent scheme. As the computations can become intractable, an approximation procedure is used. In this method, data is sampled back and forth and error is minimized.

Inductive Logic Programming: The class of learning algorithms that form ILP are well known for representing relations and structures in data. In ILP first order logic is used to express data, background knowledge (prior knowledge) and the induced hypothesis. The standard learning framework, within which most ILP algorithms are designed, is learning from entailment. In this setting, we consider a training set $D = \{(\mathbf{d}_1, t_1), (\mathbf{d}_2, t_2), \ldots, (\mathbf{d}_n, t_n)\}$ of input-output pair examples and background knowledge B. Each $\mathbf{d}_i$ belongs to a domain $\mathcal{D}$ and each t_i belongs to the set T, where $T = \{0, 1\}$. All the examples having $t_i = 1$ form the positive training set D^+, and the negative training set D^- comprises all the examples having $t_i = 0$. An ILP algorithm aims to induce a hypothesis (set of Horn clauses, rules) $\mathfrak{F}$ from background knowledge, B, and data D with the aim that it does not imply or more specifically entail any negative example, and all the positive examples satisfy the conditions of the hypothesis. Formally, $B \cup \mathfrak{F} \models D^+$ and $B \cup \mathfrak{F} \not\models D^-$.

In this paper we use the symbol $\mathfrak{F}$ for a hypothesis (typically a set of a few rules) returned by a standard ILP algorithm, the symbol $\mathbb{F}$ for a hypothesis that is comprised of all the R rules generated during the search of the hypothesis space and symbol F for a hypothesis that is given by a set of N rules that describe the first layer of a DRM.

3 Deep Relational Machines

The Logical Layer: We build the first layer by adopting the learning from entailment setting and term it the logical layer. The data D comprising positive and negative examples are encoded as ground facts. The background knowledge B is given by a set of Horn clauses where the set comprises ground facts or non-ground clauses. A hypothesis F, in the form of Horn Clauses (rules), is induced. The data is transformed for the next layer by determining whether an example $\mathbf{d}_i$ is implied by the hypothesis F conjoined with the background knowledge. A rule $f_j \in F$ conjoined with background knowledge B is viewed as an encoding function $E(\mathbf{d}, B, F)$ that maps data into the Boolean representation as described below:

$$E(\mathbf{d}_i, B, f_j) = \begin{cases} 1 & \text{if } B \cup f_j \models \mathbf{d}_i \\ 0 & \text{if } B \cup f_j \not\models \mathbf{d}_i \end{cases}$$

In contrast to DBN and SAE, a DRM performs supervised learning in the logical layer. The methodology is presented as Algorithm 2. The algorithm takes as input data that is encoded in FOL form. It utilizes an ILP system to generate a set of first order logical rules and considers all the clauses generated during the search of the hypothesis space as candidate features. The set of induced features can be very very large but finite. In this paper we introduce a well known measure, namely Gain Ratio (GR) to score the rules. The selection of the rules

Algorithm 1 Quantification of the goodness of rules and selection of a subset

Input: A set of input-output pair examples D where $\mathbf{d}_i \in \mathcal{D}$ and $t_i \in \{0,1\}$. The domain background knowledge B, a set $\mathbb{F}$ of R induced rules, and N
Output: A set of selected rules F
for $j = 1$ to R **do**
Compute gain ratio value for each rule f_j that quantifies its goodness
end for
Sort the computed values and select the first N rules with highest GR values.
Return: $F = \{f_1, f_2, \ldots, f_N\}$

Algorithm 2 The logical layer for DRM for binary classification

Input: A set of input-output pair training examples $D = \{(\mathbf{d}_1, t_1), (\mathbf{d}_2, t_2) \ldots, (\mathbf{d}_n, t_n)\}$ where $\mathbf{d}_i \in \mathcal{D}$ and $t_i \in \{0,1\}$
Output: An encoding function E that maps data into the Boolean representation $E : \mathbf{d}_i \rightarrow (\alpha_j)_{j=1}^{j=N}$ where $\alpha_j = 0$ or $\alpha_j = 1$
Induce a set of rules $\mathbb{F}$ by using an ILP algorithm.
Consider all the R rules generated during the search of the hypothesis space.
Select the most informative subset $F = \{f_1, f_2, \ldots, f_N\}$ by using Algorithm 1.
Compute coverage by using entailment: $E(\mathbf{d}_i, B, F) = (\alpha_1, \ldots, \alpha_N)$, where
if $f_j \cup B \models \mathbf{d}_i$ **then**
$\alpha_j = 1$
else if $f_j \cup B \not\models \mathbf{d}_i$ **then**
$\alpha_j = 0$
end if
Return $E(\mathbf{d}, B, F)$

is an important process as it reduces complexity and improves generalization performance of a DRM. To derive an expression for GR we first define entropy and Information Gain (IG). The entropy of a logically encoded (training) data D is the expected value of information required to assign a category to an example: $entropy(D) = H(D) = -\sum_{1 \leq \iota \leq m} \frac{|D_\iota|}{|D|} \log_2 \left(\frac{|D_\iota|}{|D|}\right)$. $|D_\iota|$ is the number of examples that are grouped into category ι. Information gain measures expected reduction in entropy. The IG value between a rule $f_j$and a category ι is computed by considering the division of data D into subsets according to the varying values of the rule, and it is given as: $IG(f_j) = H(D) - \sum_{i \in values(f_j)} \frac{|D_i|}{|D|} H(D_i)$. We obtain a mathematical expression for GR by normalising IG. The normalization value (Z) is given by: $Z(f_j) = -\sum_{i \in values(f_j)} \frac{|D_i|}{|D|} \log_2 \left(\frac{|D_i|}{|D|}\right)$. Now we can define the gain ratio of the rule as follows:$GR(f_j) = \frac{IG(f_j)}{Z(f_j)}$. The method computes GR values for all rules, and identifies the most relevant and informative features by using the procedure described as Algorithm 1. As the higher values of GR tells the goodness of fit for a rule, the N rules with the highest GR are selected.

Algorithm 3 The logical layer for DRM for multi-class classification

Input: A set of input-output pair training examples $D = \{(\mathbf{d}_1, t_1), (\mathbf{d}_2, t_2) \ldots, (\mathbf{d}_n, t_n)$ where $\mathbf{d}_i \in \mathcal{D}$ and $t_i \in \{1, 2, \ldots, m\}$. The background domain knowledge B
Output: An encoding function E that maps data into the Boolean representation $E : \mathbf{d}_i \rightarrow (\alpha_j)_{j=1}^{j=N}$ where $\alpha_j = 0$ or $\alpha_j = 1$
$F = \{\}$
for $\iota = 1$ to m **do**
  Select a class from m classes .
  Formulate the binary class problem by assigning label '1' to examples of class p and '0' to examples of remaining classes.
  Generate a set of rules $\mathbb{F}_\iota$ by inducing an ILP algorithm.
  Consider all the R_ι rules generated during the search of the hypothesis space.
  Select the most informative and relevant subset F_ι by invoking Algorithm 1.
  Add F_ι to F.
end for
The final F is given by: $F = \{F_1, F_2, \ldots, F_m\}$.
Assess whether examples are implied by rules $f_j \in F$: $E(\mathbf{d}_i, B, f_j) = (\alpha_1, \ldots, \alpha_N)$.
Return: $E(\mathbf{d}, B, F)$

The logical layer of a DRM is learned by using the procedure described as Algorithm 2 for examples belonging to two classes. In scenarios where examples belong to $m > 2$ categories, we propose a strategy given as Algorithm 3 to construct the logical layer. The goal is to extract relevant features that describe the data belonging to diverse classes. The multi-class problem is divided into m binary tasks and a positive training set is defined by selecting a class ι where the examples belonging to the remaining classes form the negative training set. A hypothesis that consists of N rules is induced. A subset F_ι is selected by applying Algorithm 1 to a validation or training set. The process is repeated for all the classes, and hence comprises m iterations. The obtained m subsets are merged into a rule set F. If there are any redundant rules they are removed from the set F. An input data vector for the next layer is generated by assessing whether the example satisfy the conditions of the rule set F or not.

The Generative Layers: As the features, learned in the first layer, may not model all the structure, relations and variations in the data, a DRM learns more layers of representation by using RBMs. In these layers an RBM is trained in an unsupervised manner: the weights and the biases are measured, activations of input and hidden unit are computed. The activations of the hidden units of the previous layer become the input data for the current RBM. This process is repeated l times. Predictive problems are solved by using the extracted features.

4 Experiments and Analysis

In this section we describe experiments and results. We empirically tested the proposed method on datasets comprising protein domains, mutagenic and toxic

Table 1: 10-fold cross-validated accuracy ± standard deviation for mutagenesis.

kFOIL	nFOIL	PROGOL	SVILP	DRM(1)	DRM(4)
81.3 ± 11.0	75.4 ±12.3	78.7 ± 10.8	87.2 ± 07.5	88.82 ± 7.63	**89.90 ± 6.33**

Table 2: Five-fold cross validated accuracy ± standard deviation for DSSTOX dataset for MC_ILP, DL_SVILP and DRM (DRM(1) & DRM(4)).

MC_ILP	DL_SVILP	DRM(1)	DRM(4)
57.1 ± 2.4	65.2 ± 02.3	63.57 ± 2.29	**66.06 ± 2.25**

compounds. The experiments were performed by a combination of in house scripts and publicly available software systems. A deep relational architecture was learned that comprised of four layers. The first layer was given by hypothesised rules, and the remaining layers of representation were learned by an RBM. The activation function for these layers was a logistic sigmoid. The features generated at the first and the fourth layer were used in conjunction with an SVM to perform classification tasks. A DRM with one layer and four layers is denoted by DRM(1) and DRM(4) respectively. We compared the performance of DRM with SVILP, PROGOL, multi-class ILP (MC_ILP) [8], and decision list based SVILP (DL_SVILP) [9]. We used accuracy [10] as the evaluation measure. For multi-class classification problems we calculated accuracy for each class and for overall (OA) of the dataset.

Datasets: We conducted experiments on three benchmark datasets. The mutagenesis dataset [11] comprises 188 compounds. Of the 188 molecules, 125 are positive examples and 63 are negative examples. The EPA Fathead Minnow Acute Toxicity database (DSSTox) contains highly diverse organic compounds [12]. In the database there are 442 compounds with unique mode of action that is experimentally determined. The compounds are placed into 8 categories, and the class distribution is skewed in this dataset. Protein dataset [13] has been extracted from the Structural Classification of Proteins (SCOP) database. It contains protein domains belonging to forty five folds (classes). These folds are categorized into four main structural classes, namely α, β, α/β and $\alpha + \beta$. The number of test protein domains are 405. The distribution of protein folds in the dataset is skewed where number of domains in a fold ranges from four to thirty two.

In the case of mutagenesis dataset molecules were represented by using the key information that was given in the form of atom and bond. The compounds in DSSTox dataset were represented by atom bond descriptions, functional groups and rings. The background knowledge described in [13] was used to represent protein domains.

Results: The results presented in Table 1 demonstrated efficacy of DRM for mutagenesis dataset. DRM substantially improved performance of PROGOL and SVILP. We also contrasted DRM with kFOIL and nFOIL by considering the results reported in [6]. The accuracy values confirmed its low error probability as compared to the related shallow methods. Five-fold cross-validated accuracy

Table 3: Average accuracy ± standard deviation for protein fold dataset for MC_ILP, DL_SVILP and DRM (DRM(1) & DRM(4)).

Fold	MC_ILP	DL_SVILP	DRM(1)	DRM(4)
α (8 classes)	57.78 ± 5.21	62.22 ± 5.11	67.02 ± 4.85	**74.47 ± 4.50**
β (14 classes)	33.64 ± 4.57	45.79 ± 4.82	42.31± 4.84	44.23 ± 4.87
α/β (14 classes)	56.45 ± 4.45	62.90 ± 4.33	62.60 ± 4.36	**65.04 ± 4.30**
$\alpha+\beta$ 9 classes)	66.67 ± 5.41	72.62 ± 5.27	77.01 ± 5.19	**88.51 ± 5.11**
All (45 classes)	52.84 ± 2.48	60.25 ± 2.43	61.52 ± 2.41	**66.92 ± 2.33**

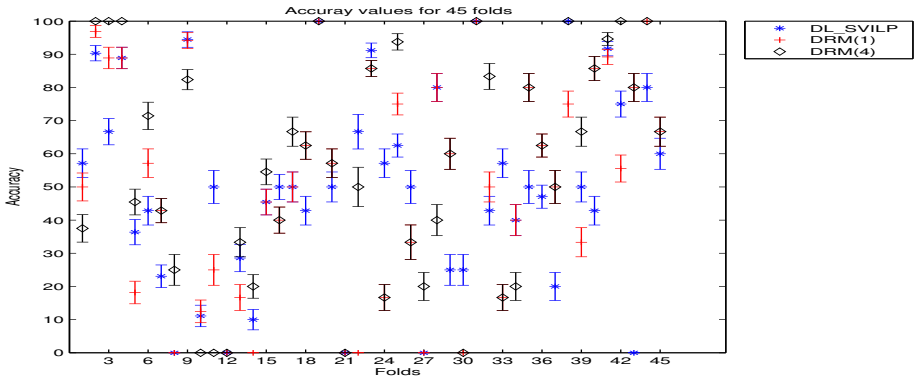

Fig. 1: Accuracy values for DL_SVILP and DRM (DRM(1) & DRM(4)) for 45 protein folds.

values were presented in Table 2 for the DSSTox dataset. We compared performance of proposed technique with MC_ILP and DL_SVILP. The results showed usefulness of deep learning in detecting toxic compounds. In Table 3 and Figure 1, the results of experiments on protein fold dataset were shown. From the results it was evident that DRM substantially and significantly improved upon MC_ILP and DL_SVILP. DRM improved upon DL_SVILP for 23 folds. It was worth noting that the over all accuracy of DRM was substantially and significantly better than DL_SVILP for α, α/β, and $\alpha+\beta$ structural classes. These experiments showed that DRM generally outperformed the related approaches in the study.

5 Conclusion

In this paper we have presented a novel methodology to learn a deep relational architecture by integrating hypothesised logical rules and generative models. The performance of the proposed technique is empirically tested by applying it to complex tasks of protein fold recognition and classification of toxic and mutagenic compounds. Experimental comparisons of the performance of deep relational machine with related approaches show that the proposed approach

generally achieves an accuracy that is substantially higher than all the other methods considered in the study. We aim to further develop DRM by extending the integration stage, exploring non-probabilistic methods for the second and the successive layers and applying it to other challenging real-world problems.

Acknowledgements

The author would like to thank David C Hogg and Tony Cohn for useful comments on draft of the paper. The author would also like to acknowledge the support of the FP7 Collaborative Project COGNITO.

References

1. Hinton, G.E., Salkhutdinov, R.: Reducing the dimensionality of data with Neural Networks. Science **33** (2006) 504–507
2. Bengio, Y., Lamblin, P., Popovici, D., Larochelle, H.: Greedy layer-wise training of deep networks. In: Adv. in Neu. Infor. Processing Systems 19, MIT Press (2007) 153–160
3. McCarthy, J.: Programs with common sense. In: Proc. of Teddington Conf. on the Mechanization of Thought Processes, London: Her Majesty's Stationery Office (1959) 75–91
4. Kramer, S., Lavrac, N., Flach, P.: Propositionalisation approaches to Relational Data Mining. In Dzeroski, S., Larac, N., eds.: Relational Data Mining. Springer, Berlin (2001) 262–291
5. Muggleton, S., Lodhi, H., Amini, A., Sternberg, M.J.E.: Support Vector Inductive Logic Programming. In: Proc. of the 8th Int. Conf. on Discovery Science. Volume 3735 of LNAI., Springer Verlag (2005) 163–175
6. Landwehr, N., Passerini, A., Raedt, L., Frasconi, P.: kFOIL: Learning simple relational kernels. In: Proc. of the Nat. Conf. on Artificial Intelligence (AAAI). Volume 21. (2006) 389–394
7. Ackley, D.H., Hinton, G.E., Sejnowski, T.J.: A learning algorithm for Boltzmann machines. Cognitive. Sci. (9) (1985) 147–169
8. Laer, W.V., de Raedt, L., Dzeroski, S.: On multi-class problems and discretization in Iductive Logic Programming. In: Proc. of the 10th Int. Symposium on Foundations of Intelligent Systems. (1997) 277–286
9. Lodhi, H., Muggleton, S., Sternberg, M.J.E.: Learning large margin first order decision lists for multi-class classification. In: Proc. of the 12th Int. Conf. on Discovery Science, Springer Verlag (2009) 168–183
10. Ding, C.H., Dubchak, I.: Multi-class protein fold recognition using support vector machines and neural networks. Bioinformatics **17** (2001) 349–358
11. Debnath, A.K., de Compadre, R.L.L., Debnath, G., Schusterman, A.J., Hansch, C.: Structure-activity relationship of mutagenic aromatic and heteroaromatics nito compounds. correlation with molecular orbital energies and hydrophobicity. J. Med. Chem. **34**(2) (1991) 786–797
12. Richard, A., Williams, C.: Distributed structure-searchable toxicity (DSSTox) public database network: A proposal. Mutat. Res. **499**(1) (2002) 27–52
13. Cootes, A.P., Muggleton, S., Sternberg, M.J.: The automatic discovery of structural principles describing protein fold space. J. Mol. Biol. **330**(4) (2003) 839–850