A Preliminary Experimental Comparison of Recursive Neural Networks and a Tree Kernel Method for QSAR/QSPR Regression Tasks

Alessio Micheli
Dept. of Computer Science
University of Pisa

Filippo Portera, Alessandro Sperduti
Dept. of Pure & Appl. Mathematics
University of Padova

Abstract. We consider two different methods for QSAR/QSPR regression tasks: Recursive Neural Networks (RecNN) and a Support Vector Regression (SVR) machine using a Tree Kernel. Experimental results on two specific regression tasks involving alkanes and benzodiazepines are obtained for the two approaches.

1 Introduction

In recent years several researchers have started to consider the adaptive processing of structured data. This interest is motivated by two main reasons: *i)* several very important computational problems in bioinformatics, chemistry, document classification and filtering (just to name a few), require the use of some machine learning procedure to be properly treated because their complexity does not allow a formal and precise definition of the problem and thus no algorithmic solution to the problem is known; *ii)* in many of the above problems, the objects of interest are more naturally represented via structured representations of different sizes, such as sequences, strings, trees, directed or undirected graphs, which retain all the structural information relevant for solving the task.Within this area there are two main streams of research relevant for the neural network community: *a)* Recurrent and Recursive Neural Networks (see, for example, [4]); *b)* Kernel Methods for Structured Data (see, for example,[5]).

The work presented in this paper is a small empirical step towards a study on the merits and drawbacks of these two approaches. Here we consider one QSAR and one QSPR regression task and we report the experimental results we have obtained by the two approaches.

2 Recursive NN and a Tree Kernel

Recursive neural networks (RecNN) [8] are neural network models able to realize mappings from a set of directed positional acyclic graphs (DPAGs) (with labeled nodes) $\mathcal{I}^{\#}$ to the set of real vectors. Specifically, the class of functions which can be computed by these models can be characterized as the class of functional graph transductions $\mathcal{T} : \mathcal{I}^{\#} \rightarrow \mathbb{R}^k$, which can be represented in the following form $\mathcal{T} = g \circ \hat{\tau}$, where $\hat{\tau} : \mathcal{I}^{\#} \rightarrow \mathbb{R}^m$ is the *encoding* function and $g : \mathbb{R}^m \rightarrow \mathbb{R}^k$ is the *output* function. Specifically, given a DPAG $\boldsymbol{Y}$, $\hat{\tau}$ is defined recursively as

$$\hat{\tau}(\boldsymbol{Y}) = \begin{cases} \mathbf{0} \text{ (the null vector in } \mathbb{R}^m) & \text{if } \boldsymbol{Y} = \xi \\ \tau(s, \boldsymbol{Y}_s, \hat{\tau}(\boldsymbol{Y}^{(1)}), \ldots, \hat{\tau}(\boldsymbol{Y}^{(o)})) & \text{otherwise} \end{cases} \quad (1)$$

where a (*stationary*) τ can be defined as $\tau : \mathbb{R}^n \times \underbrace{\mathbb{R}^m \times \cdots \times \mathbb{R}^m}_{\text{o times}} \rightarrow \mathbb{R}^m$, $\mathbb{R}^n$ is the label space, the remaining domains represent the encoded subgraphs spaces up to the maximum out-degree of the input domain $\mathcal{I}^{\#}$, o is the maximum out-degree of DPAGs in $\mathcal{I}^{\#}$, $s = source(\boldsymbol{Y})$, $\boldsymbol{Y}_s$ is the label attached to the source of $\boldsymbol{Y}$, and $\boldsymbol{Y}^{(1)}, \ldots, \boldsymbol{Y}^{(o)}$ are the subgraphs pointed by s. The specific neural architecture, based on the above recursive function, we have used for the experiments reported in this paper is Recursive Cascade Correlation (RecCC), fully described in [8].

Concerning the kernel operating on trees, we have chosen the most popular and used Tree kernel proposed in [2]. It is based on counting matching subtrees between two input trees. Given an input tree x, let s_x be a subtree of x if s_x is rooted in a node of x and the set of arcs of s_x is a subset of connected arcs of x. We assume that each of the m subtrees in the whole training data set is indexed by an integer between 1 and m. Then $h_s(x)$ is the number of times the tree indexed with s occurs in x as a subtree. We represent each tree x as a feature vector $\phi(x) = [h_1(x), h_2(x), \ldots]$. The inner product between two trees under the representation $\phi(x) = [h_1(x), h_2(x), \ldots h_m(x)]$ is: $K(x, y) = \phi(x) \cdot \phi(y) = \sum_{s=1}^{m} h_s(x) h_s(y)$.

Experimental results showed that this kernel may weight larger substructures too highly, producing a Gram matrix with large diagonals. In [2], they describe a method to dim the effect of the exponential blow-up in the number of subtrees with their depth. We can downweight larger subtrees modifying the kernel as follows: $K(x, y) = \sum_{s=1}^{m} \lambda^{\text{size}(s)} h_s(x) h_s(y)$ where $0 < \lambda \leq 1$ is a weighting parameter and size(s) is the number of nodes of the subtree sx. The Tree kernel can be calculated with a recursive procedure in $O(|NX| \cdot |NY|)$ time where NX and NY are the sets of nodes of trees x and y, respectively.

3 QSPR/QSAR Tasks

Here we consider two paradigmatic instances of the regression problem defined on a structured domain, one for QSPR analysis, and one for QSAR analysis. Both problems have been previously faced by RecNN and favorably compared with respect to state-of-the-art standard approaches used in the QSPR/QSAR field [7, 1].

The QSPR problem consists in the prediction of the boiling point for a group of acyclic hydrocarbons (alkanes). The data set used is described in [6] and comprised all the 150 alkanes with up to 10 carbon atoms, allowing to consider the problem of coping with structures of different sizes. The target values are in the range approximatively, in Celsius degrees, [-164 , 174]. The QSAR

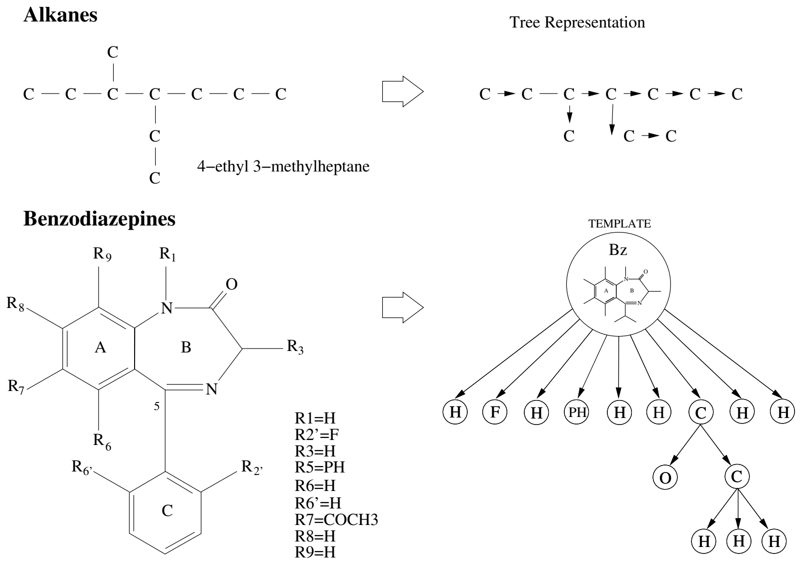

Figure 1: Example of representation for an alkane and a benzodiazepine.

problem considered here involves a class of chemical compounds belonging to a class of therapeutical interest: benzodiazepines. Several QSAR studies have been carried out aiming at the prediction of the non-specific activity (affinity) towards the Benzodiazepine/GABA$_A$ receptor. A group of benzodiazepines (Bz) (classical 1,4-benzodiazepin-2-ones) has been used for our experiments [7]. The total number of molecules is 72, of which 5 are used as test set. The target values range in [6, 9]. The analyzed molecules present a common structural aspect given by the benzodiazepine ring and they differ each other because of a large variety of substituents at the positions showed in Fig. 1.

Molecular Structure Representation

An appropriate description of the molecular structures analyzed in this work is based on a labeled tree representation. Thus, both RecNN and Tree kernel can be applied, allowing us to preliminary compare them on a fair basis.

In order to obtain an unique structured representation of each compound, and their substituent fragment, as labeled positional trees (k-ary trees, which are a subclass of DPAGs), we have defined a set of representation rules.

It is worth to note that alkanes (acyclic hydrocarbons molecules) are trees. In order to represent them as labeled k-ary trees, carbon-hydrogens groups are associated with vertexes, and bonds between carbon atoms are represented by edges; the root of the tree can be determined by the first carbon-hydrogens group according to the IUPAC nomenclature system and the total order over the edges can be based on the size of the sub-compounds.

In the case of benzodiazepines, the major atom group that occurs unchanged throughout the class of analyzed compounds (common template) constitutes the root of the tree. Note that, an alternative representation, would have been to explicitly represent each atom in the major atom group (by a graph based representation). However, since this group occurs in all the compounds, no additional information is conveyed by adopting this representation. Finally, each substituent fragment is naturally represented as a tree once cycles are treated as replicated atom groups and described by the label information.

As a result the use of labeled trees (namely labeled k-ary trees) does not imply the loss of relevant information for these classes of compounds, which are representative of a large class of QSPR and QSAR problems. In particular, the representation of compounds is strictly related to the molecular topology and also conveys detailed information about the presence and types of the bonds, the atoms, and the chemical groups and chemical functionalities. Examples of representations for alkanes and benzodiazepines are shown in Fig. 1.

Summarizing, the representation rules (that are fully discussed in [7] and [1] for these sets of compounds) allows us to give an unique labeled k-ary tree representation of various sets of compounds through a conventional representation of cycles, by giving direction to edges, and by defining a total order over the edges. Since the rules are defined according to the IUPAC nomenclature, they retain the standard representational conventions used in Chemistry.

4 Experimental Comparison

The target values of the datasets are obtained by experimental procedures, so it is useful to fit them according to a maximal tolerance (ϵ_t) on the error compatible with the experimental error, i.e. $\epsilon_t = 8$ for the alkanes dataset and $\epsilon_t = 0.4$ for the benzodiazepines dataset.

For RecNN we decided to stop training whenever the maximum absolute training error was below ϵ_t. The software we used for the SVR algorithm (SVMLight 5.0) follows a stop criterion based on the violation of the Kuhn-Tucker conditions of the computed dual solution. In fact, the criterion used by the solver disregards patterns with large error and with a related dual variable equal to C. So, the solution given in output can exhibit a maximum absolute training error that is above the experimental error. For the sake of comparison, we considered also a stop criterion where training is stopped when every support vector has an absolute error below ϵ_t. We decided to compose the tree kernel $K(x,y)$ with an RBF kernel obtaining the kernel $K_{RBF}(x,y) = e^{-\sigma(K(x,x)-2K(x,y)+K(y,y))}$.

For the alkanes dataset we performed a 10-fold cross validation. The benzodiazepines dataset consists in a split of 67 training patterns and 5 test patterns.

Due to the large amount of parameters allowed by the RecCC models, an

alkanes								
Method	m_{AE} *tr*	V_{AE} *tr*	m_{AE} *ts*	V_{AE} *ts*	C	σ	w	λ
RecCC	2.15	0.013	2.86	0.492	$\epsilon_t \leq 8$			
TK_{RBF}	3.23	0.137	3.59	0.840	1e3	1	0.1	0.45
$TK_{RBF,c}$	2.68	0.015	3.09	0.463	1e3	0.1	2	0.55
benzodiazepines								
Method	M_{AE} *tr*	m_{AE} *tr*	M_{AE} *ts*	m_{AE} *ts*	C	σ	w	λ
RecCC	0.360	0.087	0.606	0.255	$\epsilon_t \leq 0.4$			
TK_{RBF}	1.355	0.616	1.175	0.779	1e3	0.01	1	0.55
$TK_{RBF,c}$	0.366	0.175	0.883	0.242	1e3	1e-3	0	0.63

Table 1: Results for the alkanes and benzodiazepines datasets.

initial set of preliminary trials were performed just to determine an admissible range for the learning parameters. However, no effort was done to optimize these parameters with respect to the two specific tasks: the main aim of the experiments was to show how RecCC could deal with two completely different tasks using the same basic models. Due to the different result achieved by different random initialization for the connection weights, various trials were carried out for the RecCC simulations and the mean values have been reported over five trials (alkanes) and six trials (Bz), respectively.

For the calibration of SVR hyperparameters for alkanes, we shuffled the 150 patterns and we created 30 splits of 5 patterns each. The calibration involved a set of 4 parameters: the SVR training error weight constant C, the RBF kernel width σ, the SVR regression tube width w and the tree kernel downweighting factor λ. On the last 3 splits we applied a 3-fold cross validation based on a mesh of $5 \times 5 \times 5 \times 9$ vectors spanning the parameters space. We selected the parameter vector that gave the median of the best mean absolute validation error on the three splits and then we used these parameters for the 10-fold cross validation. For benzodiaziepines calibration we applied a 3-fold cross validation based on the same parameters mesh. We selected the parameter vector that gave the best mean absolute validation error and we evaluated the algorithm on the original test set.

At the top of Table 1 we report the results obtained on the alkanes dataset with the RecCC algorithm (RecCC) [6], with an SVR with a standard termination criteria (RBF TK) and with an SVR that terminates only when the maximum absolute error on input patterns is below the given tolerance ϵ_t (RBF TK_c). We report the mean absolute training error (m_{AE} tr), the variance of the absolute training error distribution (V_{AE} tr), the mean absolute test error (m_{AE} ts), the variance of the absolute test error distribution (V_{AE} ts) and the obtained calibration parameters. The bottom of Table 1 shows the results obtained with the three algorithms on the benzodiazepines dataset. For the each run on the training and test set, we report the maximum absolute error on the training set (M_{AE} tr), the mean absolute error on the training set (m_{AE} tr), the maximum absolute error on the test set (M_{AE} ts), the mean absolute error on the test set (m_{AE} ts) and the obtained calibration parameters.

5 Conclusion

The preliminary experimental results we have obtained show that using the same stopping criterion there is no much difference between RecCC and SVR with the tree kernel proposed in [2]. Due to the relatively small size of trees in the used datasets, SVR training is usually faster then RecCC training. However, when considering larger datasets, the computational complexity of the kernel is going to slow down training of SVR (see for example [3]), since the kernel matrix computation is $O(L^2 \times N^2)$ where L is the number of training examples and $N = \max_{s\in[1,m]} size(s)$, while in RecCC each batch training iteration costs $O(H^2 \times L \times N)$, where H is the number of hidden units. Thus it is clear that the matrix kernel computation depends quadratically on both L and N while for RecCC there is only a linear dependence.

References

[1] A.M. Bianucci, A. Micheli, A. Sperduti, and A. Starita. Application of cascade correlation networks for structures to chemistry. *Journal of Applied Intelligence (Kluwer Academic Publishers)*, 12:117–146, 2000.

[2] M. Collins and N. Duffy. Convolution kernels for natural language. In *NIPS 14*, Cambridge, MA, 2002. MIT Press.

[3] F. Costa, P. Frasconi, and S. Menchetti. http://www.dsi.unifi.it/˜paolo/talks/nips-02-workshop.pdf

[4] P. Frasconi, M. Gori, A. Kuechler, and A. Sperduti. From sequences to data structures: Theory and applications. In *A Field Guide to Dynamic Recurrent Networks*, pages 351–374. Wiley-IEEE Press, 2001.

[5] T. Gaertner. A survey of kernels for structured data. *Newsletter of the ACM Special Interest Group on Knowledge Discovery and Data Mining*, 5(1):49–58, July 2003.

[6] A. Micheli, D. Sona, and A. Sperduti. Contextual processing of structured data by recursive cascade correlation. Submitted to IEEE Trans. on Neural Networks, 2003.

[7] A. Micheli, A. Sperduti, A. Starita, and A.M. Bianucci. Analysis of the internal representations developed by neural networks for structures applied to quantitative structure-activity relationship studies of benzodiazepines. *Journal of Chem. Inf. and Comp. Sci.*, 41(1):202–218, January 2001.

[8] A. Sperduti and A. Starita. Supervised neural networks for the classification of structures. *IEEE Transactions on Neural Networks*, 8(3):714–735, 1997.