ESANN 2016 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 27-29 April 2016, i6doc.com publ., ISBN 978-287587027-8.
Available from http://www.i6doc.com/en/.

A Reservoir Activation Kernel for Trees

Davide Bacciu and Claudio Gallicchio and Alessio Micheli *

Dipartimento di Informatica - Università di Pisa - Italy

Abstract. We introduce an efficient tree kernel for reservoir computing models exploiting the recursive encoding of the structure in the state activations of the untrained recurrent layer. We discuss how the contractive property of the reservoir induces a topographic organization of the state space that can be used to compute structural matches in terms of pairwise distances between points in the state space. The experimental analysis shows that the proposed kernel is capable of achieving competitive classification results by relying on very small reservoirs comprising as little as 10 sparsely connected recurrent neurons.

1 Introduction

The classification of tree data is a general and popular problem since much information can naturally be represented in a hierarchical structured form (e.g. images, molecules, text documents, biomedical data). Here, we turn our attention on reservoir computing approaches for trees, which implement a recursive encoding of input structures by exploiting the recurrent layer of sparsely connected reservoir neurons. By this means, they implicitly realize a mapping of a structure into a state space defined by the activation of the reservoir neurons following the tree encoding process. In other words, the reservoir activations can be interpreted as features summarizing structural properties of an input tree. More importantly, the contractive property of the reservoir [1] ensures that the state space of the reservoir activations is characterized by a topographical organization, such that similar structures are likely to produce the same reservoir neuron activations or, else said, that they are mapped to neighboring points in the reservoir state space. In this work, we exploit such intuition to define an efficient kernel that computes similarity between trees in terms of distances between the corresponding projections in the reservoir state space of a Tree Echo State Network (TreeESN) [1]. The proposed approach falls into a family of *activation kernels* exploiting the internal representation developed by a recursive model for trees. In [2], a kernel is defined on the discrete activations of a recurrent self-organizing map for trees, while in [3] the kernel exploits the topographic projections defined by the activations of the hidden Markov states of a generative mapping for trees [4]. Both approaches require a costly training phase of the underlying unsupervised learning model and require a large number of recurrent units/hidden states in order to develop a feature space representation that allows capturing sufficient structural information. In this work, we put forward the idea that the dense recursive encoding defined by the untrained reservoir

*This work is partially supported by the MIUR-SIR project LIST-IT (grant n. RBSI14STDE)

ESANN 2016 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 27-29 April 2016, i6doc.com publ., ISBN 978-287587027-8.
Available from http://www.i6doc.com/en/.

of a TreeESN provides a very rich representation of the structural knowledge that allows defining an efficient kernel over very small reservoirs. The experimental analysis shows how this leads to compact reservoirs with competitive classification accuracies over heterogeous application, despite the kernel being non adaptive given the untrained nature of the reservoir.

2 A topographic kernel on reservoir encodings of tree data

Let us consider a dataset $\mathcal{D} = \{\mathbf{t}^1, \ldots, \mathbf{t}^L\}$ of L structures $\mathbf{t}^l$ that are labeled and rooted trees with maximum finite out-degree D (i.e. the maximum number of children of a node). Each vertex u in the input tree is associated with a N_{in}-dimensional label t_u. The set of children of node u is denoted by ch_u.

Input trees are mapped into tree-structured feature (state) representations through the computation carried out by a TreeESN. A TreeESN is neural network model for tree structured data, implementing a recursive encoding of input trees by means of a hidden *reservoir* layer of non-linear and sparsely connected recurrent units. Given an input tree $\mathbf{t}$, the TreeESN encoding process is performed by resorting to a bottom up visit of $\mathbf{t}$, starting from the leaf nodes and ending in the root. For each node u, the N-dimensional reservoir computes a feature representation (or state) denoted by x_u, according to

$$x_u = f(W_{in} t_u + \sum_{v \in ch_u} \hat{W} x_v), \tag{1}$$

where $W_{in} \in \mathbb{R}^{N \times N_{in}}$ is the input-to-reservoir weight matrix (possibly including a bias term), $\hat{W} \in \mathbb{R}^{N \times N}$ is the recurrent reservoir weight matrix and f is a unit-wise applied activation function (we use tanh). The reservoir parameters are initialized to implement a contraction mapping, which ensures stability of state dynamics. In this regard, a key hyper-parameter of TreeESN is the *contraction coefficient* $\sigma = D\|\hat{W}\|_2$, which relates to the stability of TreeESN state dynamics. Although a sufficient condition for contractive initialization of reservoirs in the L-2 norm prescribes that $\sigma < 1$, this is often too restrictive in practical applications, thereby values of σ slightly larger than 1 are also considered [1]. In practice, values in $\hat{W}$ are randomly initialized and then scaled to the desired value of σ, whereas values in W_{in} are chosen from a uniform distribution over $[-scale_{in}, scale_{in}]$. One of the distinctive characterizations of the TreeESN encoding process is its efficiency, as indeed the reservoir parameters are left *untrained* after initialization. Although out of the focus of this paper, when TreeESNs are applied to supervised learning tasks, a linear readout layer is used for output computation, typically trained to solve a least mean squared problem by using direct methods. Details on TreeESNs can be found in [1].

The state space of the reservoir activations is characterized by a topographical organization induced by the recursive encoding and the contractive properties of the reservoir, which leads to a Markovian characterization of reservoir dynamics extended to the case of tree domains [1]. In this sense, the reservoir state space organization closely recalls that of the hidden Markov states in the generative

ESANN 2016 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 27-29 April 2016, i6doc.com publ., ISBN 978-287587027-8.
Available from http://www.i6doc.com/en/.

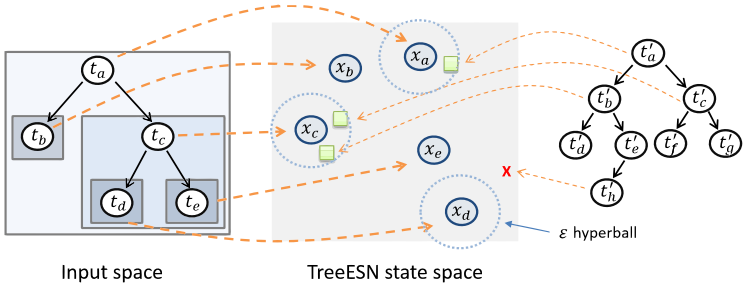

Fig. 1: Bottom up encoding process of an input tree $\mathbf{t}$ by the TreeESN reservoir and computation of the matches with another tree $\mathbf{t}'$ by means of the ϵ-neighborhood of the node projections: matching and non-matching $\mathbf{t}'$ nodes are denoted as squares and Xs, respectively.

topographic mapping model for trees [4]. Such topographic principle ensures that points that are neighbors in the reservoir state space can be considered to be encoding similar structural knowledge. By exploiting such information (along the lines of what [3] does for the generative topographic model) one can define an efficient tree kernel expressing structural similarity between two trees in terms of the distance of their encodings in the reservoir state space.

The recursive TreeESN encoding in (1) projects each substructure $\mathbf{t}_u$ composing a tree to a N-dimensional point x_u corresponding to the reservoir activation for the tree node acting as root of the substructure. Figure 1 shows how a tree $\mathbf{t}$ comprising T nodes is transformed into T vectors $x_u = \Phi(\mathbf{t}_u) \in \mathbb{R}^N_{[-1,1]}$, one for each node u of the tree. Evaluating the similarity between two structures, in this context, becomes a matter of computing distances between points in the reservoir state space. To this end, we define the following *weight function* between two generic N-dimensional points x and x' on the reservoir activation space, i.e.

$$W_\epsilon(x, x') = \begin{cases} \epsilon - d(x, x'), & \text{if } d(x, x') \leq \epsilon \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

where $d(x, x')$ is the standard Euclidean distance. The term ϵ determines a neighborhood for the reservoir states which regulates the influence of distant substructures in defining the kernel-induced similarity measure (see the ϵ-hyperballs in Fig. 1). In other words, it is the parameter regulating the soft-matching among the states, determining which reservoir activations configurations can be considered sufficiently similar. The resulting *kernel on TreeESN* (KTESN, in short) between trees $\mathbf{t}^1$ and $\mathbf{t}^2$ is

$$k_{KTESN}(\mathbf{t}^1, \mathbf{t}^2) = \sum_{u \in \mathcal{U}_1} \sum_{u' \in \mathcal{U}_2} W_\epsilon(x_u, x_{u'}) \tag{3}$$

where x_u and $x_{u'}$ are the recursive reservoir encodings of subtrees $\mathbf{t}^1_u$ and $\mathbf{t}^2_{u'}$ from tree $\mathbf{t}^1$ and $\mathbf{t}^2$, respectively. Note that (3) can be shown to be a particular case of Wendland function [5] and as such it is positive definite.

ESANN 2016 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 27-29 April 2016, i6doc.com publ., ISBN 978-287587027-8. Available from http://www.i6doc.com/en/.

3 Experimental Results

Experiments have been performed to benchmark KTESN on classification tasks against the underlying TreeESN and relevant tree kernels in literature. The datasets span different application areas to show generality of the approach with respect to varying characteristics of the tree population. The first dataset is from the INEX 2006 competition [6] and comprises 12,107 trees representing XML documents from 18 classes, with 65 possible node labels representing XML tags. Standard training and test sets are available for this dataset [6], with a 50%-50% split. We also consider two datasets from the KEGG/Glycan database [7], concerning a binary classification of the molecular structure of glycans. The first is the Leukemia dataset, comprising 442 glycan structures related to leukemic cells, with 57 different node labels. The second is the Cystic dataset, comprising 160 molecules and 29 node labels, where positive class denotes those associated to cystic fibrosis. Both datasets come with a predefined 10-fold partitioning [1]. These benchmark differs considerably from INEX: the task is binary and a small number of samples is available; trees are small, such that maximum number of nodes is 23 for Leukemia and 15 for Cystic, and have a small outdegree $L = 3$.

Different reservoir configurations and TreeESN model hyper-parametrization have been assessed, by considering reservoirs of 10 and 25 neurons, $scale_{in} \in \{0.01, 0.1, 1\}$ and $\sigma \in \{0.5, 1, 2, 3\}$. In addition, the effect of the choice of the KTESN neighborhood metaparameter ϵ has been evaluated by allowing ϵ to vary in $\{0.05, 0.1, 0.2\}$, which ensures a coverage of maximum 6% of the entire TreeESN state space, along the lines of [2]. The tree classifiers have been realized through support vector classification, using the publicly available LIBSVM [8] software, by means of a C-SVM classifier that receives in input the KTESN Gram matrices. Different values of the SVM cost parameter C_{svm} have been explored, i.e. 0.001, 0.01, 0.1, 1, 10, 100, 1000. To account for randomization effects, we have generated 5 different random reservoir topologies and weights for each reservoir size: results reported in the following refer to average accuracies on such 5 reservoirs. In INEX 2006, a 3-fold cross-validation (CV) procedure has been applied to the training data split to select all the meta-parameters from the ranges listed above. As for the Glykans datasets, we provide the best 10-fold predictive performance averaged on the 5 reservoir guesses expressed in terms of the area under the curve (AUC).

Table 1 reports the performance of the KTESN kernel on INEX 2006 and confronts it with the results obtained by TreeESN, whose recursive encoding process is exploited by KTESN to compute the feature space mapping of the trees. Additionally, we confront with the AM-SOM kernel [2], which has introduced a kernel on the activations of a SOM-SD recursive neural network for structures. Further, we consider an adaptive generative kernel, i.e. AM-GTM [3], exploiting the hidden states of a generative topographic map for trees (GTM-SD) [4]. Since the proposed KTESN is essentially a non-adaptive convolutional kernel, we also report the performance of the most relevant syntactic tree kernels in literature, i.e. ST, SST and PT. The results in Table 1 show that the KTESN

ESANN 2016 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 27-29 April 2016, i6doc.com publ., ISBN 978-287587027-8. Available from http://www.i6doc.com/en/.

KTESN ($N = 10$)	KTESN ($N = 25$)	TreeESN ($N = 500$)	ST	SST	PT	AM SOM	AM GTM
43.41 (0.06)	43.42 (1.15)	42.62	32.02	40.41	41.13	40.07	43.71

Table 1: Results on the INEX 2006 dataset: average classification accuracy on the test set; standard deviation is between brackets (when available). The number of KTESN reservoir neurons selected in validation is highlighted in bold.

yields to high classification accuracies while using a small number of reservoir neurons: already 10 units are sufficient to achieve an accuracy that is superior to that of a TreeESN model with a reservoir of 500 neurons. In this sense, the compact reservoir of KTESN compares favourably also with the activation mask kernels computed on the SOM-SD neural network and on the GTM-SD generative model which require maps of 110×80 and 20×20 units, respectively, to achieve the results reported in Table 1. Using barely 10 units, KTESN can outperform AM-SOM and achieves a comparable accuracy to an AM-GTM model which has 40-times more encoding units and whose inferential-encoding process is quadratic in the number of units, while that by KTESN is $O(N)$ (see (1)). Differently from AM-SOM and AM-GTM, KTESN is not an adaptive kernel hence it does not incur in the cost of training the underlying adaptive model. When compared to other non-adaptive tree kernels in Table 1, KTESN shows a consistently superior accuracy, even with respect to expressive but computationally demanding kernels such as PT, whose complexity is $O(T^3)$ (where T is the number of nodes in the larger tree). KTESN is $O(T^2)$ in the worst case, which is very unlikely as it requires all the nodes from the two trees being projected in an hyperball of radius $\leq \epsilon$; more commonly KTESN complexity would be $O(c_\epsilon T)$, where c_ϵ is a constant depending on the value of the neighborhood parameter ϵ.

Table 2 shows the results of the experimental assessment on Glycans classification. Despite being a non-adaptive kernel, KTESN shows a competitive performance even when the characteristics of the tree population change drastically with respect to previous application (smaller outdegree and size, deeper structures, reduced sample size). In particular on the Cystic dataset KTESN is capable of considerably increasing the accuracy with respect to a TreeEsn model, again using as little as 10 reservoir units. KTESN accuracy is again competitive, if not superior, to that of syntactic kernels of superlinear complexity such as ST, SST and the Linkage (LK) kernel.

Dataset	KTESN ($N = 10$)	KTESN ($N = 25$)	TreeESN	ST	SST	LK
Leukemia	0.9762	0.9769	0.9710	0.9607	0.9710	0.9627
Cystic	0.8356	0.8385	0.7719	0.7983	0.8500	0.7754

Table 2: Results on the Glycans datasets: average AUC on the 10-folds; KTESN results are averaged on the 5 random reservoirs.

ESANN 2016 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 27-29 April 2016, i6doc.com publ., ISBN 978-287587027-8.
Available from http://www.i6doc.com/en/.

4 Conclusion

We have introduced a tree kernel exploiting the topographical organization of the reservoir activation space induced by the recursive encoding and contractive properties of a TreeESN. The proposed kernel maps each tree substructure to an N-dimensional vector of reservoir activations, allowing to measure the similarity between two trees in terms of distances between the reservoir activation vectors of the nodes. A preliminary experimental assessment shows that the proposed approach compares favourably with adaptive and syntactic tree kernels in literature on heterogeneous tree classification tasks. In particular, it can effectively exploit the structural information captured by the dense encoding of untrained reservoirs comprising only 10 sparsely connected neurons. Conversely, activation-based kernels in literature, both for recurrent neural networks and generative tree Markov models, require a number of units that is over 40 times larger and have considerably higher computational requirements both for model training and for computing tree encoding. Future work will deepen the experimental analysis by exploring additional contractive random tree encodings and associated kernels, such as using the root node encoding as an explicit feature representation, and will characterize the computational complexity of the approach. On a more theoretical side, we intend to explore the relationship between reservoir activation kernels and the family of Binet-Cauchy kernels on dynamical systems [9].

References

[1] C. Gallicchio and A. Micheli. Tree echo state networks. *Neurocomputing*, 101:319–337, 2013.

[2] F. Aiolli, G. Da San Martino, M. Hagenbuchner, and A. Sperduti. Learning nonsparse kernels by self-organizing maps for structured data. *IEEE Transactions on Neural Networks*, 20(12):1938 –1949, 2009.

[3] D. Bacciu, A. Micheli, and A. Sperduti. Adaptive tree kernel by multinomial generative topographic mapping. In *Proc. of the 2011 IEEE International Joint Conference on Neural Networks (IJCNN'11)*, pages 1651–1658. IEEE, 2011.

[4] D. Bacciu, A. Micheli, and A. Sperduti. Compositional generative mapping for tree-structured data - part II: Topographic projection model. *IEEE Transactions on Neural Networks and Learning Systems*, 24(2):231–247, 2013.

[5] M.S. Floater and A. Iske. Multistep scattered data interpolation using compactly supported radial basis functions. *J. Comput. Appl. Math.*, 73(1-2):65–78, October 1996.

[6] L. Denoyer and P. Gallinari. Report on the XML mining track at INEX 2005 and INEX 2006: categorization and clustering of XML documents. *SIGIR Forum*, 41(1):79–90, 2007.

[7] M. Kanehisa, S. Goto, S. Kawashima, Y. Okuno, and M. Hattori. The KEGG resource for deciphering the genome. *Nucleic Acids Research*, 32(suppl 1):277–280, 2004.

[8] C. Chih-Chung and J. Chih-Jen. *LIBSVM: a library for support vector machines*, 2001. `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[9] S.V.N. Vishwanathan, A.J. Smola, and R. Vidal. Binet-Cauchy kernels on dynamical systems and its application to the analysis of dynamic scenes. *International Journal of Computer Vision*, 73(1):95–119, 2007.