ESANN'2004 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 28-30 April 2004, d-side publi., ISBN 2-930307-04-8, pp. 281-292

Neural methods for non-standard data

Barbara Hammer[1] and Brijnesh J. Jain[2]

[1] AG LNM, Dept. of Math./Comp.Science, University of Osnabrück, Germany
hammer@informatik.uni-osnabrueck.de,
[2] Methods of Artificial Intelligence, Faculty IV, TU Berlin, Germany
bjj@cs.tu-berlin.de

Abstract. Standard pattern recognition provides effective and noise-tolerant tools for machine learning tasks; however, most approaches only deal with real vectors of a finite and fixed dimensionality. In this tutorial paper, we give an overview about extensions of pattern recognition towards non-standard data which are not contained in a finite dimensional space, such as strings, sequences, trees, graphs, or functions. Two major directions can be distinguished in the neural networks literature: models can be based on a similarity measure adapted to non-standard data, including kernel methods for structures as a very prominent approach, but also alternative metric based algorithms and functional networks; alternatively, non-standard data can be processed recursively within supervised and unsupervised recurrent and recursive networks and fully recurrent systems.

1 Introduction

Pattern recognition tools and statistical classifiers such as feed-forward neural networks (FNNs), support vector machines (SVMs), or self-organizing maps (SOMs) can deal with real-life noisy data in an efficient way and they constitute successful models in various areas of applications. However, most neural methods are restricted to real vectors of a finite and fixed dimensionality as input. As a consequence, extensive preprocessing of data is usually necessary for typical applications of neural methods to real-life scenarios. Thereby, inputs are represented by a finite-dimensional vector of problem-dependent real-valued features: categorical variables are encoded by one-hot-encoding, time series are embedded into a finite dimensional vector space using time windows, preprocessing of images includes edge detection and various filters, sound signals and utterances can be represented by cepstrum vectors, chemical data is characterized by topological indices and physicochemical attributes, to mention a few examples. Alternative data formats and data representations exist:

sets without a specified order can describe objects in a scene or a set of measurements such as contact points of a gripper,

functions evaluated at specific points constitute a natural description for time series or spectral data,

sequences of arbitrary length represent time series or spatial data, such as natural language, text documents, or DNA sequences,

tree structures describe terms, logical formulas, parse trees, or phylogenetic trees,

ESANN'2004 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 28-30 April 2004, d-side publi., ISBN 2-930307-04-8, pp. 281-292

graph structures can be used to encode chemical formulas, scenes in images, or objects built of various primitives.

Feature encoding of these type of data yields compact vectors, however, the encoding is often problem specific and time-consuming. Moreover, information is usually lost if complex data structures such as sequences, trees, or graphs of possibly arbitrary size are encoded in fixed dimensional vectors. An extension of neural methods to directly deal with more complex data is thus desirable.

In this article, we give an overview about extensions of neural methods to non-vectorial data. We distinguish two main directions: similarity based approaches and recursive models. In similarity based approaches, the similarity measure constitutes the interface to process more complex data. Structures are processed as a whole based on the output of the similarity measure adapted for non-standard data. Recursive models, on the other hand, decompose the structures and recursively process the basic constituents within the context given by the already processed related constituents.

2 Similarity based approaches

A variety of neural methods processes data based on a similarity measure or metric: given an input vector x, feed-forward neural networks compute the dot product $w \cdot x$ of the input and the weights w of the neurons; the support vector machine substitutes the standard Euclidean dot product by a kernel $k(x^i, x)$, x^i being a support vector, which can be interpreted as a dot product in an implicit feature space; radial basis function networks and self-organizing maps are based on the Euclidean metric $|x-w|$, w denoting the weight vector of a neuron. If the input x is not element in a finite dimensional real-vector space, this dot product, kernel, or similarity measure can be substituted by a generalization which measures the similarity of more complex objects. Depending on the respective model, further issues such as how to adapt the weight vector are to be specified. To give an example, the approach [47] suggests a structural perceptron for adaptive processing of graphs within a supervised and unsupervised setting. To facilitate adaptive processing, each neuron is associated with an attributed weight graph and the concept of an inner product of vectors is replaced by the Schur-Hadamard inner product of graphs. Despite its name, the Schur-Hadamard inner product is not an inner product, but shares some useful properties of an inner product to extend supervised and unsupervised neural learning machines for attributed graphs. Training of networks composed of structural units is based on minimizing a suitable error function as a function of adjustable weights. In the following, we tackle three different approaches within the context of similarity based methods.

2.1 Functional networks

Functional networks fall within the framework of functional data analysis processing functions f as data points [75]. Thereby, exact input functions are usually not available in practice. Rather, a vector of input-output pairs $(x^i, f(x^i))_{i=1}^{n_f}$ is given. Examples for these type of data are time series, which can be seen as the evaluation of a function at different time steps, whereby the length of the time series or the sampling frequency, i.e. the positions of function evaluation might vary. Spectrometric data provide an alternative application area since it can be interpreted as the signal of the same frequency function evaluated for different ranges. Given this vector, the

input to functional data models is a function fitted to the values. Thereby, standard approximation techniques such as B-spline approximation can be used.

The space of square integrable functions is infinite dimensional; however, it possesses a dot product $f \cdot g = \int_X f(x)g(x)dx$ and techniques from standard vector algebra can be transferred to this case. Thus, linear techniques such as principal component analysis or linear discriminants and non-parametric models have been transferred to functional data [15, 22, 42]. Several proposals to extend nonlinear feedforward networks to functional inputs have also been investigated. Commonly, functional data is only present in the first layer of the network, and the models differ in the way in which the input functions are internally evaluated. In [17], it is shown that a specific one-hidden layer functional network provides universal approximation of nonlinear continuous operators. The input function is evaluated within the network at several points and the weights of the network are contained in a finite-dimensional vector space. The approach [3] extends this result to multiple nonlinear operators. Multiple nonlinear operators provide an important tool to simulate systems which, based on specific parameter choices, show fundamentally different dynamical behavior within the same system. The contribution [80] investigates an alternative formulation of functional networks where on the base layer dot products of input functions and functional weights are computed. In practice, these dot products are evaluated numerically. In the extended version [79], approximation completeness of the model and consistency for numerical integration and universal function approximators is proved. In (Rossi/Conan-Guez, this volume), an application of this model to chemometric data is presented, thereby tackling the problem of possibly missing data. The contributions (Delannay/Rossi/Conan-Guez/Verleysen, this volume) and (Rossi/Conan-Guez/ElGolli, this volume) extend the same idea to alternative neural network models, radial basis function networks and the self-organizing map.

2.2 Unsupervised models

Multidimensional scaling, ISOMAP, ISODATA, and related tools constitute popular unsupervised techniques for data visualization and clustering. For data visualization, these methods aim at projecting given data points to low dimensions whereby the pairwise distances are preserved as far as possible. Data clustering partitions a given set of data into clusters based on similarity. Since the given data is only indirectly characterized by pairwise proximity values, the methods can be directly utilized to non-standard data provided an appropriate similarity measure is defined. Thereby, no specific mathematical properties such as symmetry are to be fulfilled to apply the algorithms. Reasonable similarity measures are provided by kernels as described in the next section. Finally, for pairwise clustering there are mean-field methods which can be used to iteratively compute cluster-membership weights [39].

The self-organizing map constitutes another popular unsupervised method for data visualization and clustering. Apart from an appropriate notion of similarity for non-standard data, SOM training requires a concept of how to adapt prototypes within a given non-standard domain. A very elegant and general solution has been proposed by Kohonen in [52]: a batch-SOM algorithm can be applied to any type of data by setting the prototypes to the generalized median, i.e. a point in the training set which minimizes a generalized quantization error. In [52], this approach is used to visualize proteins based on a similarity measure which punishes mismatches in pairwise aligned sequences. At the same time, Günter and Bunke [31] extended the SOM algorithm to

ESANN'2004 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 28-30 April 2004, d-side publi., ISBN 2-930307-04-8, pp. 281-292

attributed graphs by means of the edit distance and a generalization of the weighted mean of a set of graphs [49]. In [29, 47] simple competitive learning algorithms for clustering weighted graphs are proposed.

2.3 Kernel methods

Training and classification of the support vector machine can be formulated in terms of $k(x, x^i)$, x^i being a support vector and k being the kernel. Thus, a modification of the kernel to more complex data allows to transfer the SVM to more complex domains. Thereby, the resulting classification model can be interpreted as a linear classifier in a high dimensional feature space if the kernel decomposes into $k(x, y) = \Phi(x) \cdot \Phi(y)$. This is valid for positive semi-definite kernels. Analogously, other kernel based methods such as kernel principal component analysis rely solely on an appropriate choice of k and the task thus reduces to the design of kernels for structured data.

Various different methods to extend kernels to non-standard data have recently been proposed. [26] gives an overview about kernels for structures. Here we use a slightly different taxonomy and we distinguish three types of kernels: kernels based on common substructures, kernels based on a statistical model, and kernels based on local transformations of data. Note that this decomposition is not fully disjoint: the string kernel which is based on the data structure can also be interpreted as a kernel derived from a specific generative model [82].

2.3.1 Kernels based on common substructures

In the articles [37, 102], the basic principle of composite kernels is introduced. Simple kernels defined on subparts of given structures can be extended by generic operations to complex, convolutional kernels. In particular, strong closure properties for positive definite kernels hold which allow to easily construct problem specific versions.

As a consequence of this general proposal, kernels which are based on the comparison of primitives of given structures have been proposed for different data types. For sequences, the string kernel and variants count the number of occurrences of common substrings of limited length k. Thereby, the approaches differ with respect to the weighting of matches, whether partial matches are allowed, and whether substrings need to be contiguous [59, 63]. Since direct computation of these kernels is complex, much effort is put on efficient computation schemes. Dynamic programming or suffix-trees are two alternative techniques within this context. Further improvement of the efficiency and accuracy of the approaches can be obtained when using words instead of single symbols for document classification [14] or extending the methods to transduction tasks [103]. The approach [89] introduces the locality improved kernel, which also takes local correlations of neighbored sequence entries into account, but, unlike the string kernel, has been proposed only for fixed structures.

The more general data structure of directed acyclic graphs can be addressed in a similar way, counting the number of matching or partially matching subtrees of two given structures as proposed in [19, 94, 108] and also in (Micheli/Portera/Sperduti, this volume). The approach [20] adapts the general idea of matching subtrees to a kernel which compares two specific labeled acyclic graphs which come from a limited description language. As an alternative, string kernels can directly be applied to the prefix representation of trees as proposed in [98].

For graphs, the situation is more difficult since determining matching substructures is a complex problem. In (Geibel/Jain/Wysotzki, this volume), graphs are compared

as a whole by the Schur-Hadamard inner product which measures the similarity of the connection structure and labels of graphs. The NP-hard problem of finding an optimum matching is approximated by a heuristic which need not yield a valid kernel. An alternative graph kernel is proposed in [51]: labels on randomly generated paths of infinite length are compared. Thereby, efficient computation is possible by means of a fixed-point equation.

2.3.2 Kernels based on a statistical model

An alternative point of view to define kernels for non-standard data relies on semantic information about the data and represents data by feature vectors derived from generative models. The Fisher kernel constitutes an early and very prominent approach within this line [41]. A probabilistic model is fitted to the data and input structures are represented by the finite dimensional gradients of the log likelihood of the model at the respective data point. Thus, the generation process of the data is captured through the model and compared in this approach. Usually, the Fisher information metric is used as dot product in the feature space since it describes the Riemannian metric on the space of models. It can be shown that the Fisher kernel is good if the class information is contained as latent variable. Often, the Fisher kernel is used in combination with hidden Markov models in the context of sequential data [89]. Alternative stochastic models might be appropriate for alternative domains, such as a mixture of probabilistic principal components as proposed in [88]. The approach [96] proposes the tangent vector of log-odds or TOP-kernel, which is very similar to the Fisher kernel, but directly derived from a classification model. Here, the class information is contained as a class variable and the TOP-kernel compares favorably to the Fisher kernel in an application to biological sequences [89]. Other alternatives to the Fisher kernel can be derived from the general model of marginalized kernels as described in [51].

Several approaches which fit a separate probabilistic model to each data point and which compare these probabilistic models have also been proposed. In [66], a Gaussian model is fitted to each data point and the distributions are compared using Kulback-Leibler divergence. This method is used for audio- and image-data. A similar procedure is presented in [54] for sets of vectors. Here Gaussian distributions are fitted to the sets and their affinity serves as the kernel.

2.3.3 Kernels based on local transformations

The basic idea of the diffusion kernel as introduced in [53] is to extend known local similarity of objects, e.g. a neighborhood structure given by valid local transformation steps, to a global kernel imitating a diffusion process. The main algebraic tool is matrix exponentiation, to iterate the generator square matrix H which describes the local neighborhood structure of the given data. The approach [53] proposes to use the negative Laplacian as generator H for the generic setting of an undirected graph as local neighborhood structure. [58] extends the diffusion kernel introduced over discrete neighborhood structures to general Riemannian manifolds. The diffusion kernel has been applied to document processing whereby the generator H is induced by co-occurrence information [50], and to bioinformatics data whereby the generator H links genes which participate in successive reactions in metabolic pathways [97].

ESANN'2004 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 28-30 April 2004, d-side publi., ISBN 2-930307-04-8, pp. 281-292

3 Recursive models

Recursive models decompose the structures into constituents and recursively process the basic parts. Thereby, the already processed data sets a context for further computation, such that the single parts can be integrated to a whole structure. Basically, two different directions of recursive processing can be distinguished: partially recurrent system which dynamic is driven by the data structure, and fully recurrent systems which can be seen as complex discrete or continuous time dynamic systems.

3.1 Partially recurrent systems

Simple recurrent networks constitute a well-established tool for time series data. Assume x_t denotes the sequence entry at time point t. Then the dynamic is given by the equation $c_t = f(x_t, c_{t-1})$ whereby f is some function computed by the network, and c_t denotes the network state at time point t. A more detailed overview of recurrent network models can be found in [56]. This dynamic can immediately be generalized to more complex recursive structures. Recursive networks as presented in [24, 30, 33, 90] process tree structures as inputs. Given a binary tree t with root label x and subtrees l and r, the state of the network c_t after processing t is defined as $c_t = f(x, c_l, c_r)$. Recurrent and recursive models are well investigated mainly for supervised learning.

3.1.1 Supervised models

Supervised recurrent network training faces some problems, and complex dynamic behavior has to be dealt with. A tutorial overview of various aspects concerning learnability, dynamical properties, training algorithms, etc. can be found in [36]. A prime application of recurrent networks is language learning and a very clear discussion about possibilities and restrictions to learn languages is given in [10]. Further prototypical training schemes and analysis have been presented e.g. in [78, 85].

As already mentioned, recursive networks enlarge the dynamic of recurrent models to tree structures and they are trained by an adaptation of back-propagation. Thus, they share most of the dynamic properties and difficulties of simple recurrent networks [33, 36]. Various alternative training schemes have recently been adapted to recursive networks [8, 18, 90] and widespread successful applications of recursive networks can be found in the literature such as theorem proving [30], discourse representation theory [13], picture processing [18], document image classification [21], connectivity prediction for molecules [100], natural language parsing [92], protein structure prediction [73], and chemistry [8]. Thereby, recursive networks compete with kernel methods, see [106] and (Micheli/Portera/Sperduti, this volume).

The dynamic introduced so far mirrors the causality of time series data or tree structures. Spatial data and acyclic graphs constitute generalizations thereof. They can be encoded as time series or tree structures by specifying an order of the vertices. However, potential loss of information and dependencies of vertices are introduced by this procedure. Several generalizations of basic recursive models to better fit these type data have been proposed: recursive networks for acyclic graphs ([7] and (Bianchini/Maggini/Sarti/Scarselli, this volume)), bicausal networks for protein secondary structure prediction [4], an extension of recursive networks to lattices applied to protein contact map prediction [73], and contextual models for graph structures [65]. These adaptations extend the scope of information available at one recursive processing step according to the given data structure and yield improved accuracy.

ESANN'2004 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 28-30 April 2004, d-side publi., ISBN 2-930307-04-8, pp. 281-292

3.1.2 Unsupervised models

Recently, an increasing interest in unsupervised recursive processing of structured data can be observed, see e.g. [2, 5] for overviews about this topic. The aim of these approaches is to obtain visualization and clustering tools for temporal signals, spatial data, and also more complex structures. In principle, the dynamics of unsupervised models can be borrowed from the supervised case: $c_t = f(x_t, c_{t-1})$ for sequences and $c_t = f(x, c_l, c_r)$ for trees. Unlike in the supervised case, however, the concrete choice of the function f and the network activation c_t is less obvious. Unsupervised models do not compute an explicit output. Thus the activation c_t can be interpreted in different ways e.g. as best matching neuron or as distance profile computed for the whole map. Most unsupervised recursive models have been proposed only for temporal data, and they obey a simple dynamics given by leaky integrators or traveling waves [16, 74, 104]. The recursive SOM constitutes a more powerful though computationally quite complex model which relies on the whole map activation [99]. Efficient compression schemes using characteristics of only the winner neuron have been proposed in [32, 91] which achieve comparable results as the recursive SOM but which are computationally much more efficient. Thereby, the SOM for structured data [32] constitutes the first recursive SOM which has also been proposed for tree structured data. Recently, a general dynamics of recursive models which subsumes most of the above approaches has been proposed [34, 35]. Based on this general framework, important mathematical properties and comparisons of the models can be investigated.

3.2 Fully recurrent systems

This section focuses on recurrent systems for solving graph matching problems (GMP). The GMP refers to finding a structure preserving correspondence between the vertices of two different graphs such that some similarity function is maximized [81, 86]. Finding such correspondences is an NP-hard combinatorial problem [27]. Therefore and due to its wide applicability several approximate solutions for the GMP have been proposed. A multitude of methods originate from the neural network community.

Considerable interest in solving combinatorial optimization problems (COP) by means of neural networks has been initiated by the seminal paper of Hopfield and Tank [40]. Following this work, the general approach to solve COPs maps the objective function of the optimization problem onto an energy function of the network. The constraints of the problem are included in the energy function as penalty terms, such that global minima of the energy function correspond to optimal solutions of the COP. Thus, in the context of graph matching, the constituents of a solution are match hypotheses of pairs of vertices. The recurrent network dynamic aims at converging to a stable coalition of active neurons representing a maximal set of compatible matches.

Optimizing networks for GMPs can be roughly divided into two major groups: a quite intuitive direction poses the graph matching problem as that of recovering a structure preserving permutation matrix. An alternative direction transforms the matching problem to a maximum clique problem in an association graph. Approaches which do not fall in these realms are, for example, the dynamic link architecture [57], extensions of associative memories for storing and retrieving graphs [55, 64], or self-organizing winner-takes-all classifiers for structures [46].

ESANN'2004 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 28-30 April 2004, d-side publi., ISBN 2-930307-04-8, pp. 281-292

3.2.1 Recovering Permutation Matrices

A permutation matrix is a matrix representation of an injective mapping between the vertices of two graphs. If the graphs being matched are of order n and m, resp., then the permutation matrix is a $n \times m$ matrix with rows and columns summing to one or zero. The entries of the permutation matrix determine the correspondences between the vertices of both graphs. Early work uses binary threshold units and fixed penalty terms to express the constraints [61, 62, 68]. These approaches suffer from infeasible solutions and instable convergence properties. In the work [72, 76, 87, 93, 107] the graph matching problem is casted on the more principled statistical physics setting in terms of the mean-field theory and combined with self-amplification, softmax and penalty terms to improve solution quality and convergence properties.

Almost all approaches are concerned with determining the similarity of two graphs by means of minimizing an energy function that is quadratic in the assignment variables which are subject to a two way constraint. Two conceptual extensions to that issue are noteworthy to mention: Suganthan et al. [93] relaxed the two way constraints imposed on the permutation matrix to an one way constraint for retrieving several occurrences of a model in a scene in parallel. Another extension due to Finch et al. [23] extremized a non-quadratic energy function for graph matching to compute the similarity of two graphs and rectify structural errors at the same time.

3.2.2 Association graph techniques

The second strand of activities in neural graph matching is based on an idea originating from computer vision. Ambler [1], Barrow and Burstall [6], and Levi [60] suggested to transform the graph matching problem to the maximum clique problem (MCP) in a so-called association graph, a product structure derived from the graphs. Association graph techniques have been applied to several graph matching problems [11, 71, 69, 77]. To meet the requirements of practical applications [9, 48, 83, 84, 95], weights are annotated to the vertices and edges of an association graph to express the similarities between pairs of items of both graphs being matched. This recasts the graph matching problem to the maximum weighted clique problem (MWCP).

Wysotzki [105] applied a Hopfield-style network for approximately solving the MCP in an association graph. Since then this approach has been applied to learn structural prototypes of chemical compounds [83], to predict mutagenicity [84], and to similarity based recognition of segmented images [9]. Neural solutions which solely focus on the MCP can be found in [25, 43, 45, 101]. Recently, [44] and (Jain/Wysotzki, this volume) proposed a Hopfield clique network such that the global and local minima of the energy function are in one-to-one correspondence with the maximum weighted and maximal cliques, respectively.

Recently, Pelillo [69] used the Replicator dynamics for solving the MCP within the same framework. The Replicator dynamics is derived from evolutionary game theory [38]. It uses the Motzkin-Strauss formulation of the maximum clique problem [67] and its spurious-free extensions [12, 28, 70]. This formulation allows us to transform the maximum clique problem onto the problem of extremizing a quadratic form. This method has successfully been applied to match articulated and deformed shapes described by shock trees [71].

ESANN'2004 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 28-30 April 2004, d-side publi., ISBN 2-930307-04-8, pp. 281-292

4 Conclusions

This contribution provides a brief overview about neural network techniques applied to non-standard data, i.e. data which are not represented in terms of static feature vectors. Non-vectorial representations such as trees and graphs are often better suited to capture functional, structural, or other complex informations inherent in real world data. Standard vectorial feature-based representation is usually problem specific, prone to information loss or, alternatively, the curse of dimensionality. Structured representations, on the other hand, allow to store data structures of different sizes and complexity in a natural way whereby information loss is prohibited and, at the same time, the number of parameters can be limited. As reported, several successful applications of structure based networks in various different areas of applications such as chemistry, bioinformatics, or natural language processing have been developed.

However, structure based approaches often suffer from analytical poverty or computational intractability since standard analytical methods cannot easily be transferred to more complex structures or discrete optimization problems arise as subproblems during training. Apparently for these reasons, neural networks for non-standard data are still – despite their importance and potential applicability – widely unexplored. Nevertheless, an emerging interest in structure based models can be observed in the recent years, in particular in the context of kernel methods. Comparisons of different structure based methods have been conducted in the literature, such as (Micheli/Portera/Sperduti, this volume) and quite interesting approaches combine the best of different structure based methods such as (Geibel/Jain/Wysotzki, this volume).

References

[1] A. Ambler, H. Barrow, C. Brown, R. Burstall, and R. J. Popplestone. A versatile computer-controlled assembly system. IJCAI 1973.

[2] A.F.R. Araújo and A. Barreto. Context in temporal sequence processing: A self-organizing approach and its application to robotics. IEEE Transactions on Neural Networks 13(1):45-57, 2002.

[3] A.D. Back and T. Chen. Universal approximation of multiple nonlinear operators by neural networks. Neural Computation 14:2561-2566, 2003.

[4] P. Baldi, S. Brunak, P. Frasconi, G. Pollastri, and G. Soda. Bidirectional dynamics for protein secondary structure prediction. R. Sun, C.L. Giles (eds.), Sequence Learning: Paradigms, Algorithms, and Applications, pp. 80-104, Springer, 2001.

[5] A. Barreto, A.F.R. Araújo, and S.C. Kremer. A taxonomy for spatiotemporal connectionist networks revisited: the unsupervised case. Neural Computation 15:1255-1320, 2003.

[6] H. Barrow and R. Burstall. Subgraph isomorphism, matching relational structures and maximal cliques. Information Processing Letters, 4:83–84, 1976.

[7] M. Bianchini, M. Gori, and F. Scarselli. Processing directed acyclic graphs with recursive neural networks. IEEE Transactions on Neural Networks 12(6):1464-1470, 2001.

[8] A.M. Bianucci, A. Micheli, A. Sperduti, and A. Starita. Application of cascade correlation networks for structures to chemistry. Journal of Applied Intelligence, 12:117-146, 2000.

[9] S. Bischoff, D. Reuss, and F. Wysotzki. Applied connectionist methods in computer vision to compare segmented images. KI 2003: Advances in Artificial Intelligence. LNAI 2821, pp. 312–326. Springer, 2003.

[10] M. Bodén and J. Wiles. On learning context-free and context-sensitive languages. IEEE Transactions on Neural Networks 13(2):491-493, 2002.

[11] R. Bolles and P. Horaud. 3DPO: A three dimensional part orientation system. International Journal of Robotic Research, 5(3):3–26, 1986.

[12] I. Bomze. Evolution towards the maximum clique. Journal of Global Optimization, 10:143–164, 1997.

[13] A. Bua, M. Gori, and F. Santini. Recursive neural networks applied to discourse representation theory. ICANN'02.

[14] N. Cancedda, E. Gaussier, C. Goutte, and J.-M. Renders. Word-sequence kernels. Journal of Machine Learning Research 3:1059-1082, 2003.

ESANN'2004 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 28-30 April 2004, d-side publi., ISBN 2-930307-04-8, pp. 281-292

[15] H. Cardot, J.H. Ferraty, and P. Sarda. Functional linear model. Statist. & Prob. Letters 45:11-22, 1999.

[16] G. Chappell and J. Taylor. The temporal Kohonen map. Neural Networks 6:441-445, 1993.

[17] T. Chen and H. Chen. Universal approximation to nonlinear operators by neural networks with application to dynamic systems. IEEE Transactions on Neural Networks 6(4):911-917, 1995.

[18] S.-Y. Cho, Z. Chi, W.-C. Siu, and A.C. Tsoi. An improved algorithm for learning long-term dependency problems in adaptive processing of data structures. IEEE Transactions on Neural Networks 14(4):781-793, 2003.

[19] M. Collins and N. Duffy. Convolution kernels for natural languages. NIPS'2001.

[20] C. Cumby and D. Roth. On kernel methods for relational learning. ICML'2003.

[21] M. Diligenti, P. Frasconi, and M. Gori. Hidden tree Markov models for document image classification. IEEE Transactions on Pattern Analysis and Machine Intelligence, 25(4):519-523, 2003.

[22] F. Ferraty and P. Vieu. The functional nonparametric model and applications to chemometric data. Computational Statistics 17(4), 2002.

[23] A. Finch, R. Wilson, and E. Hancock. An energy function and continuous edit process for graph matching. Neural Computation, 10(7):1873–1894, 1998.

[24] P. Frasconi, M. Gori, and A. Sperduti. A general framework of adaptive processing of data structures. IEEE Transactions on Neural Networks, 9(5):768-786, 1998.

[25] N. Funabiki, Y. Takefuji, and K. Lee. Comparisons of energy-descent optimization algorithms for maximum clique. IEICE Trans. Fundamentals, E79-A(4):452–460, 1996.

[26] T. Gärtner. A survey of kernels for structured data. SIGKDD explorations 2003.

[27] M. Garey and D. Johnson. Computers and Intractability: A Guide to the Theory of NP-Completeness. W.H. Freeman and Company, New York, 1979.

[28] L. Gibbons, D. Hearn, P. Pardalos, and M. Ramana. Continuous characterizations of the maximum clique problem. Mathematics of Operations Research, 22:754–768, 1997.

[29] S. Gold and A. Rangarajan and E. Mjolsness. Learning with preknowledge: clustering with point and graph matching distance measures. NIPS, 1995.

[30] C. Goller A connectionist approach for learning search control heuristics for automated deduction systems. PhD thesis, Technische Universität München, 1997.

[31] S. Günter and H. Bunke. Self-organizing map for clustering in the graph domain. Pattern Recognition Letters, 23:401–417, 2002.

[32] M. Hagenbuchner, A. Sperduit, and A.C. Tsoi. A self-organizing map for adaptive processing of structured data. IEEE Transactions on Neural Networks 14(3):491-505, 2003.

[33] B. Hammer. Learning with recurrent neural networks. Springer, 2000.

[34] B. Hammer, A. Micheli, A. Sperduti, and M. Strickert. A general framework for unsupervised processing of structured data. To appear in Neurocomputing.

[35] B. Hammer, A. Micheli, A. Sperduti, and M. Strickert. Recursive self-organizing network models. Submitted to Neural Networks (invited).

[36] B. Hammer and J.J. Steil. Perspectives on learning with recurrent networks. ESANN'2002.

[37] D. Haussler. Convolutional kernels on discrete structures. Technical report, Department of Computer Science, University of Caloifornia at Santa Cruz, 1999.

[38] J. Hofbauer and K. Sigmund. Evolutionary Games and Population Dynamics. Cambridge University Press, UK, 1998.

[39] T. Hofmann and J.M. Buhmann, Pairwise data clustering by deterministic annealing. IEEE Transactions on Pattern Analysis and Machine Intelligence, 19(1):1-14, 1997.

[40] J. Hopfield and D. Tank. Neural computation of decisions in optimization problems. Biological Cybernetics, 52:141–152, 1985.

[41] T. Jaakkola, M. Diekhans, and D. Haussler. A discriminative framework for detecting remote protein homologies. Journal of Computational Biology 7(1/2), 2000.

[42] G.M. James and T.J. Hastie. Functional linear discriminant analysis of irregularly sampled curves. Journal of the Royal Statistical Society Series B 63:533-550, 2001.

[43] A. Jagota. Approximating maximum clique with a Hopfield network. IEEE Trans. Neural Networks, 6:724–735, 1995.

[44] B. Jain and F. Wysotzki. The maximum ω-clique problem and the Hopfield ω-clique model. Submitted to Neural Computation.

[45] B. Jain and F. Wysotzki. Fast winner-takes-all networks for the maximum clique problem. KI 2002: Advances in Artificial Intelligence. LNAI 2479, Springer, 2002.

[46] B. Jain and F. Wysotzki. A competitive winner-takes-all architecture for classification and pattern recognition of structures. In 4th IAPR International Workshop, GbRPR 2003. LNCS 2726, Springer, 2003.

[47] B. Jain and F. Wysotzki. Structural perceptrons for attributed graphs. SSPR 2004. Submitted for publication.
[48] B. Jain and F. Wysotzki. Central clustering in the domain of graphs. Machine Learning: Special Issue on Theoretical Advances in Data Clustering, 2004. Accepted for publication.
[49] X. Jiang, A. Münger and H. Bunke. On median graphs: properties, algorithms, and applications. IEEE Transactions on Pattern Analysis and Machine Intelligence, 23(10):1144–1151, 2001.
[50] J. Kandola, J. Shawe-Taylor, and N. Cristianini. Learning semantic similarity. NIPS'2002.
[51] H. Kashima, K. Tsuda, and A. Inokuchi. Marginalized kernels between labeled graphs. ICML'2003.
[52] T. Kohonen and P. Somervuo. How to make large self-organizing maps for nonvectorial data. Neural Networks 15(8-9):945-952, 2002.
[53] R.I. Kondor and J. Lafferty. Diffusion kernels on graphs and other discrete input spaces. ICML'2002.
[54] R. Kondor and T. Jebara. A kernel between sets of vectors. ICML'2003.
[55] R. Kree and A. Zippelius. Recognition of topological features of graphs. Journal of Physics A: Mathematical and General, 21:813–818, 1988.
[56] S.C. Kremer. Spatiotemporal connectionist networks: a taxonomy and review. Neural Computation 13:249-306, 2001.
[57] M. Lades, J. Vorbrüggen, J. Buhmann, J. Lange, C. von der Malsburg, R. Würtz, and W. Konen. Distortion invariant object reognition. IEEE Transactions on Computers, 42(3), 1993.
[58] J. Lafferty and G. Lebanon. Information diffusion kernels. NIPS'2002.
[59] C. Leslie, E. Eskin, J. Weston, and W.S. Noble. Mismatch string kernels for SVM protein classification. NIPS'2002.
[60] G. Levi. A note on the derivation of maximal common subgraphs of two directed or undirected graphs. Calcolo, 9:341–352, 1972.
[61] W. Li and N. Nasrabadi. Object recognition based on graph matching implemented by a Hopfield–style neural network. IJCNN 1989.
[62] W. Lin, F. Liao, C. Tsao, and T.Lingutla. A hierarchical multipe-view approach to three-dimensional object recognition. IEEE Transaction on Neural Networks, 2(1):84–91, 1991.
[63] H. Lodhi, J. Shawe-Taylor, N. Cristianini, C. Watkins. Text classification using string kernels. Journal of Machine Learnin Research 2:419-444, 2002.
[64] C.v.d. Malsburg. Pattern recognition by labeled graph matching. Neural networks, 1:141–148, 1988.
[65] A. Micheli, D. Sona, A. Sperduti. Contextual processing of structured data by recursive cascade correlation. Submitted to IEEE Transactions on Neural Networks.
[66] P.J. Moreno, P.P. Ho, and N. Vasconcelos. A Kullback-Leibler divergence based kernel for SVM classification in multimedia applications. NIPS'2002.
[67] T. Motzkin and E. Strauss. Maxima for graphs and a new proof of a theorem of Turan. Canadian Journal of Mathematics, 17:533–540, 1965.
[68] N. Nasrabadi and W. Li. Object recognition by a hopfield neural network. IEEE Transactions on Systems, Man, and Cybernetics, 21(6):1523–1535, 1991.
[69] M. Pelillo. Replicator equations, maximal cliques, and graph isomorphism. Neural Computation, 11(8):1933–1955, 1999.
[70] M. Pelillo and A. Jagota. Feasible and infeasible maxima in a quadratic program for maximum clique. Journal of Artificial Neural Networks, 2:411–420, 1995.
[71] M. Pelillo, K. Siddiqi, and S. Zucker. Matching hierarchical structures using association graphs. IEEE Transactions on Pattern Analysis and Machine Intelligence, 21(11):1105–1120, 1999.
[72] C. Peterson and B. Soderberg. A new method for mapping optimisation problems. International Journal of Neural Systems, 1:2–33, 1989.
[73] G. Pollastri, P. Baldi, A. Vullo, and P. Frasconi. Prediction of protein topologies using GIOHMMs and GRNNs. NIPS'2002.
[74] J. Principe, N. Euliano, and S. Garani. Principles and networks for self-organization in space-time. Neural Networks 15(8-9):1069-1084, 2002.
[75] J. Ramsay and B. Silverman. Functional data analysis. Springer Series in Statistica, 1997.
[76] A. Rangarajan and E. Mjolsness. A Lagrangian relaxation network for graph matching. IEEE Transactions on Neural Networks, 7(6):1365–1381, 1996.
[77] J. Raymond, E. Gardiner, and P. Willett. Heuristics for similarity searching of chemical graphs using a maximum common edge subgraph algorithm. Journal of Chemical Information and Computer Sciences, 42(2):305–316, 2002.
[78] P. Rodriguez. Simple recurrent networks learn context-free and context-sensitive languages by counting . Neural Computation 13:2093-2118, 2001.
[79] F. Rossi and B. Conan-Guez. Functional mutli-layer perceptrons: a nonlinear tool for functional data analysis. CEREMADE preprint 0331, 2003, http://www.ceremade.dauphine.fr

ESANN'2004 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 28-30 April 2004, d-side publi., ISBN 2-930307-04-8, pp. 281-292

[80] F. Rossi, B. Conan-Guez, and F. Fleuret. Theoretical properties of functional multi layer perceptrons. ESANN'2002.

[81] A. Sanfeliu and K. Fu. A distance measure between attributed relational graphs for pattern recognition. IEEE Transactions on Systems, Man, and Cybernetics, 13:353–362, 1983.

[82] C. Saunders, J. Shawe-Taylor, and A. Vinokourov. String kernels, Fisher kernels and finite state automata. NIPS'2002.

[83] K. Schädler and F. Wysotzki. Application of a neural net in classification and knowledge discovery. ESANN 1998.

[84] K. Schädler and F. Wysotzki. Comparing structures using a Hopfield-style neural network. Applied Intelligence, 11:15–30, 1999.

[85] J. Schmidhuber, F. Gers, D. Eck. Learning nonregular languages: a comparison of simple recurrent networks and LSTM. Neural Computation 14:2039-2041, 2002.

[86] L. Shapiro and R. Haralick. A metric for comparing relational descriptions. IEEE Transaction on Pattern Analysis and Machine Intelligence, 7(1):90–94, 1985.

[87] P. Simic. Constrained nets for graph matching and other quadratic assignement problems. Neural Computation, 3:268–281, 1991.

[88] G. Siolas and F. d'Alché-Buc. Mixtures of probabilistic PCAs and Fisher kernels for word and document modeling. ICANN'2002.

[89] S. Sonnenburg, G. Rätsch, A. Jagota, and K.-R. Müller. New methods for splice site recognition. ICANN'2002.

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

[91] M. Strickert and B. Hammer. Neural gas for sequences. WSOM'2003.

[92] P. Sturt, F. Costa, V. Lombardo, and P. Frasconi. Learning first-pass structural attachment preferences with dynamic grammars and recursive neural networks. Cognition, 88(2):133-169, 2003.

[93] P. Suganthan, E. Teoh, and D. Mital. Pattern recognition by graph matching using Potts MFT networks. Pattern Recognition, 28:997–1009, 1995.

[94] J. Suzuki, Y. Sasaki, and E. Maeda. Kernels for structured natural language data. NIPS'2003.

[95] A. Torsello and E. Hancock. Efficiently computing weighted tree edit distance using relaxation labeling. EMMCVPR'2001. LNCS 2134, Springer, 2001.

[96] K. Tsuda, M. Kawanabe, G. Rätsch, S. Sonnenburg, and K.-R. Müller. A new discriminative kernel from probabilistic models. Neural Computation 14:2397-2414, 2002.

[97] J.-P. Vert and M. Kanehisa. Graph-driven features extraction from microarray data using diffusion kernels and kernel CCA. NIPS'2002.

[98] S.V.N. Vishwanathan and A.J. Smola. Fast kernels for string and tree matching. NIPS'2002.

[99] T. Voegtlin. Recursive self-organizing maps. Neural Networks 15(8-9):979-992, 2002.

[100] A. Vullo and P. Frasconi. Disulfide Connectivity Prediction Using Recursive Neural Networks and Multiple Alignments. To appear in Bioinformatics.

[101] R. Wang, Z. Tang, and Q. Cao. An efficient approximation algorithm for finding a maximum clique using Hopfield network learning. Neural Computation, 15(7):1605–1619, 2003.

[102] C. Watkins. Dynamic alignment kernels. Technical report, Department of Computer Science, Royal Holloway, University of London, 1999.

[103] J. Weston, C. Leslie, D. Zhou, A. Elisseeff, and W.S. Noble. Semi-supervised protein classification using cluster kernels. NIPS'2003.

[104] J.C. Wiemer. The time-organized map algorithm: extending the self-organizing map to spatiotemporal signals. Neural Computation 15:1143-1171, 2003.

[105] F. Wysotzki. Artifical intelligence and artifical neural nets. Neural Informatics, 12/1989, Akademie der Wissenschaften der DDR, Berlin, GDR, 1989.

[106] Y. Yao, G.L. Marcialis, M. Pontil, P. Frasconi, and F. Roli. Combining flat and structured representations for fingerprint classification with recursive neural networks and support vector machines. Pattern Recognition, 36(2): 397-406,2003.

[107] A. Yuille and J. Kosowsky. Statistical physics algorithms that converge. Neural Computation, 6(3):341–356, 1994.

[108] D. Zelenko, C. Aone, and A. Richardella. Kernel methods for relational extraction. Journal of Machine Learning Research 3:1083-1106, 2003.