ESANN'2000 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 26-28 April 2000, D-Facto public., ISBN 2-930307-00-5, pp. 213-218

Limitations of Hybrid Systems

Barbara Hammer,

University of Osnabrück, Dept. of Mathematics/Comp. Science,
Albrechtstraße 28, 49069 Osnabrück, Germany

Abstract. We examine the ability of combining symbolic and subsymbolic approaches by means of recursively encoding and decoding structured data. We show that encoding of symbolic data is possible in this way – hence neural networks seem well suited for control or classification in symbolic approaches – whereas decoding requires an increasing complexity of the decoding function – hence networks with this dynamics are not adequate for producing structured data. Real labeled tree structures reject a smooth encoding in general.

1. Introduction

In many areas of application data possesses both symbolic or structured elements and real valued or numerical attributes; for example web sites contain textual information and links to other sites, chemical structures can be described by single elements and their connection, pictures can be represented by a special arrangement of simple graphical objects, ... [2]. Furthermore in purely symbolic areas the main information lies in the structural aspects – characterizing a formula by only enumerating the single variables and symbols seems not appropriate. Hence there is an increasing interest in neural networks dealing with structured data. Including the structural aspects of the data leads to better performance compared to standard subsymbolic approaches [12] and enables subsymbolic adaptation in symbolic domains in a natural way [7].

Here we focus on methods which combine symbolic and subsymbolic components by means of some in general adaptive encoding and decoding of the respective data – terms, formulas, and tree structures on the one side and real vectors of fixed dimension on the other side. Frequently, the encoding and decoding process applies a standard neural network recursively to the data where the recursion corresponds to the recursiv estructure of the input or output, respectively. An early description of this paradigm are Hinton's distributed reduced descriptors [5], Pollack's RAAM [11], and Plate's holographic reduced representation [10]. Recent extensions include LRAAM [13] or folding and recurrent networks [7,14], the former encoding general trees, the latter lists.

We consider the ability of encoding and decoding with such mechanisms in principle. After a formal definition of the dynamics it is shown that encoding of symbolicdata to a fixed dimensional vector can be performed in this way whereas decoding of vectors in to terms or formulas requires an increasing complexity of the decoding function.If tree structured data with real valued labels is dealt with, every proper encoding and decoding is necessarily a trivial encoding in the worst case. Since we focus on the computation dynamics the results hold for every mechanism which uses the same dynamics no matter whether and how it is trained. Due to space limitations we will omit all but one proof.

2. Definition of the dynamics

Σ_k^* denotes the set of trees with labels in $\Sigma \subset \mathbb{R}^m$ where every node except the empty node $\perp$ possesses exactly k successors. This structure covers symbolic terms in a natural way: the labels come from a finite alphabet denoting the function symbols, the subtrees represent the subterms of the function symbol. If Σ denotes a real vector space, hybrid data like web sites is covered: the labels represent features of the basic objects such as the size of a html document, the tree structure is given by the interconnection of the data, i.e. the links.

A nonempty tree is denoted by $a(t_1, \ldots, t_k)$ where a is the root's label and t_1, ..., t_k are the k subtrees. Mimicking the recursive nature of trees constitutes a natural way of defining an encoding and decoding of the data, i.e., encoding starts at the leaves and recursively encodes the subtrees of a tree with some simple encoding function, decoding recursively applies a simple decoding function to a vector in order to obtain the label of a node and codes for the k subtrees until we arrive at the leaves (see Fig. 1); formally:

Definition 1 *A function* enc : $\Sigma \times (\mathbb{R}^m)^k \to \mathbb{R}^m$ *and initial context* $s \in \mathbb{R}^m$ *induces an* **encoding function** $\mathcal{E}_s^{\text{enc}} : \Sigma_k^* \to \mathbb{R}^m$ *where*

$$\mathcal{E}_s^{\text{enc}}(t) = \begin{cases} s & \text{if } t = \perp\,, \\ \text{enc}(a, \mathcal{E}_s^{\text{enc}}(t_1), \ldots, \mathcal{E}_s^{\text{enc}}(t_k)) & \text{if } t = a(t_1, \ldots, t_k)\,. \end{cases}$$

A function $\text{dec} = (\text{dec}_0, \text{dec}_1, \ldots, \text{dec}_k) : \mathbb{R}^m \to \Sigma \times (\mathbb{R}^m)^k$ *and final set* $F \subset \mathbb{R}^m$ *induces a* **decoding function** $\mathcal{D}_F^{\text{dec}} : \mathbb{R}^m \to \Sigma_k^*$ *where*

$$\mathcal{D}_F^{\text{dec}}(x) = \begin{cases} \perp & \text{if } x \in F\,, \\ \text{dec}_0(x)(\mathcal{D}_F^{\text{dec}}(\text{dec}_1(x)), \ldots, \mathcal{D}_F^{\text{dec}}(\text{dec}_k(x))) & \text{otherwise}\,. \end{cases}$$

This dynamics is common in neural network literature dealing with hybrid systems: In the LRAAM enc and dec are standard feedforward networks which are trained such that the composition produces the identity [13]. In Plate's approach enc and dec are fixed mappings, sums of convolution and correlation [10]. Folding and recurrent neural networks combine an encoding part given by a feedforward network with a further network mapping the encoded trees to the outputs [7,14]. Both parts are trained simultaneously for the specific task.

3. Encoding and decoding of symbolic data

First we consider purely symbolic data, i.e. $\Sigma = \{1, \ldots, B\}$. We ask the question whether these trees can be encoded into a finite dimensional vector space with the proposed dynamics. I.e., does a mapping enc exist such that $\mathcal{E}_s^{\text{enc}}$ is injective on all trees or on the subset of trees of some arbitrary but restricted

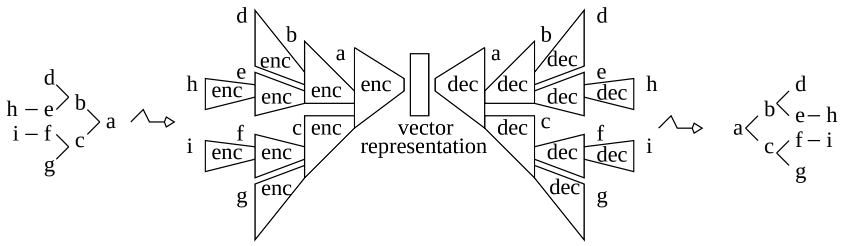

Figure 1: *Example for the encoding/deoding dynamics: Applying a mapping* enc *recursively to the single nodes and the already encoded subtrees yields a representation for the tree as a vector of fixed dimension. Applying a mapping* dec *recursively to the vector outputs the labels and codes for the subtrees.*

height? If so, what is the complexity of enc? This is of interest if the approach is to be applied to learning tasks involving logical formulas, for example.

Theorem 2 *There exists a mapping* $\text{enc} : \Sigma \times (\mathbb{R}^2)^k \to \mathbb{R}^2$ *and* $s \in \mathbb{R}^2$ *such that* $\mathcal{E}_s{}^{\text{enc}}$ *is injective. For every* $T \in \mathbb{N}$ enc *can be approximated by a feedforward network such that the induced encoding is injective on trees up to height* T*. The number of neurons is independent of* T*, the activation can be any function with nonzero and continuous second derivative in the neighborhood of one point.*

Hence injective encoding is possible in principle and requires a limited amount of resources which is inpedendent of the maximum input height of the trees. Furthermore, this dynamics has some nice properties: It leads to universal approximators of mappings $\Sigma_k^* \to \mathbb{R}^n$ if the output is combined with a standard feedforward network [4]. It has been successfully applied in several applications [7,12] indicating the ability of storing the relevant attributes of a tree in such a way that the information can be used in further neural processing. However, when dealing with very large trees the finite precision of the computation and inherent noise reduce the computational properties to the power of at most tree automata [8,9,14] and numerical problems may arise [1].

Now the question arises as to whether the above decoding dynamics allows a reconstruction of the encoded trees with a neural network. In general this depends on the concrete encoding function. First we consider a special function enc and show that trees of a restricted and almost linear structure allow proper decoding. One interesting special case are linear trees, i.e. lists.

Theorem 3 *Assume that for every* $t \in S \subset \Sigma_k^*$ *the number of leaves is restricted by* b*. Then* $\text{enc} : \Sigma \times (\mathbb{R}^{2b})^k \to \mathbb{R}^{2b}$*,* $\text{dec} : \mathbb{R}^{2b} \to \Sigma \times (\mathbb{R}^{2b})^k$*, and* $s \in \mathbb{R}^{2b}$ *exist such that* $\mathcal{D}_{\{s\}}^{\text{dec}} \circ \mathcal{E}_s{}^{\text{enc}}$ *yields the identity on* S*. For every finite subset* S' *in* S enc *and* dec *can be approximated by a feedforward network such that the identity on* S' *is approximated. The number of neurons only depends on* b *and* k*. The activation function can be any squashing function with a nonvanishing and continuous second derivative in the neighborhood of one point.*

Hence encoding and decoding is possible for almost linear trees and lists with the proposed dynamics; the RAAM and LRAAM, for example, can succeed if applied to a corresponding learning task. Of course, this does not hold automatically for every mechanism where enc and dec are not trained but defined a priori or for every activation function. A network with linear activation functions cannot solve the decoding task with a limited number of neurons.

The argumentation for this fact uses a combinatorial quantity of a network architecture: The **VC-dimension** of a function class $\mathcal{F} : X \to \{0,1\}$ is the largest number of points $x_1, \ldots, x_l \in X$ which are shattered by $\mathcal{F}$, i.e. for every mapping $d : \{x_1, \ldots, x_l\} \to \{0,1\}$ some function $f \in \mathcal{F}$ exists with $f|\{x_1, \ldots, x_l\} = d$. If $\mathcal{F}$ maps to $\mathbb{R}$ we consider $\{g \mid \exists f \in \mathcal{F}, g(x) = H(f(x) - 0.5)\}$ instead. The VC-dimension caracterizes information theoretic learnability. Besides, it measures in some sence the richness or complexity of the class and hence can be used to obtain lower bounds for the parameters.

Theorem 4 *If 2^T points in $\mathbb{R}^m$ are to be mapped to 2^T sequences of length T which approximate all binary sequences of length T (i.e., a value > 0.5 corresponds to the label 1, a value < 0.5 corresponds to the label 0) with some function $\mathcal{D}_Y^{\text{dec}}$ where* dec *is a feedforward network with linear activation function then* dec *possesses $\Omega((T/\ln T)^{1/3})$ neurons.*

Note that this result does not rely on the fact how the sequences are encoded as points in $\mathbb{R}^m$. Furthermore, nearly all reasonable definitions of how the decoded values are interpreted do not affect the result. Hence recursive linear networks are not appropriate for decoding of symbolic data with a linear structure in principle. In concrete applications usually the sigmoidal activation function is used which allows a proper encoding and decoding of almost linear trees. Decoding of general trees requires an increasing number of neurons:

Theorem 5 *Assume points in $\mathbb{R}^m$ exist which are approximately decoded to all binary trees of height at most T with labels in $\{0,1\}$ with some $\mathcal{D}_Y^{\text{dec}}$. If* dec *is a feedforward network, the number of neurons is bounded by $2^{\Omega(T)}$ if the activation function is the standard sigmoidal function or piecewise polynomial.*

Proof: dec $= (\text{dec}_0, \text{dec}_1, \text{dec}_2) : \mathbb{R}^m \to \mathbb{R} \times \mathbb{R}^m \times \mathbb{R}^m$ gives rise to a recurrent network $\mathcal{E}_s^{\text{enc}} : \mathbb{R}_1^* \to \mathbb{R} \times \mathbb{R}^m$ induced by enc $: \mathbb{R} \times \mathbb{R} \times \mathbb{R}^m \to \mathbb{R} \times \mathbb{R}^m$,

$$(x, y, z) \mapsto (\text{dec}_0(z), (1 - x) \cdot \text{dec}_1(z) + x \cdot \text{dec}_2(z))\,.$$

If $\mathcal{D}_F^{\text{dec}}$ maps the value z to some tree t then $\pi_1 \circ \mathcal{E}^{\text{enc}}_{(0,z)}$, π_1 being the projection to the first component, maps any binary sequence of length i to some node in the ith level of the tree t; the exact number of the node depends on the sequence: $[0, \ldots, 0]$ is mapped to the leftmost node in the ith level, $[1, \ldots, 1]$ is mapped to the rightmost node, the other sequences lead to the nodes in between. The last component is not relevant; see Fig. 3.

If points in $\mathbb{R}^m$ exist which are approximately mapped to all trees of height T in $\{0,1\}_2^*$ with $\mathcal{D}_F^{\text{dec}}$ then the neural architecture $\pi_1 \circ \mathcal{E}^{\text{enc}}_{(0,\_)}$ shatters all binary sequences of length T with last component 1: One can simply choose the second part of the initial context corresponding to a vector z which encodes a tree of height T and leaves according to the dichotomy.
enc can be computed adding a constant number of neurons with some at

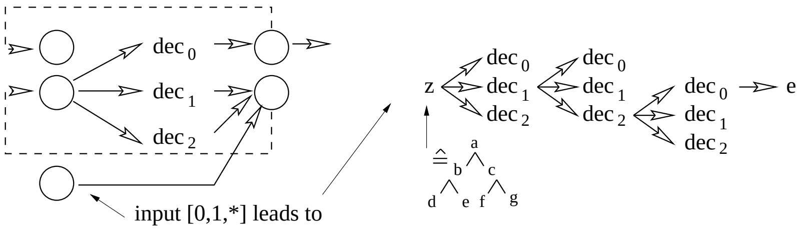

Figure 2: *An appropriate input to the recurrent network as defined in Theorem 5 restores a path of the tree $\mathcal{D}_Y^{\mathrm{dec}}(z)$, the length of the input sequence indicates the length of the path, the entries 0 and 1 stand for the left or right subtree, respectively,* dec_0 *yields the output label.*

most quadratic activation function and it can be approximated arbitrarily well adding a constant number of sigmoidal units. Consequently, the VC-dimension of $\pi_1 \circ \mathcal{E}^{\mathrm{enc}}_{(0,\_)}$ restricted to inputs of height at most T is limited by $\mathrm{O}(N^3 T \ln(qd))$ if the activation function in dec is piecewise polynomial with at most q pieces and degree at most $d \geq 2$. The VC-dimension is limited by $\mathrm{O}(N^4 T^2)$ if the activation function is the standard sigmoidal function [3,6]. In both cases N denotes the number of neurons in dec. The lower bound 2^{T-1} for the VC-dimension leads to the bound $N = 2^{\Omega(T)}$ for the neurons in dec. □

Note that it is not important how the trees are encoded. Furthermore, a more sophisticated decoding of the single binary nodes or using other standard activation functions leads to the same bound since for any reasonable modification the VC-dimension of the recurrent network as defined in the proof is still bounded by some polynomial in the number of nodes and maximum input height. Consequently, the decoding formalism requires an increasing amount of resources even for purely symbolic data. Hence a formalism like the LRAAM can deal only with restricted situations, i.e. almost linear trees or limited height, whereas these restrictions do not apply to methods which merely focus on the encoding, like recurrent and folding networks. This constitutes a motivation for the success of folding networks in practice compared to the LRAAM [12].

4. Structured data with real valued labels

Up to now we have considered the encoding problem from a set theoretical point of view, dropping the question whether similarity of trees is mirrored by a small euclidian distance of the codes. If data with real labeled nodes is to be encoded, the possibility of similarity preserving encoding is essential for the encoding and decoding in general. Continuous data can only partially be recovered if the vector encoding is nested. Unfortunately, results from topology tell us that proper encoding of real valued data is not possible in general:

Theorem 6 *Assume* $\mathrm{enc} : [0,1] \times (\mathbb{R}^m)^k \to \mathbb{R}^m$ *is continuous. Then for every tree structure with more than m nodes trees t and t' of this structure and at least one pair of corresponding labels l in t and $1-l$ in t' exist with $\mathcal{E}_y^{\mathrm{enc}}(t) = \mathcal{E}_y^{\mathrm{enc}}(t')$.*

This result does not rely on the special dynamics but only on topological issues. Hence hybrid data with real valued nodes cannot be encoded in general. However, since for an a priori limited number of real values direct encoding is possible in an obvious way, the ability of the above dynamics to deal with symbolic data and restricted real valued information as input can be stated.

5. Conclusion

The ability of combining symbolic and subsymbolic methods in principle has been investigated. We have focused on approaches which encode and decode the respective data such that the processing mirrors the recursive nature of the data as proposed in [2,5,7,10,11,13]. Encoding of symbolic data is possible in principle. In contrast, decoding requires an increasing number of resources, unless very restricted data is dealt with. Encoding of real valued trees is not possible in general. Hence neural networks seem well suited if applied as a control mechanism or classification tool to classical symbolic approaches. Adequate recovering of structured data requires a more sophisticated dynamics. However, in a concrete application the possibility of learning an adequate dynamic rather than the in principle existence is to be examined additionally.

References

[1] Bengio, Y., Simard, P. & Frasconi, P. (1994) Learning long-term dependencies with gradient descent is difficult *IEEE Transactions on Neural Networks* **5**(2):157-166.
[2] Frasconi, P., Gori, M. & Sperduti, A. (1997) A general framework for adaptive processing of data sequences. *IEEE Transactions on Neural Networks* **9**(5):768-786.
[3] Hammer, B. (1999) On the learnability of recursive data. *MCSS* **12**:62-79.
[4] Hammer, B. (1999) On the Approximation Capability of Recurrent Neural Networks. To appear in *Neurocomputing*.
[5] Hinton, G. E. (1990) Mapping part-whole hierarchies into connectionist networks. *AI* **46**(1-2):47-75.
[6] Koiran, P. & Sontag, E. D. (1998) Vapnik-Chervonenkis dimension of recurrent neural networks. *Discrete Applied Mathematics* **86**:63-79.
[7] Küchler, A. & Goller, C. (1996) Inductive learning symbolic domains using structure-driven neural networks. In G. Görz and S. Hölldobler (eds.), *KI-96: Advances in AI*, pp.183-197. Berlin: Springer.
[8] Maass, W. & Orponen, P. (1998) On the effect of analog noise in discrete-time analog computation. *Neural Computation* **10**(5):1071-1095.
[9] Maass, W. & Sontag, E.D. (1999) Analog neural nets with Gaussian or other common noise distributions cannot recognize arbitrary regular languages. *Neural Computation* **11**:771-782.
[10] Plate, T. (1995) Holographic reduced representations. *IEEE Transactions on Neural Networks* **6**(3):623-641.
[11] Pollack, J. (1990) Recursive distributed representation. *AI* **46**(1-2):77-106.
[12] Schmitt, T. & Goller, C. (1998) Relating chemical structure to activity with the structure processing neural folding architecture. In *Engineering Applications of Neural Networks*, Gibraltar.
[13] Sperduti, A. (1994) Labeling RAAM. *Connection Science* **6**(4):429-459.
[14] Tino, P., Horne, B.G., Giles, C.L. & Colligwood, P.C. (1998) Finite state machines and recurrent neural networks. In J.E. Dayhoff and O. Omidvar (eds.), *Neural Networks and Pattern Recognition*, pp. 171-220. Academic Press.