ESANN'1999 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 21-23 April 1999, D-Facto public., ISBN 2-600049-9-X, pp. 33-38

Approximation Capabilities of Folding Networks

Barbara Hammer,

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

Abstract. In this paper we show several approximation results for folding networks – a generalization of partial recurrent neural networks such that not only time sequences but arbitrary trees can serve as input: Any measurable function can be approximated in probability. Any continuous function can be approximated in the maximum norm on inputs with restricted height, but the resources necessarily increase at least exponentially in the input height. In general, approximation on arbitrary inputs is not possible in the maximum norm.

1. Introduction

When dealing with structured data, for example, formulas, terms, sequences, graphs, etc., some kind of recurrence can be found in most cases. This a priori unlimited recurrence suggests using recurrent connections if a neural network deals with such data, for example, in a classification task. Indeed, partial recurrent networks are a natural tool which works on sequences of a priori unlimited length [3]. Furthermore, they can naturally be generalized to so called folding networks [4] such that arbitrary trees may serve as inputs. Since trees are a very general data structure folding networks offer the possibility to use subsymbolic methods in various symbolic domains, e.g. theorem proving or classification of chemical formulas [4, 9]. Additionally, the nodes of the trees may be labeled with symbolic or subsymbolic data, i.e. discrete or continuous values, such that folding networks are capable to deal with hybrid data as well which occur naturally in several domains [2].

Here the question occurs whether folding networks are universal approximators of mappings from trees into a real vector space. This is a necessary condition for any formalism to succeed if a function with structured inputs is to be learned. We will show that folding networks are capable of approximating any measurable function on trees with real valued labels into a real vector space *in probability* even if the weights, the number of layers, and the encoding dimension are restricted. On the contrary, mappings exist that cannot be approximated *in the maximum norm* for inputs with *unlimited height* even if the formalism of folding networks is somehow extended. If the *maximum input height is restricted* any mapping can be approximated, but the resources increase exponentially in the input height.

ESANN'1999 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 21-23 April 1999, D-Facto public., ISBN 2-600049-9-X, pp. 33-38

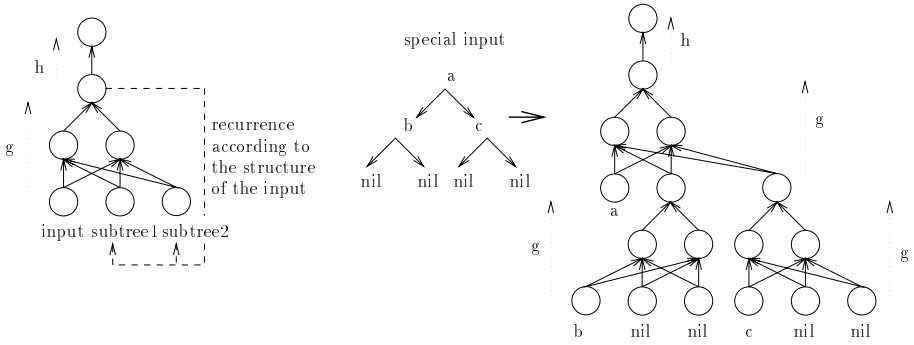

Figure 1: Example for a computation by a folding network: The recurrent part g (left) becomes unfolded according to a specific input tree (right).

2. The architecture

We assume that the labels are in $\mathbb{R}^m$. Denote by $(\mathbb{R}^m)_k^*$ the set of trees with labels in $\mathbb{R}^m$ where each node has at most k successors, and by $(\mathbb{R}^m)_k^{\leq T}$ the restriction to trees of height at most T. The empty node is denoted with nil. Any nonempty tree t is denoted with $t = a(t_1, \ldots, t_k)$ if a is the root of t and t_i are the k subtrees. After substituting all open places with nil if necessary, we can assume that any node except nil has exactly k successors. Due to the recursive definition of trees we can define an **induced** recursive mapping $\tilde{f}_{\mathbf{y}} : (\mathbb{R}^m)_k^* \to \mathbb{R}^l$ for any $f : \mathbb{R}^{m+k\cdot l} \to \mathbb{R}^l$ and **initial context** $\mathbf{y} \in \mathbb{R}^l$, where

$$\begin{aligned} \tilde{f}_{\mathbf{y}}(\text{nil}) &= \mathbf{y} \\ \tilde{f}_{\mathbf{y}}(a(t_1, \ldots, t_k)) &= f(a, \tilde{f}_{\mathbf{y}}(t_1), \ldots, \tilde{f}_{\mathbf{y}}(t_k)) \,. \end{aligned}$$

Definition 1 *A mapping $f : (\mathbb{R}^m)_k^* \to \mathbb{R}^n$ is computed by a* **folding network** *if a number l, the* **encoding dimension**, *an initial context $\mathbf{y} \in \mathbb{R}^l$, and mappings $g : \mathbb{R}^{m+k\cdot l} \to \mathbb{R}^l$, $h : \mathbb{R}^l \to \mathbb{R}^n$ exist which can be computed by standard feedforward neural networks such that $f = h \circ \tilde{g}_{\mathbf{y}}$.*

A folding network consists of two parts, a part induced by g which is applied recursively according to the structure of the input tree and which encodes the input tree in a real vector, and a feedforward part h which maps the encoded tree to an output vector. One example for a computation is depicted in Fig.1. For $k = 1$ we obtain partial recurrent networks with sequences as inputs.

3. Approximation in probability

We call a mapping $f : (\mathbb{R}^m)_k^* \to \mathbb{R}^n$ measurable or continuous if and only if any restriction of f to input trees with a fixed structure is measurable or continuous, respectively. Let P be a probability measure on $(\mathbb{R}^m)_k^*$. A property of a function holds locally if it is valid in the neighborhood of at least one point. A function is C^n if it is n times continuously differentiable. All requirements we

ESANN'1999 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 21-23 April 1999, D-Facto public., ISBN 2-600049-9-X, pp. 33-38

will pose on the activation functions are fulfilled in particular for the standard sigmoidal activation.

Theorem 2 *For any measurable function $f : (\mathbb{R}^m)_k^* \to \mathbb{R}^n$ and $\epsilon > 0$ a folding network $h \circ \tilde{g}_{\mathbf{y}}$ exists such that*

$$P(t \in (\mathbb{R}^m)_k^* \mid |f(t) - h \circ \tilde{g}_{\mathbf{y}}(t)| > \epsilon) < \epsilon \,.$$

The encoding dimension l can be chosen as 2. h can be chosen as a multilayer network with one hidden layer, locally Riemann integrable and nonpolynomial hidden activation function, and linear outputs, g can be chosen as a multilayer network with $O(\log k)$ layers with an activation which is locally C^2 with a non-vanishing second derivative, and which is additionally a squashing function in the first hidden layer, i.e. monotonous, $\lim_{x\to\infty} f(x) = 1$, $\lim_{x\to-\infty} f(x) = 0$.

Proof: In a first step f is reduced to a discrete mapping: Since f restricted to trees of a fixed structure can be approximated by a continuous mapping in probability with arbitrary precision, we can assume that f is continuous. Since $(\mathbb{R}^m)_k^*$ can be written as the countable union of trees with a fixed height and limited labels, we can find a maximum height T and a positive number B such that $P((\mathbb{R}^m)_k^* \setminus]\text{-}B, B[^m)_k^{\leq T})$ is arbitrarily small. It is sufficient to approximate $f := f|]\text{-}B, B[^m)_k^{\leq T}$. Since f is equicontinuous, we can find $\delta > 0$ such that for trees t_1 and t_2 with the same structure and labels the coefficients of which differ at most δ from each other, $|f(t_1) - f(t_2)| < \epsilon/2$. $]\text{-}B, B[$ can be decomposed into disjoint intervals of diameter at most δ: $I_1 =]b_0, b_1[$, ..., $I_q =]b_{q-1}, b_q[$ such that the probability of trees with some coefficient in $\{b_0, \ldots, b_q\}$ is arbitrarily small. This defines a discretization of f.

In the following step, the recursive part of the network is constructed such that the single labels are encoded via the corresponding interval numbers, and the entire tree is encoded in a real number where the encoded labels are written in prefix order: $g_1 : (x_1, \ldots, x_m) \mapsto 2 + \sum_{i=1}^m q^{i-1} \sum_{j=1}^q (j \cdot 1_{I_j}(x_i) - 1)$ maps $]\text{-}B, B[^m$ to $\{2, \ldots, q^m + 1\}$ such that the image encodes the intervals uniquely where the coefficients of $\mathbf{x}$ belong to. The characteristic function 1_{I_j} can be computed via a step function $H(x) = 0$, if $x < 0$, and $H(x) = 1$, otherwise. Define $d = \lceil \log(q^m + 1) \rceil$ and $g_2 : \mathbb{R}^{1+k\cdot 2} \to \mathbb{R}^2$,

$$\begin{aligned} g_2(x, x_1^1, x_2^1, \ldots, x_1^k, x_2^k) \;=\; & ((0.1)^d \cdot (1 + (0.1)^d x + (0.1)^d x_1^1 + (0.1)^{2d} x_2^1 x_1^2 + \\ & \ldots + (0.1)^{kd} x_2^1 \ldots x_2^{k-1} x_1^k, (0.1)^{d(k+1)} x_2^1 \ldots x_2^k) \,. \end{aligned}$$

g_2 implements somehow the concatenation of k strings x_1^1, ..., x_1^k. The mapping $g_2 \circ g_1$ can be computed with a multilayer feedforward network with $\mathrm{O}(\log k)$ layers, activation x, x^2 (substituting products), and H (first layer). Starting with $(0, 0)$ it induces the mapping of a tree $t = a(t_1, \ldots, t_n)$ to its prefix representation $(0.0 \ldots 01 g_1(a) \mathrm{repr.}(t_1) \ldots \mathrm{repr.}(t_k), (0.1)^{\text{length of repr.}-d})$.

In a third step the activations x, x^2, and H are substituted except for a set of arbitrarily small probability. Because the activation σ in g is locally C^2, we can approximate $x = \lim_{\epsilon\to 0}(\sigma(x_0 + \epsilon x) - \sigma(x_0))/\epsilon\sigma'(x_0))$ and $x^2 = \lim_{\epsilon\to 0}(\sigma(x_1 +$

ESANN'1999 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 21-23 April 1999, D-Facto public., ISBN 2-600049-9-X, pp. 33-38

$\epsilon x) + \sigma(x_1 - \epsilon x) - 2\sigma(x_1))/\epsilon^2\sigma''(x_1))$ uniformly on compact intervals for some x_0 and x_1, and $H(x) = \lim_{\epsilon\to\infty}\sigma(\epsilon x)$ for a squashing activation σ uniformly outside a neighborhood of 0. One can find different ϵ in the above formulas such that the image of trees of height at most T and labels in the intervals $]b_i + \alpha, b_{i+1} - \alpha[$ for arbitrarily small α consists of disjoint intervals J_j. Two trees are mapped to the same interval J_j if and only if their respective labels are contained in the same intervals I_i. The resulting mapping g can be computed as stated in the theorem. The linear terms in the above quotients are integrated in the weights, which changes the initial context from $(0,0)$ to $\mathbf{y} = (\sigma(x_0), \sigma(x_0))$.

In a last step h is constructed such that it maps each interval J_i obtained via $\tilde{g}_{\mathbf{y}}$ to a representative value where trees which are encoded in J_i are mapped to under f with a maximum deviation of $\epsilon/2$. h can be chosen as described in the theorem because of [7]. □

Note that the number of neurons used in this construction depends on the fan-out k and the number of intervals I_i. In particular, it is limited by $O(k \cdot \log k + pn)$ if a mapping on a finite set of p trees with purely symbolic labels is to be interpolated as proved in [6]. Furthermore, results from the feedforward case lead to the following modification:

Corollary 3 *For any measurable function $f : (\mathbb{R}^m)_k^* \to \mathbb{R}^n$ exists a folding network $h \circ \tilde{g}_{\mathbf{y}}$ which approximates f in probability. The encoding dimension can be chosen as 2. h and g can be chosen as multilayer networks with one hidden layer, weights restricted by an arbitrary positive number B, locally Riemann integrable and nonpolynomial activations in g and the hidden layer of h, and linear outputs in h. Additionally, the output activation of g has to be locally homeomorphic (i.e. invertible, g, g^{-1} continuous).*

Proof: Because of Theorem 2 we can approximate f in probability with a function $F \circ \tilde{G}_{\mathbf{y}}$ where F and G are continuous. The relevant range of G is contained in a neighborhood of 0 and can be scaled and shifted such that the output activation σ of g is locally homeomorphic in this range. F and $(\sigma^{-1}, \sigma^{-1}) \circ G$ can be approximated on compact intervals with feedforward networks h or $\bar{g}$, respectively, with restricted weights, one hidden layer, and linear outputs [7]. h and $g = (\sigma, \sigma) \circ \bar{g}$ fulfill the conditions as stated above. □

Note that we could integrate in the above construction the linear outputs of $\bar{g}$ into the other parts of the network. As a consequence, we could drop the hidden layer in the recursive part, but this leads to an unlimited encoding dimension.

4. Approximation in the maximum norm

Assume we want to approximate a continuous mapping arbitrarily well on any input tree. The above construction fails because of two reasons: We have restricted the maximum input height, and the encoding of input trees in one real value is not continuous. The latter problem enforces to use an exponentially increasing encoding dimension in the worst case:

ESANN'1999 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 21-23 April 1999, D-Facto public., ISBN 2-600049-9-X, pp. 33-38

Theorem 4 *Choose $T \in \mathbb{N}$ and a compact set $B \subset \mathbb{R}^m$. For any continuous mapping $f : B_k^{\leq T} \to \mathbb{R}^n$ and $\epsilon > 0$ a folding network $h \circ \tilde{g}_{\mathbf{y}}$ exists such that $|h \circ \tilde{g}_{\mathbf{y}}(t) - f(t)| < \epsilon$ for all $t \in B_k^{\leq T}$. g can be a feedforward network without hidden layer and h a single hidden layer feedforward network with linear outputs. The other activations are locally Riemann integrable and nonpolynomial. If g is continuous, the encoding dimension increases at least exponentially in T for $k \geq 2$ and linearly in T for $k = 1$ for some ϵ and real valued f.*

Proof: Assume $K \geq 2$. $k = 1$ is analogous. If the encoding dimension is not limited it is easy to construct an encoding g such that $\tilde{g}_{\mathbf{y}}$ simply writes the single labels of an input tree of height T into one real vector of dimension $k^T(m+1)+1$. If the actual number of places which are already used, p, is encoded in the last dimension as $(0.1)^p$ and a is a number which is not contained in any label in B, an encoding is given by $h(\mathbf{x}, (\mathbf{x}_1^1, 0, \ldots), x_2^1, \ldots, (\mathbf{x}_1^k, 0, \ldots), x_2^k) = (a, \mathbf{x}, \mathbf{x}_1^1, \mathbf{x}_1^2, \ldots, \mathbf{x}_1^k, 0, \ldots, 0, x_2^1 \cdot \ldots \cdot x_2^k \cdot (0.1)^{m+1})$ where the exact places where the vectors $\mathbf{x}_1^2$, $\mathbf{x}_1^3 \ldots$ start can be computed with a finite gain using x_2^1, $x_2^2 \ldots$. Since the finite gain can be approximated for example with a sigmoidal network, the entire mapping g can be approximated with a single hidden layer network with linear outputs. The linearity can be integrated into the part h and the connections in g. h can be chosen such that it approximates the continuous mapping on these encoded trees in $\mathbb{R}^{k^T(m+1)} + 1$ to $\mathbb{R}^n$ [7].

Surprisingly, this brute force method is somehow the best possible encoding. Let an encoding dimension l be given. Assume g is continuous. Choose $\mathbf{x} \in B$ and $\epsilon > 0$ such that the ball of radius ϵ with center $\mathbf{x}$ is contained in B. Choose T with $mk^{T-1} > lk$. Assume, $a_1, \ldots, a_{mk^{T-1}}$ are different points in B. The mapping f where the image of a tree with height T, root a_i, and internal nodes a_1 is the $(i \bmod m) + 1$st coefficient of the $(i \operatorname{div} m) + 1$st leaf can be completed to a continuous mapping. The approximation $h \circ g(a_i, \tilde{g}_{\mathbf{y}}(t_1), \ldots, \tilde{g}_{\mathbf{y}}(t_k))$ on these trees decomposes into a mapping $\bar{g} : B^{k^{T-1}} \to \mathbb{R}^{lk}$ and $h \circ g$ where necessarily points $\mathbf{p}$ and $-\mathbf{p}$ in the sphere of radius ϵ and center $(\mathbf{x}, \ldots, \mathbf{x})$ in $B^{k^{T-1}}$ exist with $\bar{g}(\mathbf{p}) = \bar{g}(-\mathbf{p})$ because of the Theorem of Borsuk-Ullam [1]. Consequently, at least one value of $h \circ \tilde{g}_{\mathbf{y}}$ differs from the desired output at least $\epsilon/\sqrt{mk^{T-1}}$. An appropriate scaling leads to the distance ϵ. □

As a consequence, continuous mappings cannot be approximated in the maximum norm with limited resources for restricted inputs. Furthermore, the above mapping f which implements somehow the identity on the leafs demonstrates that an approximation of a mapping and its *derivative* in probability is not possible in general with a restricted encoding dimension l. This is due to the fact that the determinant of the Jacobian of an approximation $h \circ \tilde{g}_{\mathbf{y}}$ is 0 because the first part $\bar{g}$ as above maps into a low dimensional vector space.

The necessity to increase the encoding dimension shows that arbitrary continuous mappings cannot be approximated in the maximum norm on real labeled input trees with unlimited height with a continuous folding network. But even for purely symbolic inputs mappings exist which cannot be approximated with any reasonable network for arbitrary input length [5]. The argumentation

ESANN'1999 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 21-23 April 1999, D-Facto public., ISBN 2-600049-9-X, pp. 33-38

for this fact holds for sequences with binary labels, and consequently for trees with $k \geq 2$ and unary labels, too. The argumentation only uses the fact that the VC dimension of folding networks with limited resources is restricted by a polynomial in 2^T if T is the maximum input height. Therefore it even transfers to more general models, where for example the weights depend polynomially on the inputs. However, for special functions an approximation is possible as shown for example in [8, 10] for finite automata or tree automata, respectively.

5. Conclusion

It has been shown that folding networks are capable of approximating any reasonable function on real labeled trees in probability even with restricted resources, i.e. limited weights, encoding dimension, and number of hidden layers. This fact makes them well suited for application areas which deal with symbolic as well as subsymbolic data. However, several problems occur if an arbitrary mapping shall be approximated in the maximum norm. This indicates that an approximation of symbolic mappings with large inputs or of hybrid mappings that are sensitive to small changes of the labels may lead to very large networks.

References

[1] P. Alexandroff and H. Hopf. *Topologie*. Springer, 1974.

[2] P. Frasconi, M. Gori, and A. Sperduti. A general framework for adaptive processing of data structures. *IEEE Trans. on Neural Networks*, 9(5), 1998.

[3] C.L. Giles and M. Gori, editors. *Adaptive Processing of Sequences and Data Structure*. Springer, 1998.

[4] C. Goller. *A connectionist approach for learning search control heuristics for automated deduction systems*. PhD thesis, Technical University of Munich, 1997.

[5] B. Hammer. On the approximation capability of recurrent neural networks. In *International Symposium on Neural Computation*, 1998.

[6] B. Hammer and V. Sperschneider. Neural networks can approximate mappings on structured objects. In *2nd International Conference on Computational Intelligence and Neuroscience*, 1997.

[7] K. Hornik. Some new results on neural network approximation. *Neural Networks*, 6, 1993.

[8] A. Küchler. On the correspondence between neural folding architectures and tree automata. Technical report, University of Ulm, 1998.

[9] E. Schmitt and C. Goller. Relating chemical structure to activity with the structure processing neural folding architecture. In *Engineering Applications of Neural Networks*, 1998.

[10] P. Tino, B.G. Horne, C.L. Giles, and P.C. Collingwood. Finite state machines and recurrent neural networks – automata and dynamical systems approaches. In *Neural Networks and Pattern Recognition*. Academic Press, 1998.