Recurrent Neural Networks Are Universal Approximators

Anton Maximilian Schäfer[1,2] and Hans Georg Zimmermann[1]

[1] Information & Communications, Learning Systems
Siemens AG, Corporate Technology, 81739 Munich, Germany
{Schaefer.Anton.ext, Hans_Georg.Zimmermann}@siemens.com
[2] Department Optimisation and Operations Research, University of Ulm, 89069 Ulm, Germany

Abstract. Neural networks represent a class of functions for the efficient identification and forecasting of dynamical systems. It has been shown that feedforward networks are able to approximate any (Borel-)measurable function on a compact domain [1,2,3]. Recurrent neural networks (RNNs) have been developed for a better understanding and analysis of open dynamical systems. Compared to feedforward networks they have several advantages which have been discussed extensively in several papers and books, e.g. [4]. Still the question often arises if RNNs are able to map every open dynamical system, which would be desirable for a broad spectrum of applications. In this paper we give a proof for the universal approximation ability of RNNs in state space model form. The proof is based on the work of Hornik, Stinchcombe, and White about feedforward neural networks [1].

1 Introduction

Recurrent neural networks (RNNs) allow the identification of dynamical systems in form of high dimensional, nonlinear state space models. They offer an explicit modeling of time and memory [5].

In previous papers, e.g. [6], we discussed the modeling of open dynamical systems based on time-delay recurrent neural networks which can be represented in a state space model form. We solved the system identification task by finite unfolding in time, i.e., we transferred the temporal problem into a spatial architecture [6], which can be handled by error backpropagation through time [7]. Further we enforced the learning of the autonomous dynamics in an open system by overshooting [6]. Consequently our RNNs not only learn from data but also integrate prior knowledge and first principles into the modeling in form of architectural concepts. However, the question arises if the outlined RNNs are able to identify and approximate any open dynamical system, i.e., if they hold an universal approximation ability.

In 1989 Hornik, Stinchcombe, and White [1] could show that any Borel-measurable function on a compact domain can be approximated by a three-layered feedforward network, i.e., a feedforward network with one hidden layer, with an arbitrary accuracy. In the same year Cybenko [2] and Funahashi [3] found similar results, each with different methods. Whereas the proof of Hornik, Stinchcombe, and White [1] is based on the Stone-Weierstrass theorem, Cybenko [2] makes in principle use of the Hahn-Banach

S. Kollias et al. (Eds.): ICANN 2006, Part I, LNCS 4131, pp. 632–640, 2006.

und Riesz theorem. Funahashi [3] mainly applies the Irie-Miyake and the Kolmogorov-Arnold-Sprecher theorem.

Some work has already been done on the capability of RNN to approximate measurable functions, e.g. [8]. In this paper we focus on open dynamical systems and prove that those can be approximated by RNNs in state space model form with an arbitrary accuracy. We start with a short introduction on open dynamical systems and RNNs in state space model form (sec. 2). We further recall the basic results of the universal approximation theorem of Hornik, Stinchcombe, and White [1] (sec. 3). Subsequent we show that these results can be extended to RNNs in state space model form and we consequently give a proof for their universal approximation ability (sec. 4). We conclude with a short summary and an outlook on further research (sec. 5).

2 Open Dynamical Systems and Recurrent Neural Networks

Figure 1 illustrates an open dynamical system in discrete time which can be described as a set of equations, consisting of a state transition and an output equation [5,6]:

$$\begin{aligned} s_{t+1} &= g(s_t, u_t) \quad \text{state transition} \\ y_t &= h(s_t) \quad \text{output equation} \end{aligned} \tag{1}$$

The state transition is a mapping from the present internal hidden state of the system s_t and the influence of external inputs u_t to the new state s_{t+1}. The output equation computes the observable output y_t.

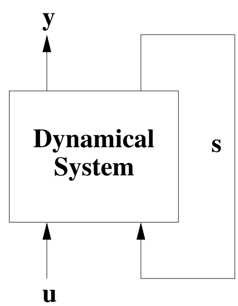

Fig. 1. Open dynamical system with input u, hidden state s and output y

The system can be viewed as a partially observable autoregressiv dynamic state transition $s_t \rightarrow s_{t+1}$ that is also driven by external forces u_t. Without the external inputs the system is called an autonomous system [5]. However, most real world systems are driven by a superposition of an autonomous development and external influences.

If we assume that the state transition does not depend on s_t, i.e., $y_t = h(s_t) = h(g(u_{t-1}))$, we are back in the framework of feedforward neural networks [6]. However, the inclusion of the internal hidden dynamics makes the modeling task much harder, because it allows varying inter-temporal dependencies. Theoretically, in the recurrent framework an event s_{t+1} is explained by a superposition of external inputs $u_t, u_{t-1}, \ldots$ from all the previous time steps [5].

In previous papers, e.g. [6], we proposed to map open dynamical systems (eq. 1) by a recurrent neural network (RNN) in state space model form

$$\begin{aligned} s_{t+1} &= f(As_t + Bu_t + \theta) \quad \text{state transition} \\ y_t &= Cs_t \quad \text{output equation} \end{aligned} \tag{2}$$

where A, B, and C are weight matrices of appropriate dimensions and θ is a bias, which handles offsets in the input variables u_t [5,6]. f is the so called activation function of the network which is typically sigmoidal (def. 3) like e.g., the hyperbolic tangent.

A major advantage of RNNs written in form of a state space model (eq. 2) is the explicit correspondence between equations and architecture. It is easy to see, that the set of equations (2) can be directly transferred into a spatial neural network architecture using so called finite unfolding in time and shared weight matrices A, B, and C [5,7]. Figure 2 depicts the resulting model [6].

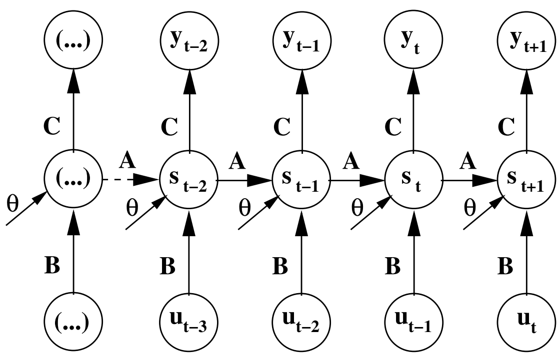

Fig. 2. Recurrent neural network unfolded in time

A more detailed description about RNNs in state space model form can be found in [4] or [6].

3 Universal Approximation Theorem for Feedforward Neural Networks

Our proof for RNNs in state space model form (sec. 4) is based on the work of Hornik, Stinchcombe und White [1]. In the following we therefore recall their definitions and main results:

Definition 1. *Let* A^I *with* $I \in \mathbb{N}$ *be the set of all affine mappings* $A(x) = w \cdot x - \theta$ *from* $\mathbb{R}^I$ *to* $\mathbb{R}$ *with* $w, x \in \mathbb{R}^I$ *and* $\theta \in \mathbb{R}$. *'·' denotes the scalar product.*

Transferred to neural networks x corresponds to the input, w to the network weights and θ to the bias.

Definition 2. *For any (Borel-)measurable function* $f(\cdot) : \mathbb{R} \to \mathbb{R}$ *and* $I \in \mathbb{N}$ *be* $\sum^I(f)$ *the class of functions*

$$\{NN : \mathbb{R}^I \to \mathbb{R} : NN(x) = \sum_{j=1}^{J} v_j f(A_j(x)), x \in \mathbb{R}^I, v_j \in \mathbb{R}, A_j \in \mathrm{A}^I, J \in \mathbb{N}\}. \tag{3}$$

Here NN stands for a three-layered feedforward neural network, i.e., a feedforward network with one hidden layer, with I input-neurons, J hidden-neurons and one output-neuron. v_j denotes the weights between hidden- and output-neurons. f is an arbitrary activation function (sec. 2).

Remark 1. *The function class* $\sum^I(f)$ *can also be written in matrix form*

$$NN(x) = vf(Wx - \theta) \tag{4}$$

where $x \in \mathbb{R}^I$, $v, \theta \in \mathbb{R}^J$ *and* $W \in \mathbb{R}^{J \times I}$.

In this context the computation of the function $f(\cdot) : \mathbb{R}^J \to \mathbb{R}^J$ *be defined component-wise, i.e.,*

$$f(Wx - \theta) := \begin{pmatrix} f(W_1 \cdot x - \theta_1) \\ \vdots \\ f(W_j \cdot x - \theta_j) \\ \vdots \\ f(W_J \cdot x - \theta_J) \end{pmatrix} \tag{5}$$

where W_j $(j = 1, \ldots, J)$ *denotes the* $j - th$ *row of the matrix* W.

Definition 3. *A function* f *is called a sigmoid function, if* f *is monotonically increasing and bounded, i.e.,*

$$f(a) \in [\alpha, \beta], \textit{ whereas } \lim_{a \to -\infty} f(a) = \alpha \textit{ and } \lim_{a \to \infty} f(a) = \beta \tag{6}$$

with $\alpha, \beta \in \mathbb{R}$ *and* $\alpha < \beta$. *In the following we define* $\alpha = 0$ *and* $\beta = 1$ *which bounds the sigmoid function on the interval* $[0, 1]$.

Definition 4. *Let* $\mathcal{C}^I$ *and* $\mathcal{M}^I$ *be the sets of all continuous and respectively all Borel-measurable functions from* $\mathbb{R}^I$ *to* $\mathbb{R}$. *Further denote* $\mathbb{B}^I$ *the Borel-*σ*-algebra of* $\mathbb{R}^I$ *and* $(\mathbb{R}^I, \mathbb{B}^I)$ *the I-dimensional Borel-measurable space.*

$\mathcal{M}^I$ contains all functions relevant for applications. $\mathcal{C}^I$ is a subset of it. Consequently, for every Borel-measurable function f the class $\sum^I(f)$ belongs to the set $\mathcal{M}^I$ and for every continuous f to its subset $\mathcal{C}^I$.

Definition 5. *A subset* S *of a metric space* (X, ρ) *is* ρ*-dense in a subset* T*, if there exists, for any* $\varepsilon > 0$ *and any* $t \in T$, $s \in S$*, such that* $\rho(s, t) < \varepsilon$.

This means that every element of S can approximate any element of T with an arbitrary accuracy. In the following we replace T and X by $\mathcal{C}^I$ and $\mathcal{M}^I$ respectively and S by $\sum^I(f)$ with an arbitrary but fixed f. The metric ρ is chosen accordingly.

Definition 6. *A subset S of $\mathcal{C}^I$ is uniformly dense on a compact domain in $\mathcal{C}^I$, if, for any compact subset $K \subset \mathbb{R}^I$, S is ρ_K-dense in $\mathcal{C}^I$, where for $f, g \in \mathcal{C}^I$ $\rho_K(f,g) \equiv \sup_{x \in K} |f(x) - g(x)|$.*

Definition 7. *Given a probability measure μ on $(\mathbb{R}^I, \mathbb{B}^I)$, the metric $\rho_\mu : \mathcal{M}^I \times \mathcal{M}^I \rightarrow \mathbb{R}^+$ be defined as follows*

$$\rho_\mu(f,g) = \inf\{\varepsilon > 0 : \mu\{x : |f(x) - g(x)| > \varepsilon\} < \varepsilon\}. \tag{7}$$

Theorem 1. ***(Universal Approximation Theorem for Feedforward Networks)*** *For any sigmoid activation function f, any dimension I and any probability measure μ on $(\mathbb{R}^I, \mathbb{B}^I)$, $\sum^I(f)$ is uniformly dense on a compact domain in $\mathcal{C}^I$ and ρ_μ-dense in $\mathcal{M}^I$.*

This theorem states that a three-layered feedforward neural network, i.e., a feedforward neural network with one hidden layer, is able to approximate any continuous function uniformly on a compact domain and any measurable function in the ρ_μ-metric with an arbitrary accuracy. The proposition is independent of the applied sigmoid activation function f (def. 3), the dimension of the input space I, and the underlying probability measure μ. Consequently three-layered feedforward neural networks are universal approximators.

Theorem 1 is only valid for feedforward neural networks with I input-, J hidden- and a single output-neuron. Accordingly, only functions from $\mathbb{R}^I$ to $\mathbb{R}$ can be approximated. However with a simple extension it can be shown that the theorem holds for networks with a multiple output (cor. 1).

For this, the set of all continuous functions from $\mathbb{R}^I$ to $\mathbb{R}^n$, $I, n \in \mathbb{N}$, be denoted by $\mathcal{C}^{I,n}$ and the one of (Borel-)measurable functions from $\mathbb{R}^I$ to $\mathbb{R}^n$ by $\mathcal{M}^{I,n}$ respectively. The function class $\sum^I$ gets extended to $\sum^{I,n}$ by (re-)defining the weights v_j ($j = 1, \ldots, J$) in definition 2 as $n \times 1$ vectors. In matrix-form the class $\sum^{I,n}$ is then given by

$$NN(x) = Vf(Wx - \theta) \tag{8}$$

with $x \in \mathbb{R}^I, \theta \in \mathbb{R}^J, W \in \mathbb{R}^{J \times I}$ and $V \in \mathbb{R}^{n \times J}$. The computation of the function $f(\cdot) : \mathbb{R}^J \rightarrow \mathbb{R}^J$ be once more defined component-wise (rem. 1).

In the following, function $g : \mathbb{R}^I \rightarrow \mathbb{R}^n$ has got the elements g_k, $k = 1, \ldots, n$.

Corollary 1. *Theorem 1 holds for the approximation of functions in $\mathcal{C}^{I,n}$ and $\mathcal{M}^{I,n}$ by the extended function class $\sum^{I,n}$. Thereby the metric ρ_μ is replaced by $\rho_\mu^n := \sum_{k=1}^n \rho_\mu(f_k, g_k)$.*

Consequently three-layered multi-output feedforward networks are universal approximators for vector-valued functions.

4 Universal Approximation Theorem for RNNs

The universal approximation theorem for feedforward neural networks (theo. 1) proves, that any (Borel-)measurable function can be approximated by a three-layered feedforward neural network. We now show, that RNNs in state space model form (eq. 2) are also universal approximators and able to approximate every open dynamical system (eq. 1) with an arbitrary accuracy.

Definition 8. *For any (Borel-)measurable function $f(\cdot) : \mathbb{R}^J \to \mathbb{R}^J$ and $I, n \in \mathbb{N}$ be $RNN^{I,n}(f)$ the class of functions*

$$\begin{aligned} s_{t+1} &= f(As_t + Bu_t - \theta) \\ y_t &= Cs_t \quad . \end{aligned} \tag{9}$$

Thereby be $u_t \in \mathbb{R}^I$, $s_t \in \mathbb{R}^J$ and $y_t \in \mathbb{R}^n$, with $t = 1, \ldots, T$. Further be the matrices $A \in \mathbb{R}^{J \times J}$, $B \in \mathbb{R}^{J \times I}$, and $C \in \mathbb{R}^{n \times J}$ and the bias $\theta \in \mathbb{R}^J$. In the following, analogue to remark 1, the calculation of the function f be defined component-wise, i.e.,

$$s_{t+1_j} = f(A_j s_t + B_j u_t - \theta_j), \tag{10}$$

where A_j and B_j $(j = 1, \ldots, J)$ denote the $j - th$ row of the matrices A and B respectively.

It is obvious, that the class $RNN^{I,n}(f)$ is equivalent to the RNN in state space model form (eq. 2). Analogue to its description in section 2 as well as definition 2, I stands for the number of input-neurons, J for the number of hidden-neurons and n for the number of output-neurons. u_t denotes the external inputs, s_t the inner states and y_t the outputs of the neural network. The matrices A, B, and C correspond to the weight-matrices between hidden- and hidden-, input- and hidden- and hidden- and output-neurons respectively. f is an arbitrary activation function.

Theorem 2. ***(Universal Approximation Theorem for Recurrent Neural Networks)***
Let $g : \mathbb{R}^J \times \mathbb{R}^I \to \mathbb{R}^J$ be measurable and $h : \mathbb{R}^J \to \mathbb{R}^n$ be continuous, the external inputs $u_t \in \mathbb{R}^I$, the inner states $s_t \in \mathbb{R}^J$, and the outputs $y_t \in \mathbb{R}^n$ $(t = 1, \ldots, T)$. Then, any open dynamical system of the form

$$\begin{aligned} s_{t+1} &= g(s_t, u_t) \\ y_t &= h(s_t) \end{aligned} \tag{11}$$

can be approximated by an element of the function class $RNN^{I,n}(f)$ (def. 8) with an arbitrary accuracy, where f is a continuous sigmoide activation function (def. 3).

Proof. The proof is given in two steps. Thereby the equations of the dynamical system are traced back to the representation by a three-layered feedforward network.

In the first step, we conclude that the state space equation of the open dynamical system, $s_{t+1} = g(s_t, u_t)$, can be approximated by a neural network of the form $\bar{s}_{t+1} = f(A\bar{s}_t + Bu_t - \theta)$ for all $t = 1, \ldots, T$.

Let now be $\varepsilon > 0$ and $f : \mathbb{R}^{\bar{J}} \to \mathbb{R}^{\bar{J}}$ be a continuous sigmoid activation function. Further let $K \in \mathbb{R}^J \times \mathbb{R}^I$ be a compact set, which contains $s_t, \bar{s}_t$ and u_t for all $t = 1, \ldots, T$. From the universal approximation theorem for feedforward networks (theo. 1) and the subsequent corollary (cor. 1) we know, that for any measurable function $g(s_t, u_t) : \mathbb{R}^J \times \mathbb{R}^I \to \mathbb{R}^J$ and for an arbitrary $\delta > 0$, a function

$$NN(s_t, u_t) = Vf(Ws_t + Bu_t - \bar{\theta}), \tag{12}$$

with weight matrices $V \in \mathbb{R}^{J\times \bar{J}}$, $W \in \mathbb{R}^{\bar{J}\times J}$ and $B \in \mathbb{R}^{\bar{J}\times I}$ and a bias $\bar{\theta} \in \mathbb{R}^{\bar{J}}$ exists, such that

$$\sup_{s_t,u_t\in K} |g(s_t,u_t) - NN(s_t,u_t)| < \delta \quad \forall\ t=1,\ldots,T. \tag{13}$$

As f is continuous and T finite, there exists a $\delta > 0$, such that according to the ε-δ-criterion we get out of equation (13), that for the dynamics

$$\bar{s}_{t+1} = Vf(W\bar{s}_t + Bu_t - \bar{\theta}) \tag{14}$$

the following condition holds

$$|s_t - \bar{s}_t| < \varepsilon \quad \forall\ t=1,\ldots,T. \tag{15}$$

Further let

$$s'_{t+1} := f(W\bar{s}_t + Bu_t - \bar{\theta}) \tag{16}$$

which gives us, that

$$\bar{s}_t = Vs'_t. \tag{17}$$

With the help of a variable transformation from $\bar{s}$ to s'_t and the replacement $A := WV (\in \mathbb{R}^{\bar{J}\times\bar{J}})$, we get the desired function on state s':

$$s'_{t+1} = f(As'_t + Bu_t - \bar{\theta}) \tag{18}$$

Remark 2. *The transformation from s to s' might involve an enlargement of the internal state space dimension.*

In the second step we show, that the output equation $y_t = h(s_t)$ can be approximated by a neural network of the form $\bar{y}_t = C\bar{s}_t$. Thereby we have to cope with the additional challenge, to approach the nonlinear function $h(s_t)$ of the open dynamical system by a linear equation $C\bar{s}_t$.

Let $\tilde{\varepsilon} > 0$. As h is continuous per definition, there exist an $\varepsilon > 0$, such that (according to the ε-δ-criterion) out of $|s_t - \bar{s}_t| < \varepsilon$ (eq. 15) follows, that $|h(s_t) - h(\bar{s}_t)| < \tilde{\varepsilon}$. Consequently it is sufficient to show, that $\hat{y}_t = h(\bar{s}_t)$ can be approximated by a function of the form $\bar{y}_t = C\bar{s}_t$ with an arbitrary accuracy. The proposition then follows out of the triangle inequality.

Once more we use the universal approximation theorem for feedforward networks (theo. 1) and the subsequent corollary (cor. 1), which gives us that equation

$$\hat{y}_t = h(\bar{s}_t) \tag{19}$$

can be approximated by a feedforward neural network of the form

$$\bar{y}_t = Nf(M\bar{s}_t - \hat{\theta}) \tag{20}$$

where $N \in \mathbb{R}^{n\times\hat{J}}$ and $M \in \mathbb{R}^{\hat{J}\times J}$ be suitable weight matrices, $f : \mathbb{R}^{\hat{J}} \to \mathbb{R}^{\hat{J}}$ a sigmoid activation function, and $\hat{\theta} \in \mathbb{R}^{\hat{J}}$ a bias. According to equation (17) and equation (18) we know that $\bar{s}_t = Vs'_t$ and $s'_{t+1} = f(As'_t + Bu_t - \bar{\theta})$. By insertion we get

$$\begin{aligned}\bar{y}_t &= Nf(M\bar{s}_t - \hat{\theta}) \\ &= Nf(MVs'_t - \hat{\theta}) \\ &= Nf(MVf(As'_{t-1} + Bu_{t-1} - \bar{\theta}) - \hat{\theta}) \quad .\end{aligned} \tag{21}$$

Using again theorem 1 equation (21) can be approximated by

$$\tilde{y}_t = Df(Es'_{t-1} + Fu_{t-1} - \tilde{\theta}) \quad , \tag{22}$$

with suitable weight matrices $D \in \mathbb{R}^{n\times\bar{\bar{J}}}$, $E \in \mathbb{R}^{\bar{\bar{J}}\times\bar{J}}$, and $F \in \mathbb{R}^{\bar{\bar{J}}\times I}$, a bias $\tilde{\theta} \in \mathbb{R}^{\bar{\bar{J}}}$, and a (continuous) sigmoid activation function $f : \mathbb{R}^{\bar{\bar{J}}} \to \mathbb{R}^{\bar{\bar{J}}}$.

If we further set

$$r_{t+1} := f(Es'_t + Fu_t - \tilde{\theta}) \quad (\in \mathbb{R}^{\bar{\bar{J}}}) \tag{23}$$

and enlarge the system equations (18) and (22) about this additional component, we achieve the following form

$$\begin{aligned} \begin{pmatrix} s'_{t+1} \\ r_{t+1} \end{pmatrix} &= f\left(\begin{pmatrix} A & 0 \\ E & 0 \end{pmatrix} \begin{pmatrix} s'_t \\ r_t \end{pmatrix} + \begin{pmatrix} B \\ F \end{pmatrix} u_t - \begin{pmatrix} \bar{\theta} \\ \tilde{\theta} \end{pmatrix} \right) \\ \tilde{y}_t &= \begin{pmatrix} 0 & D \end{pmatrix} \begin{pmatrix} s'_t \\ r_t \end{pmatrix} . \end{aligned} \tag{24}$$

Their equivalence to the original equations (18) and (22) is easy to see by a component-wise computation.

Finally out of

$$\begin{aligned} \tilde{J} := \bar{J} + \bar{\bar{J}}, \tilde{s}_t &:= \begin{pmatrix} s'_t \\ r_t \end{pmatrix} \in \mathbb{R}^{\tilde{J}}, \\ \tilde{A} := \begin{pmatrix} A & 0 \\ E & 0 \end{pmatrix} \in \mathbb{R}^{\tilde{J}\times\tilde{J}}, \tilde{B} &:= \begin{pmatrix} B \\ F \end{pmatrix} \in \mathbb{R}^{\tilde{J}\times I}, \\ \tilde{C} := \begin{pmatrix} 0 & D \end{pmatrix} \in \mathbb{R}^{n\times\tilde{J}} \text{ and } \theta &:= \begin{pmatrix} \bar{\theta} \\ \tilde{\theta} \end{pmatrix} \in \mathbb{R}^{\tilde{J}}, \end{aligned}$$

follows

$$\begin{aligned} \tilde{s}_{t+1} &= f(\tilde{A}\tilde{s}_t + \tilde{B}u_t - \theta) \\ \tilde{y}_t &= \tilde{C}\tilde{s}_t \quad . \end{aligned} \tag{25}$$

Equation (25) is apparently an element of the function class $RNN^{I,n}(f)$. Thus the theorem is proven.

q. e. d.

5 Conclusion

In this paper we gave a proof for the universal approximation ability of RNNs in state space model form. After a short introduction into open dynamical systems and RNNs in state space model form we recalled the universal approximation theorem for feedforward neural networks. Based on this result we proofed that RNNs in state space model form are able to approximate any open dynamical system with an arbitrary accuracy.

The proof can be seen as a basis for future work on RNNs in state space model form as well as a justification for their use in many real-world applications. It also underlines the good results we achieved by applying RNNs to various time-series problems.

Nevertheless further research is done on a constant enhancement of RNNs for a more efficient use in different practical questions and problems. In this context it is important to note that for the application of RNNs to real-world problems an adaption of the model to the respective task is advantageous as it improves its quality. Besides that we will continue our work on high-dimensional and dynamical consistent neural networks [4].

References

1. Hornik, K., Stinchcombe, M., White, H.: Multi-layer feedforward networks are universal approximators. Neural Networks **2** (1989) 359–366
2. Cybenko, G.: Approximation by superpositions of a sigmoidal function. In: Mathematics of Control, Signals and Systems. Springer, New York (1989) 303–314
3. Funahashi, K.I.: On the approximate realization of continuous mappings by neural networks. Neural Networks **2** (1989) 183–192
4. Zimmermann, H.G., Grothmann, R., Schaefer, A.M., Tietz, C.: Identification and forecasting of large dynamical systems by dynamical consistent neural networks. In Haykin, S., J. Principe, T.S., McWhirter, J., eds.: New Directions in Statistical Signal Processing: From Systems to Brain. MIT Press (2006)
5. Haykin, S.: Neural Networks: A Comprehensive Foundation. Macmillan, New York (1994)
6. Zimmermann, H.G., Neuneier, R.: Neural network architectures for the modeling of dynamical systems. In Kolen, J.F., Kremer, S., eds.: A Field Guide to Dynamical Recurrent Networks. IEEE Press (2001) 311–350
7. Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning internal representations by error propagation. In Rumelhart, D.E., et al., J.L.M., eds.: Parallel Distributed Processing: Explorations in The Microstructure of Cognition. Volume 1. MIT Press, Cambridge (1986) 318–362
8. Hammer, B.: On the approximation capability of recurrent neural networks. In: International Symposium on Neural Computation. (1998)