Mantas Lukoševičius, Dan Popovici, Herbert Jaeger,Udo Siewert

Time Warping Invariant Echo State Networks

Technical Report No. 2
submitted February 2006, accepted May 2, 2006

School of Engineering and Science

Time Warping Invariant Echo State Networks

Mantas Lukoševičius, Dan Popovici, Herbert Jaeger

International University Bremen
School of Engineering and Science
28759 Bremen
Germany

E-Mail: {m.lukosevicius,d.popovici,h.jaeger}@iu-bremen.de
http://www.iu-bremen.de/

Udo Siewert

Planet GmbH
Residence Park 1-7
D-19065 Raben Steinfeld
Germany

E-Mail: siewert@planet.de
http://www.planet.de/

Summary

Echo State Networks (ESNs) is a recent simple and powerful approach to training recurrent neural networks (RNNs). In this report we present a modification of ESNs - time warping invariant echo state networks (TWIESNs) that can effectively deal with time warping in dynamic pattern recognition. The standard approach to classify time warped input signals is to align them to candidate prototype patterns by a dynamic programming method and use the alignment cost as a classification criterion. In contrast, we feed the original input signal into specifically designed ESNs which intrinsically are invariant to time warping in the input. For this purpose, ESNs with leaky integrator neurons are required, which are here presented for the first time, too. We then explain the TWIESN architecture and demonstrate their functioning on very strongly warped, synthetic data sets.

Contents

1 Introduction

Time warping of input patterns is a common problem when recognizing human generated input or dealing with data artificially transformed into time series. In many of these application areas artificial *recurrent neural networks* (RNNs) play (or could potentially play) a significant role. However, research on neural networks that can effectively deal with time warping does not seem to be very active.

The most widely used technique for dealing with time-warped patterns is called *dynamic time warping* (DTW) [Itakura, 1975] and its modifications. All of them are based on the idea of finding the cheapest (w.r.t. some cost function) mapping between the observed signal and the pattern. The price of the mapping is then taken as the classification criterion. Another very common approach to time-warped recognition is *hidden Markov models* (HMMs) [Rabiner, 1990]. HMMs model the underlying generating process as a Markov process, in which transitions are memoryless, i.e. the probability of a transition to a different state does not depend on how long we have stayed in the previous one. An interesting approach of combining HMMs and neural networks is proposed in [Levin et al., 1992], where neurons that time-warp their input to match it to its weights optimally are introduced.

A simple time warping invariant way of directly applying RNNs for time series classification was presented in [Sun et al., 1993] more than ten years ago. In this report we follow the idea in [Sun et al., 1993] to derive an effective method for dynamic recognition of time-warped patterns.

This report has two main parts, theory and architecture in Section 2, and numerical simulations in Section 3. More specifically, we briefly introduce the main concepts of ESNs together with our notation in Sections 2.1 and 2.2. In Section 2.3 we explain the principle of intrinsic time warping invariance that we use. In Sections 2.4 and 2.5 we introduce ESNs with leaky integrator neurons (LINs) and describe how the ESN learning scheme has to be adapted for this type of neurons. In Section 3.1 we describe the data which is used in our numerical simulations. In Section 3.2 we present empirical investigation of ESNs with LINs and their parameters, and in Section 3.3 we investigate the performance of our approach with different degrees of time warping in data. Finally, in Section 4 we present some insights on possible further refinements of our method.

2 Theory and Architecture

2.1 Echo State Networks

Echo state networks (ESNs) [Jaeger and Haas, 2004] is a recent approach to recurrent neural network supervised training, which overcomes some obstacles in many other approaches to training RNNs, namely implementation complexity of learning algorithms, slow convergence and suboptimal solutions in their training. In

the ESN approach a large (order of 50 to 1000 neurons), randomly connected RNN is used as a "reservoir" of dynamics which can be excited by suitably presented input and/or fed-back output. The connection weights of this reservoir network are not changed by training. In order to compute a desired output dynamics, only the weights of connections from the reservoir to the output units are calculated. A similar idea has recently been independently investigated in a more biologically oriented setting under the name of "liquid state networks" [Maass et al., 2002]. Because there are no cyclic dependencies between the trained readout connections, training an ESN becomes a simple linear regression task, for which numerous batch or adaptive online algorithms are available.

The echo state networks investigated previously were discrete-time sigmoid unit networks with the following state update equation:

$$x(n+1) = f(W_{\text{in}}u(n+1) + Wx(n)), \tag{1}$$

where x is a vector of reservoir neuron activations, n is the discrete time, f is the neuron activation function (usually the tanh sigmoid) applied component-wise, W_{in} is the input weight matrix, u is the input vector, and W is a randomly generated weight matrix of internal reservoir connections. The output equation is

$$y(n+1) = f_{\text{out}}(W_{\text{out}}[x(n+1)|u(n+1)]), \tag{2}$$

where W_{out} is the (learnt) output weight matrix, f_{out} is the output neuron activation function (usually tanh sigmoid or the identity) applied component-wise, and "|" stands for vector concatenation. The standard batch supervised training of ESN proceeds by driving them with the training input sequence $u_{\text{teacher}}(n)$ once, harvesting the internal states, and then computing the output weights W_{out} as the linear regression weights of the teacher output $y_{\text{teacher}}(n)$ on the internal states. Because the learning is essentially a linear regression task, adaptive online learning methods can be obtained by employing standard methods from adaptive linear systems, for instance the RLS algorithm [Jaeger, 2003].

2.2 Echo State Property

The *echo state property* is crucial for making the ESN learning method work. Intuitively, a RNN which is driven by an external signal $u(n)$ has the echo state property if the activations $x(n)$ of the RNN neurons are systematic variations of the driver signal $u(n)$. More formally, this means that for each internal unit x_i there exists an "echo function" e_i, such that, if the network has been run for an indefinitely long time in the past, the current state can be written as $x_i(n) = e_i(u(n), u(n-1), u(n-2), \cdots)$. For discrete-time ESNs there are several nontrivial alternative definitions of this condition and algebraic characterizations of which network weight matrices W lead to networks having the echo state property [Jaeger, 2001]. For practical purposes it suffices to fix the spectral radius $\rho(W)$ of W to a value below unity to ensure the echo state property. It is also

important that the dynamics of the reservoir neurons be richly varied. This is ensured by a sparse interconnectivity (of 1-20%) within the reservoir. The condition lets the reservoir decompose into many loosely coupled subsystems, establishing a richly structured reservoir of excitable dynamics.

2.3 Time Warping Invariance

Intuitively time warping can be understood as variations in the "speed" of a process. For discrete-time signals obtained by sampling from a continuous time series it can alternatively be cast as variations in the sampling rate. By definition two signals $\alpha(t)$ and $\beta(t)$ are connected by an approximate continuous *time warping* (τ_1, τ_2), if τ_1, τ_2 are strictly increasing functions on $[0, T]$, and $\alpha(\tau_1(t)) \cong \beta(\tau_2(t))$ for $0 \leq t \leq T$. We can choose one signal, say $\alpha(t)$, as a reference and all signals that are connected with it by some time warping (e.g. $\beta(t)$) call *(time-)warped* versions of $\alpha(t)$. We will also refer to a time warping (τ_1, τ_2) as a single *time warping (function)* $\tau(t) = \tau_2(\tau_1^{-1}(t))$ which connects the two time series by $\beta(t) = \alpha(\tau(t))$.

A common problem is recognition (which is detection plus classification) of time-warped patterns in a signal. In this report we start out from an idea of time warping invariant neural networks originally proposed in [Sun et al., 1993]. This approach (in contrast to most others) does not look for a time warping that could map an observed signal to a target pattern, but treats all signals in a time warping invariant fashion. Time warping invariance is achieved by normalizing time dependence of the state variables with respect to the length of trajectory of the input signal in its phase space. In other words, the input signal is considered in a "pseudo-time" domain, where "time span" between two subsequent pseudo time steps is proportional to the metric distance in the input signal between these time steps. As a consequence, input signals will be changing with a constant metric rate in this "pseudo-time" domain. In continuous time, for a k-dimensional input signal $u(t)$, $u : \mathbb{R}^+ \to \mathbb{R}^k$ we can define such a time warping $\tau'_u(t)$, $\tau'_u : \mathbb{R}^+ \to \mathbb{R}^+$ by

$$d\tau'_u(t)/dt = b \cdot \|du(t)/dt\|\,, \tag{3}$$

where b is a constant factor. Note, that the time warping function τ'_u depends on the signal u which it is warping. Then the signal warped by τ'_u (i.e. in the "pseudo-time" domain) becomes $u(\tau'_u(t))$, and as a consequence $\|du(\tau'_u(t))/dt\| = 1/b$, i.e. the k-dimensional input vector $u(\tau'_u(t))$ changes with a constant metric rate equal to $1/b$ in this domain. Furthermore, if two signals $u_1(t)$ and $u_2(t)$ are connected by a time warping τ, then time-warping them with τ'_{u_1} and τ'_{u_2} respectively results in $u_1(\tau'_{u_1}(t)) = u_2(\tau'_{u_2}(t))$, which is what we mean by *time warping invariance* (see Figure 1 for the graphical interpretation of the $k = 1$ case).

A continuous-time processing device could be made time warping invariant, if for any given input $u(t)$ it could vary its processing speed (i.e. its internal "pseudo-time") according to $\tau'_u(t)$ by changing the time constant in the equations describing its dynamics. This is an alternative to time-warping the input signal $u(t)$ itself,

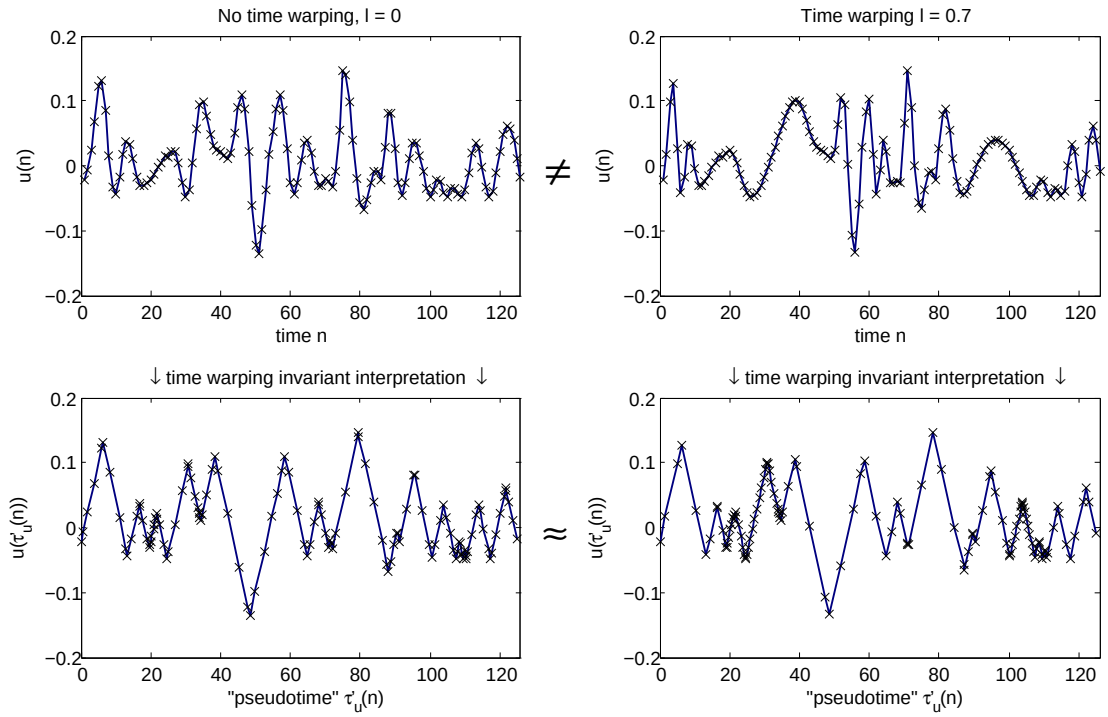

Figure 1: A time warping invariant interpretation of two one-dimensional signals connected by a time warping. We can see that in the "pseudotime" domain $\tau'_u(n)$ (3) the shape of the signal is basically time warping independent, only the density of the data points differs. We can also observe that for the specific case of one-dimensional input the interpretation causes a big loss of the original temporal information – the signal $u(\tau'_u(n))$ can be fully described by the sequence of values at local minimums and maximums of $u(n)$. The notion of time warping level l is taken from (9) here.

which may be difficult (e.q. with discrete time signals) or even impossible (e.g. in real time processing).

2.4 ESN with Leaky Integrator Neurons

As we have just argued, in order to process a time-warped signal in the "pseudotime" domain by a continuous-time processing device, the latter must be described by a differential equation featuring a time constant. In order to employ ESNs, we therefore need continuous-time reservoir dynamics. We have chosen *leaky integrator* neuron reservoirs since they are the most simple and most commonly used continuous-time neurons. A preliminary discussion of such leaky integrator ESNs was provided in [Jaeger, 2001], but we need a better understanding of such ESNs for our purposes, which we presently develop.

A leaky integrator neuron (LIN) is a biologically inspired model of a neu-

ron, which accumulates (integrates) its inputs, but also exponentially loses (leaks) accumulated excitation over time. In contrast to Eq. (1), the continuous-time dynamics $x(t)$ of leaky integrator reservoir are given by the differential equation

$$\dot{x}(t) = \frac{1}{c}(-ax(t) + f[W_{\text{in}}u(t) + Wx(t)]), \tag{4}$$

where the positive quantity c is the time constant of this equation (governing the speed of its dynamics) and a is the decay (or leakage) rate. For simulations of these dynamics on digital computers Eq. (4) must be discretized. The simplest discretization is the Euler (linear) interpolation, which turns Eq. (4) into

$$x(n+1) = (1 - a\Delta t)x(n) + \Delta t f(W_{\text{in}}u(n+1) + Wx(n)), \tag{5}$$

where Δt is a time gap between two consecutive time steps divided by the time constant c. The output equation of our ESN remains as Eq. (2). In this model we can implement time warping by varying Δt over time steps $\Delta t = \Delta t'(n+1)$, where $\Delta t'(n+1)$ is a "pseudo-time" gap between time steps n and $n+1$. Using a discrete time version of Eq. (3) we get

$$\Delta t'(n+1) = \tau'_u(n+1) - \tau'_u(n) = b \cdot \|u(n+1) - u(n)\| . \tag{6}$$

Note, that our "pseudo-time" is dimensionless, so we can choose a time constant $c = 1$ or alternatively assume that it is incorporated in b. Substituting Δt in Eq. (5) with $\Delta t'(n+1)$ from Eq. (6), we obtain the state update equation of a *time warp invariant echo state network* (TWIESN):

$$x(n+1) = x(n) - b\,\|u(n+1) - u(n)\|\,(ax(n) + f[W_{in}u(n+1) + Wx(n)]). \tag{7}$$

2.5 Parameter Constraints of ESNs with LINs

Going from classical ESN to leaky-integrator ESN, we have to reformulate the conditions for the echo state property into

$$\rho((1 - a\Delta t)I + \Delta t W) \leq 1, \tag{8}$$

where $\rho()$ denotes spectral radius, I is identity matrix and Δt denotes average time gap between two consecutive time steps [Jaeger, 2002]. Other natural constraints are imposed by the definition of leakage: $a\Delta t > 0$, otherwise it would not be a leakage, and $a\Delta t \leq 1$, as a neuron can not leak more excitation than it has. At this point TWIESNs effectively leave us with three free parameters, that should be optimized satisfying the constraints: (i) the spectral radius of the reservoir weight matrix $\rho(W)$, (ii) the decay rate a, and (iii) the (average) time gap between two time steps $\Delta t(n)$. There is no analytical method known for finding a combination of these values that minimizes the training error. In Section 3.2 we will however describe insights that help to optimize these global control parameters heuristically. Note, that the classical ESN is a special case of leaky-integrator ESN, where $a = 1$ and $\Delta t = 1$ [Eq. (5)], therefore performance of optimized leaky-integrator ESN must be better or equal to the classical ESN.

3 Numerical Simulations

3.1 Data Used for Numerical Simulations

We have performed extensive numerical simulations using synthetic data. We started out from data that combined a red noise (a $[-0.5, 0.5]$ uniformly distributed white noise with filtered-out 60% of its higher frequencies) background signal $g(t)$ with smoothly embedded random short target sequences $p(t)$ with a similar frequency makeup. Smooth continuous-time signals of this kind were produced, and then time-warped discrete-time samples were drawn (the above mentioned frequency makeup corresponds to the discrete-time signals with no time warping). We did a recognition of only one pattern (i.e. the ESN had 1-dimensional output), as recognition of multiple patterns would in essence be done independently.

More specifically, first a short (length T_p) target sequence $p(t)$, $p : [0, T_p] \to \mathbb{R}^k$ was generated in the same (above described) way as $g(t)$, $g : \mathbb{R}^+ \to \mathbb{R}^k$ (all k dimensions were generated independently). Then, in order to smoothly embed $p(t)$ into $g(t)$, a windowing signal $w(t)$ was created, where $t \in [0, T_p]$, $w(0) = w(T_p) = 0$, and $w(t)$ gently rises to 1 after $t = 0$ and smoothly falls again to zero level at $t = T_p$. $w(t)$ was made by filtering a trapezoidal window signal with the same low-pass filter which was used to produce $g(t)$ and $p(t)$, so that $w(t)$ would not introduce any new (high) frequencies. Then the input signal $u(t)$ was produced by smoothly embedding $p(t)$ into $g(t)$ at random positions t_i, where $i \in \mathbb{N}$, and $(t_{i+1} - t_i) \in (T_p, 3T_p)$ is a uniformly distributed random variable with a mean value $2T_p$. At each t_i the embedding of $p(t)$ was $u(t_i + t) = (1 - w(t))g(t_i + t) + w(t)p(t)$, where $t \in [0, T_p]$. The (1-dimensional) desired output signal $y_{\text{teacher}}(t)$ was constructed by placing Gaussian bumps centered at the time points $t_i + T_p$ on a background zero signal. The height of the bumps is 1 and the width roughly corresponds to the average width of the main bump of the autocorrelation of $p(t)$.

The above described continuous time signals were modeled in Matlab by a cubic interpolation of the signals having a six times higher discretization rate, and in some sense stood for the underlying generating process $u(t)$, where $t \in \mathbb{R}^+$. Discrete time time-warped observations $u(n)$ of the process $u(t)$ were drawn as $u(n) = u(\tau(n))$, where $\tau : \mathbb{N} \to \mathbb{R}^+$ fulfilled both time warping and discretization functions. Both $u(t)$ and corresponding $y_{\text{teacher}}(t)$ were discretized/time-warped together. More specifically, we used

$$\tau(n) = (n + 10 \cdot l \sin(0.1n)), \tag{9}$$

where $l \in [0, 1]$ is the level (degree) of time warping: $\tau(n)$ is a straight line (no time warping) when $l = 0$, and is a nondecreasing function as long as $l \leq 1$. In the obtained signals $u(n)$ a time interval T_p on average corresponded to 20 time steps and $u(n)$ had (as mentioned before) on average 40% of its lower frequencies present. The period of the time warping $\tau(n)$ is 20π, which stands in no rational relationship with T_p and $(t_{i+1} - t_i)$ (the latter being random anyway). Discretization, smooth embedding, and time-warping meant that different instances of $p(t)$

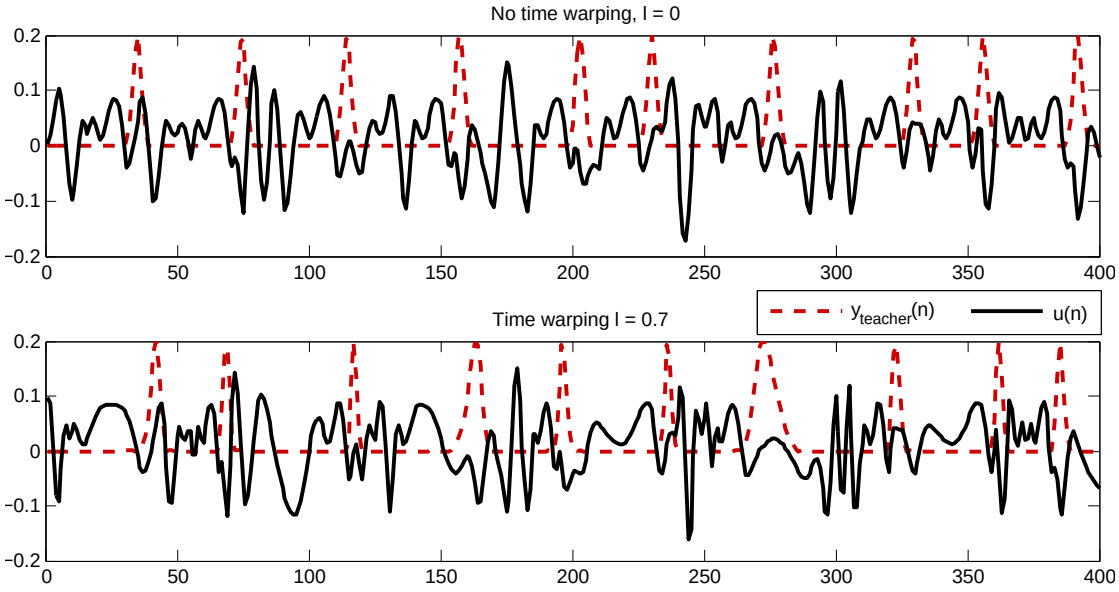

Figure 2: A fragment of a one-dimensional input and corresponding teacher signal without and with time warping.

may differ considerably in $u(n)$ (see Figure 2). This, together with $p(t)$ having similar statistics and spectrum as $u(t)$, make this a hard recognition task.

All the below-reported simulations used 1000 data points (iterations) to get rid of possible initial transients of ESN, 2000 iterations for training, and 500 for testing. All the ESNs had reservoirs of 50 units, with 20% interconnectivity.

3.2 Optimizing Leaky-integrator ESNs

As pointed out in Section 2.5, leaky-integrator ESNs leave us with three free parameters (i-iii) for which an optimal analytical solution is not known. In this section we report an empirical investigation aimed to get insight into conditions for optimizing these three variables. For this we used data with no time warping ($l = 0$). The network was trained with all combinations of the three parameters (i-iii), varying them within reasonable limits: (i) $\rho(W) \in [0.1, 1]$, (ii) $a \in [0, 2.9]$, and (iii) $\Delta t \in [0.1, 3]$, with step size 0.1. The same randomly generated data and internal weights W were used in all the trainings (W was scaled accordingly to change $\rho(W)$).

Some essential results of the investigation are presented in Figure 3. Observed self induced oscillations correspond to the settings, where $\rho((1-a\Delta t)I+\Delta tW) > 1$. We can observe, that the normalized mean square error (NMSE, that is, mean square error divided by data variance) is generally lowest alongside the hyperbola $a\Delta t = \text{const}$. This indicates, that a certain decay is optimal for different settings of Δt. It has to be expected that this rate depends on the dominating frequencies

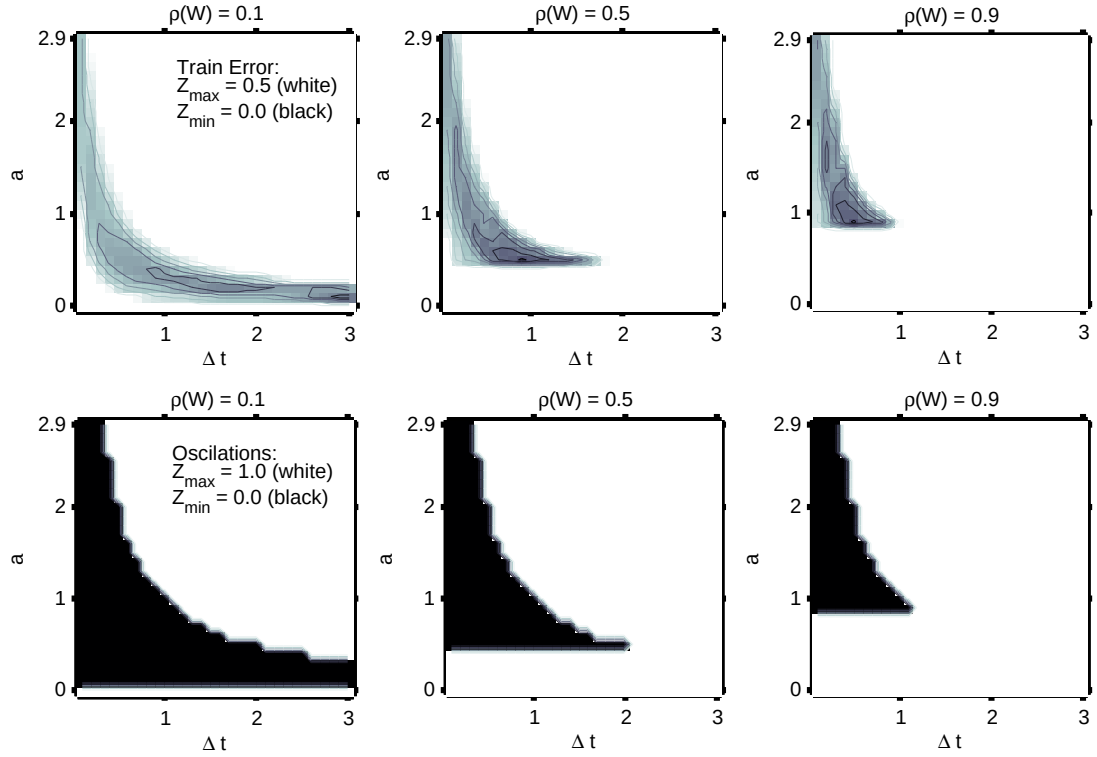

Figure 3: Dependence of mean square training error and high-frequency, self-induced oscillations in internal states of leaky-integrator ESN on leakage rate a (y-axis) and time constant Δt (x-axis) for $\rho(W) = 0.1, 0.5$ and 0.9 respectively.

of the input signals; faster signals presumably would favor bigger decay rates. One possible heuristic for a rapid model optimization could be setting a relatively small $\rho(W) \in [0.2, 0.3]$ and look for a good setting of a and Δt on a diagonal $a = C\Delta t$, where C is a constant of size about 0.3. If an optimal value has been found, compute $r = a_{\text{opt}}\Delta t_{\text{opt}}$ and further refine the model performance (if needed) by exploring parameters a and Δt "sideways" along $a\Delta t = r$.

3.3 Results

To evaluate the performance of TWIESNs, simulations were done with varying levels of time warping l in Eq. (9), and different input dimensions k. Performance of leaky-integrator ESNs was compared to the performance of TWIESNs. Both ESNs were run with parameters $a = 0.3$, $\langle \Delta t \rangle = 1.2$, and $\rho(W) = 0.3$ optimized using the heuristic in Section 3.2. For TWIESNs the parameter b in Eq. (6) was adjusted such, that $\langle \Delta t'(n) \rangle = 1.2$, i.e. $b = 1.2/\langle \|\Delta u(n)\| \rangle$, where $\langle \|\Delta u(n)\| \rangle$ is the average $\|u(n) - u(n-1)\|$ computed over training data. The range of $\Delta t'(n)$ during the runs was bounded by a hard saturation to impose constraints described in Section 2.5: $\Delta t'(n) = \min(b \|\Delta u(n)\|, 1/a)$. While varying the level of time

warping l in the simulation, the underlying continuous-time signals were kept the same. The same data and randomly generated reservoir connections W were used for all modifications of ESNs. Results of the simulation averaged over 100 runs are presented in Figure 4.

A criterion of the actual pattern recognition was the output $y(n)$ level exceeding a certain threshold h. In this simple setup each continuous interval of n for which $y(n) > h$ corresponds to one recognized instance of the pattern. The combined quality of recognition q was calculated for each output signal, by adding the number of correctly recognized patterns, the number of patterns what ESN failed to recognize and the number of "false alarms", and dividing by the number of correctly recognized patterns. A pattern is considered to be recognized correctly if the intervals of n where $y(n) > h$ and $y_{\text{teacher}}(n) > h$ overlap (one interval of $y(n)$ can only be matched with one interval of $y_{\text{teacher}}(n)$). The value of h was optimized by maximizing q over the training data.

We can see in Figure 4, that TWIESN performance remains almost constant while the level of time warping l increases. To better understand the nature of the observed slight decrease in the performance, we also constructed a version of a leaky-integrator ESN, which "unwarped" the signals artificially: $\Delta t(n) = \min(b[\tau(n) - \tau(n-1)], 1/a)$, using the (in real life unavailable) knowledge of the time warping $\tau(n)$ used for producing $u(n)$. The observed identical decrease of performance (see Figure 4) of this method implies, that this decrease is due to information loss induced by producing $u(n)$ and limited accuracy of our neuron model [Eq. (5)] (and thus constraints), but not due to our estimation of $\Delta t'(n)$.

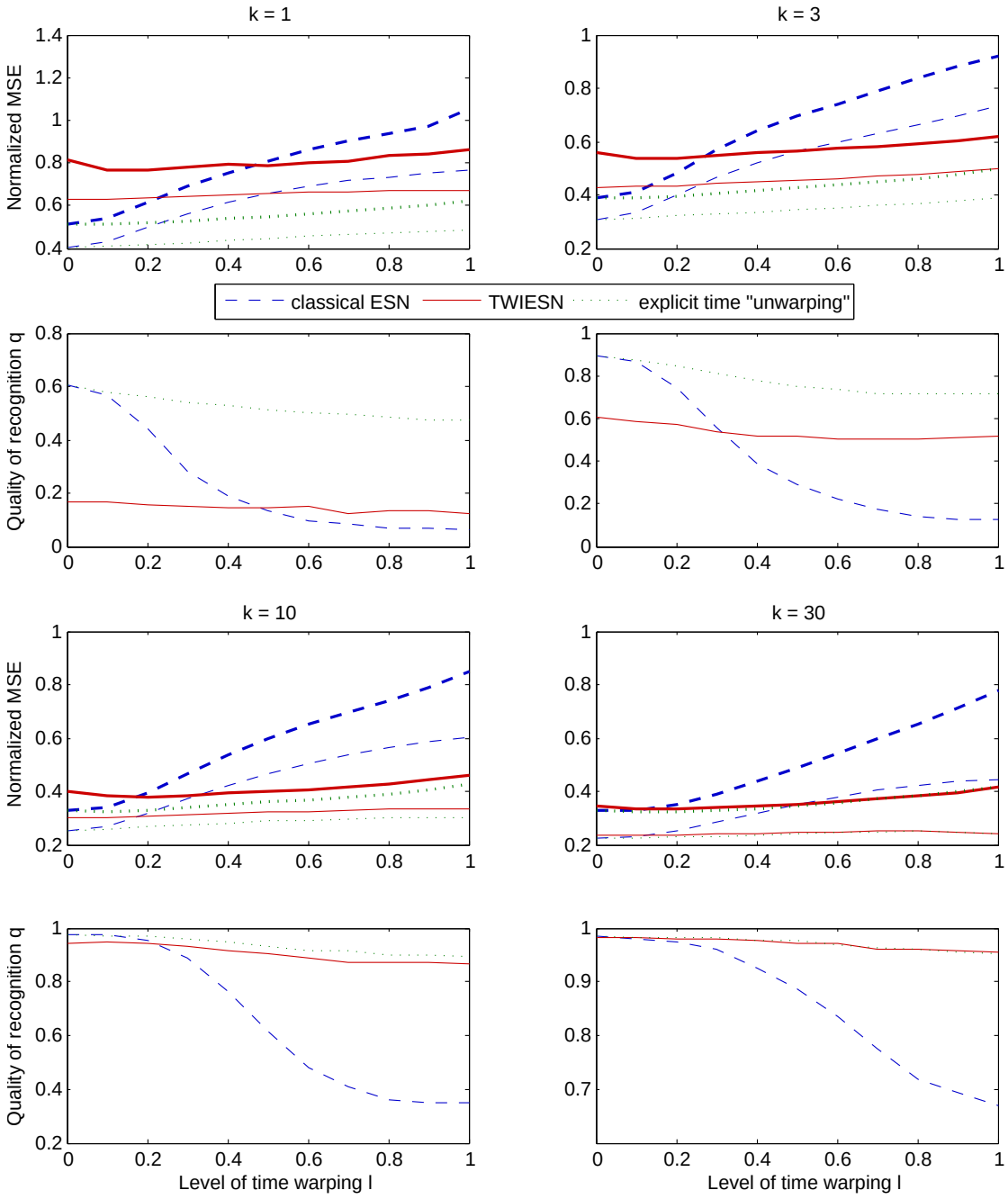

Figure 4: Performance of the classical ESN with LIN (blue dashed line), artificial time "unwarping" ESN (green dotted line) and TWIESN (red solid line), for input dimensions $k = 1, 3, 10$ and 30 respectively, and different levels of time warping l (y-axis). Thin line in normalized mean square error (MSE) plots shows training performance and bold line – testing performance. The plots below the ones with MSEs indicate corresponding comparison of combined recognition qualities q.

We can also observe that the more dimensions k the input signal $u(n)$ has, the lesser level of time warping l is needed to benefit from TWIESNs over simple ESNs (in the case of $k = 30$, performances of TWIESN and artificially "unwarping" ESN are almost identical). It is intuitively clear, that the bigger the number of independent input dimensions we have, the less important the role of the actual time axis is, i.e. the loss of temporal information which is intrinsic to TWIESNs becomes less important. In other words, the temporal information can in some sense be deduced from the dynamics of many independent input variables.

4 Discussion

In this report we presented an investigation of ESNs with leaky-integrator neurons, which is a generalization of the common ESN and thus with good choice of parameters cannot be outperformed by it. Based on the leaky-integrator ESN we proposed TWIESNs – an RNN architecture capable of dynamically recognizing temporal patterns in (strongly) time-warped signals. In contrast to dynamic time warping, this method does not require a "correct" reference pattern, to which the signals are compared.

Further improvements of the method could be done by choosing a more accurate than Euler's approximation of the leaky-integrated neuron dynamics. An alternative way of improving precision could be interpolating input data with additional intermediate points, when input changes rapidly. The work described in this report was carried out as the first step towards applying TWIESNs for handwriting recognition.

Acknowledgments

The work reported here was supported by student contract grants for ML and DP from Planet GmbH, Raben Steinfeld, Germany. The authors would also like to thank five (!) anonymous reviewers of the NIPS 2005 conference, who helped to improve this text by providing valuable feedback.

References

[Itakura, 1975] Itakura, F. (1975). Minimum prediction residual principle applied to speech recognition. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 23(1):67–72.

[Jaeger, 2001] Jaeger, H. (2001). The "echo state" approach to analysing and training recurrent neural networks. Technical Report GMD Report 148, German National Research Center for Information Technology.

[Jaeger, 2002] Jaeger, H. (2002). Tutorial on training recurrent neural networks, covering BPPT, RTRL, EKF and the echo state network approach. GMD Report 159, Fraunhofer Institute AIS, http://www.faculty.iu-bremen.de/hjaeger/pubs/ESNTutorial.pdf.

[Jaeger, 2003] Jaeger, H. (2003). Adaptive nonlinear system identification with echo state networks. In S. Becker, S. T. and Obermayer, K., editors, *Advances in Neural Information Processing Systems 15, [NIPS Conference]*, pages 593–600. MIT Press, Cambridge, MA.

[Jaeger and Haas, 2004] Jaeger, H. and Haas, H. (2004). Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication. *Science*, pages 78–80.

[Levin et al., 1992] Levin, E., Pieraccini, R., and Bocchieri, E. (1992). Time-warping network: A hybrid framework for speech recognition. In *Advances in Neural Information Processing Systems 4, [NIPS Conference]*, pages 151–158.

[Maass et al., 2002] Maass, W., Natschläger, T., and Markram, H. (2002). Real-time computing without stable states: a new framework for neural computation based on perturbations. *Neural Comput.*, 14(11):2531–2560.

[Rabiner, 1990] Rabiner, L. R. (1990). A tutorial on hidden markov models and selected applications in speech recognition. pages 267–296.

[Sun et al., 1993] Sun, G.-Z., Chen, H.-H., and Lee, Y.-C. (1993). Time warping invariant neural networks. In *Advances in Neural Information Processing Systems 5, [NIPS Conference]*, pages 180–187, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.