Analysis of Complex Dynamical Systems with Neural Networks

Open (small) and Closed (large) Recurrent Networks

Dr. Hans Georg Zimmermann

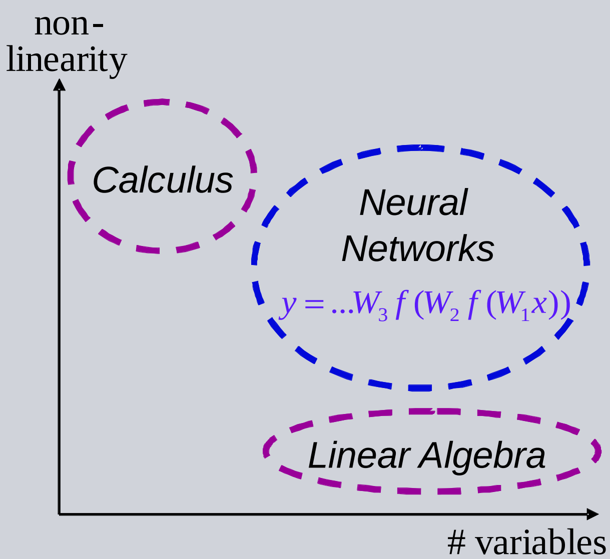
Senior Principal Research Scientist

Siemens AG, Corporate Technology

Email : Hans_Georg.Zimmermann@siemens.com

Mathematical Neural Networks

Complex Systems



Existence Theorem:
(Hornik, Stinchcombe, White 1989)

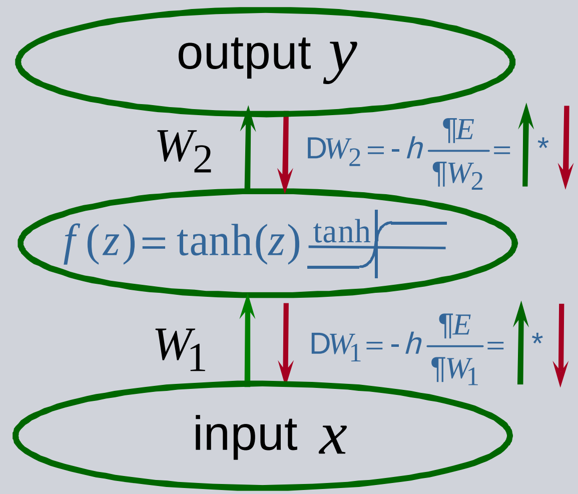
3-layer neural networks $y = W_2 f(W_1 x)$
can approximate any continuous
function on a compact domain.

Nonlinear Regression

Based on data identify an input-output relation

$$y = W_2 f(W_1 x)$$

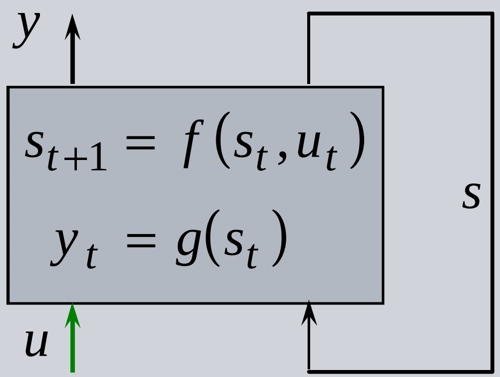
$$E = \sum_{t=1}^{T} \left(y_t - y_t^d\right)^2 \rightarrow \min_{W_1, W_2}$$



Neural networks imply a **Correspondence** of
Equations, ***Architectures***, ***Local Algorithms***.

SIEMENS

Modeling of Open Dynamical Systems with Recurrent Neural Networks (RNN)



$$s_{t+1} = \tanh(As_t + Bu_t)$$ state transition

$$y_t = Cs_t$$ output equation

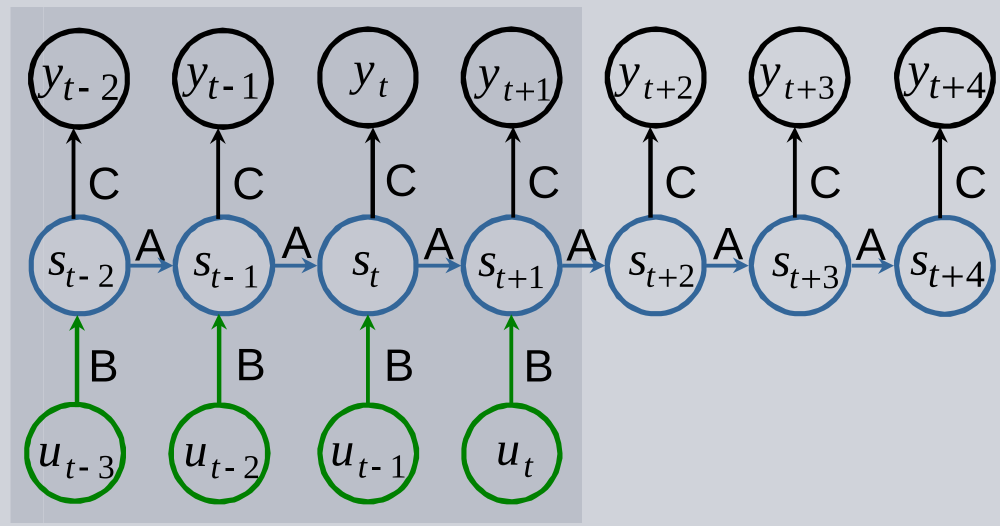
$$\sum_{t=1}^{T} (y_t - y_t^d)^2 \to \min_{A,B,C}$$ identification

Finite unfolding in time transforms time into a spatial architecture. We assume, that future inputs are const.

The analysis of open systems by RNNs allows a decomposition of autonomous & external driven subsystems.

Assume a constant environment in the future and $u_{t>t} = x_t - x_{t-1} = 0$

Long-term predictability depends on a strong autonomous subsystem.



Modeling Dynamical Systems with Error Correction Neural Networks (ECNN)

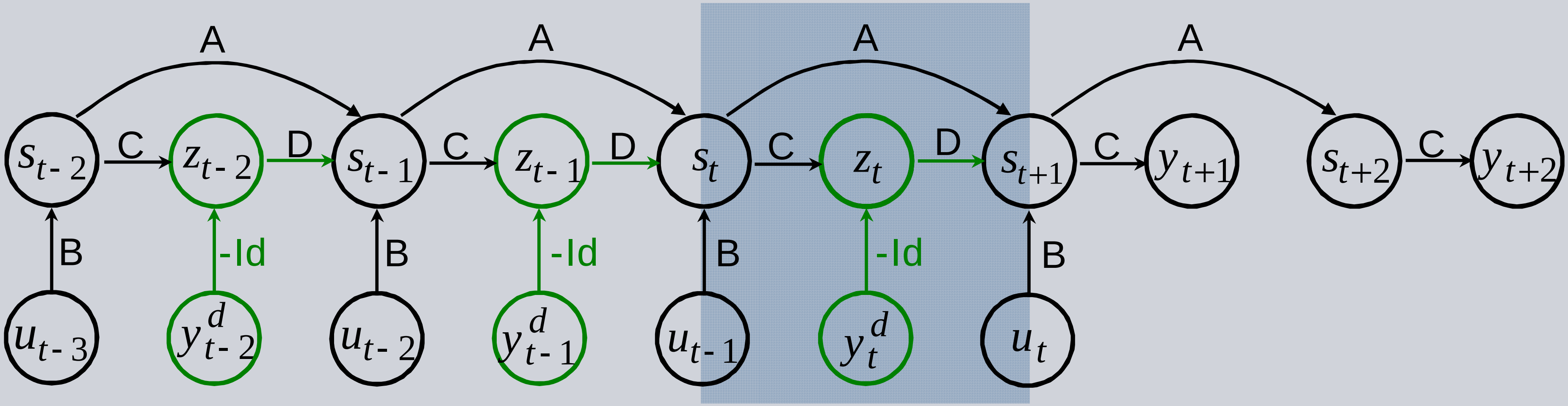
An error correction system considers the forecast error in present time as a reaction on unknown external information.

In order to correct the forecasting this error is used as an additional input, which substitutes the unknown external information.

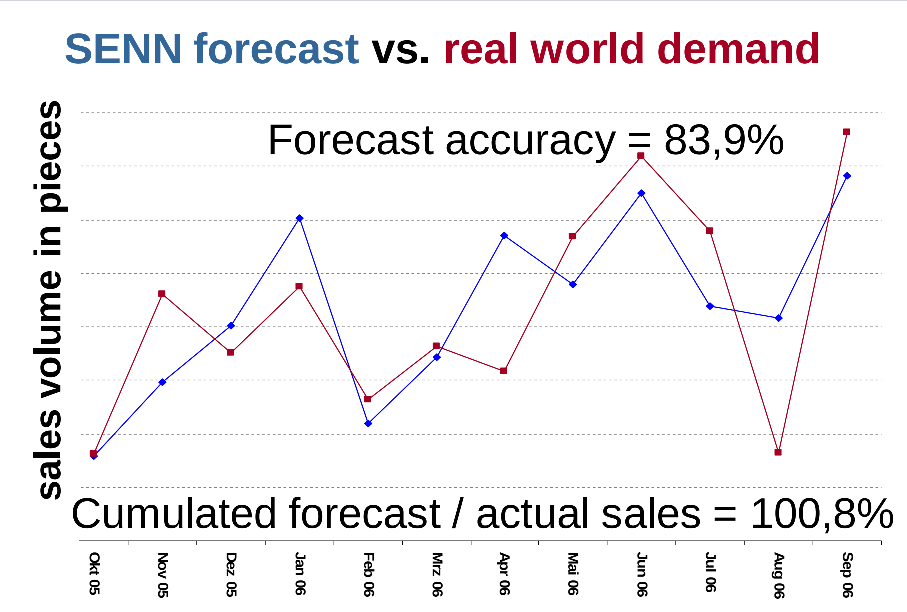
$$s_{t+1} = f\left(s_t, u_t, \left(y_t - y_t^d\right)\right)$$

$$y_t = g(s_t)$$

$$\sum_{t=1}^{T} \left(y_t - y_t^d\right)^2 \rightarrow \min_{f,g}$$



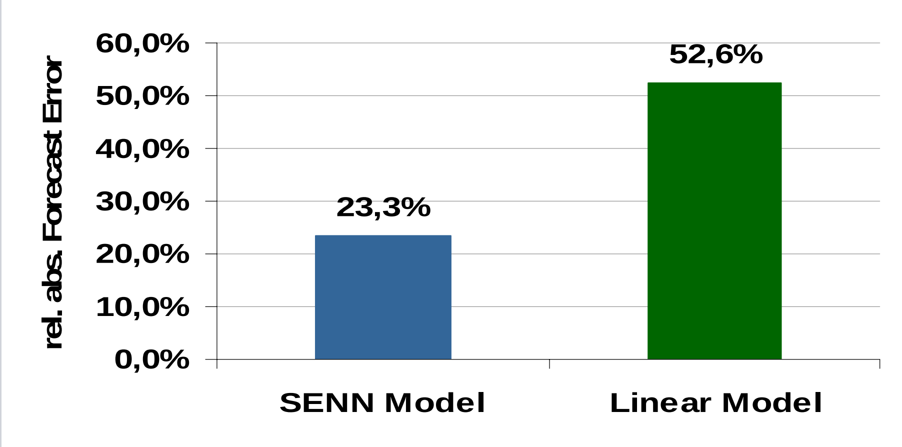
Recurrent Neural Networks in Demand Forecasting



Expert Group	Forecast Error SENN Model	Linear Model
1	8,6%	10,4%
2	42,6%	71,9%
3	13,8%	36,7%
4	10,4%	24,1%
5	27,7%	27,3%
6	15,0%	21,7%
7	24,0%	27,4%
8	16,9%	121,5%
9	26,9%	39,5%
10	26,1%	22,3%
11	23,8%	35,8%
12	35,7%	60,4%
13	51,3%	237,9%
14	12,2%	12,3%
15	19,5%	29,2%
16	18,1%	63,0%
Average	23,3%	52,6%
Std. Deviation	11,5%	54,9%

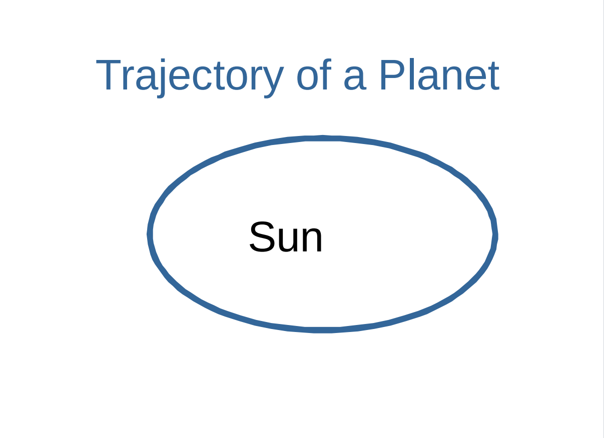
Neural networks clearly outperform linear regression models:

The forecast accuracy of the neural networks is higher and more stable.



Dynamical Systems on Manifolds

To speak about dynamical systems fulfilling side constraints in form of equations is equivalent to dynamical systems on manifolds (conservation laws, e.g. energy or impulse).



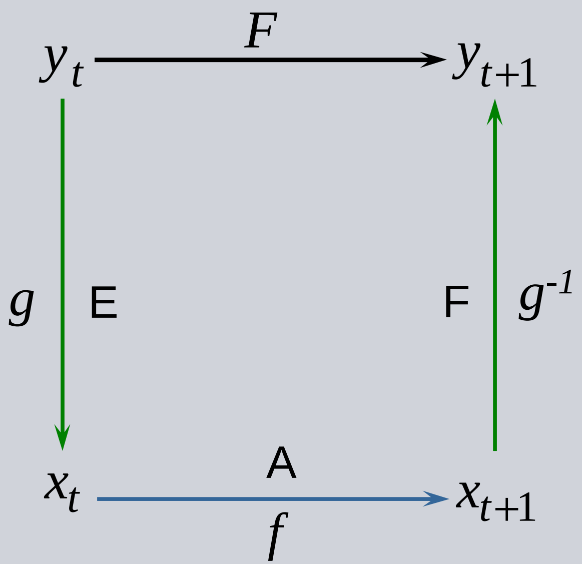
If we know that a dynamics stays on a sub-manifold we can describe it in a lower dim. coordinate system (e.g. trajectory of a fly in a 3dim space or on a 2dim sphere).

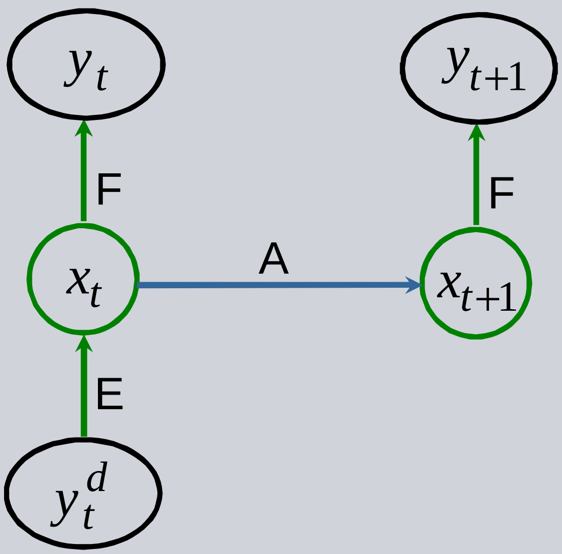
What can we do, if we have only observations of the dynamics, but no a priori knowledge of the underlying manifold ?

The Learning of Dynamical Systems on Manifolds

Assume, that we measure a dynamical system $y_{t+1} = F(y_t)$ in a high dimensional space, but the actual dynamics stays on a low dimensional sub manifold.

We try to find a coordinate transformation $x_t = g(y_t)$ such that the high dimensional flow can be written in the form of a superposition: $y_{t+1} = g^{-1} \circ f \circ g(y_t)$

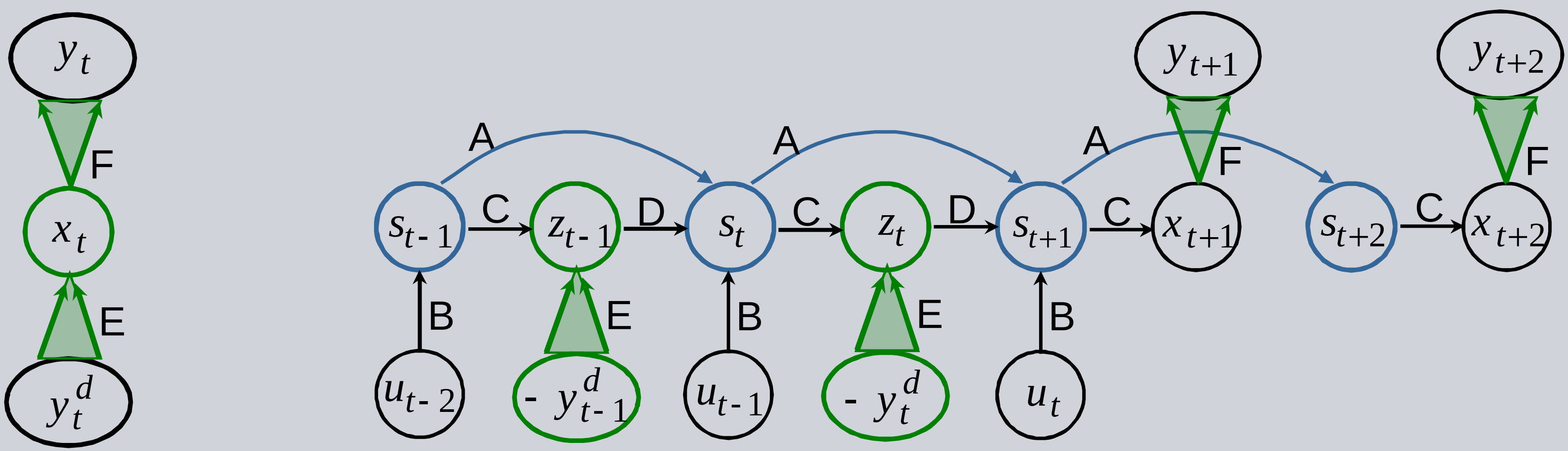




The Learning of ECNNs on Manifolds

$$\begin{tikzcd}
y_t \arrow[r, "F"] \arrow[d, "g"', "\mathrm{E}"] & y_{t+1} \\
x_t \arrow[r, "\mathrm{ECNN}", "f"'] & x_{t+1} \arrow[u, "\mathrm{F}", "g^{-1}"']
\end{tikzcd}$$

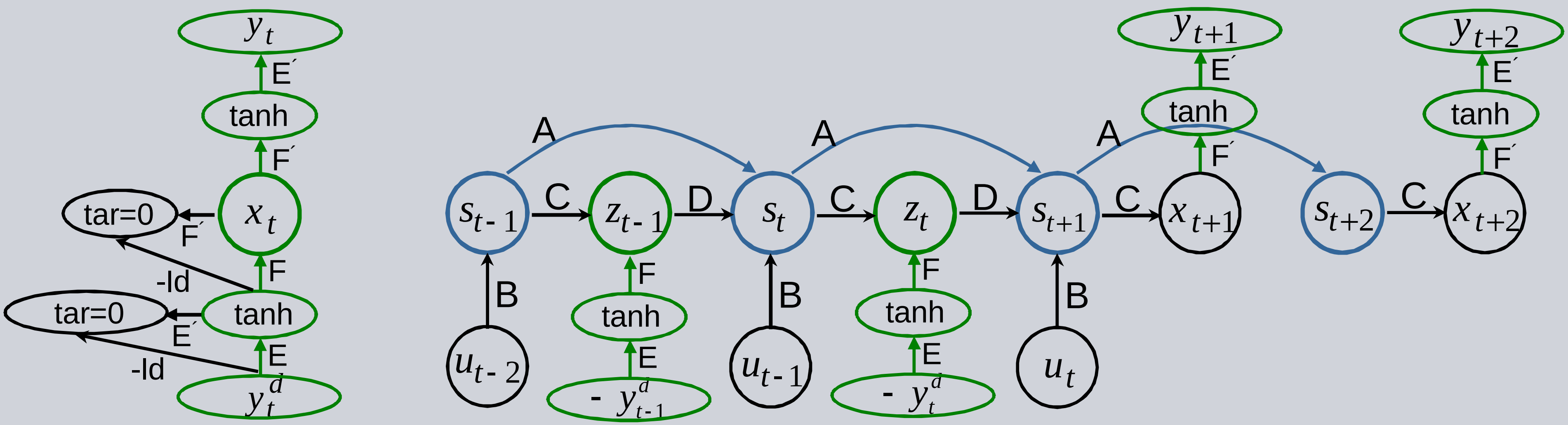
- The bottleneck autoencoder estimates the static coordinate transformations.
- The Error Correction Neural Network solves the transformed temporal problem.
- The sub-networks are implicitly coupled by shared weights.
- The model combines the identification of a manifold and a dynamics on this manifold.



Nonlinear Coordinate Transformations & Dynamical Systems on Manifolds

$$
\begin{tikzcd}
y_t \arrow[r, "F"] \arrow[d, "g"'] & y_{t+1} \\
x_t \arrow[r, "ECNN", "f"'] & x_{t+1} \arrow[u, "g^{-1}"']
\end{tikzcd}
$$

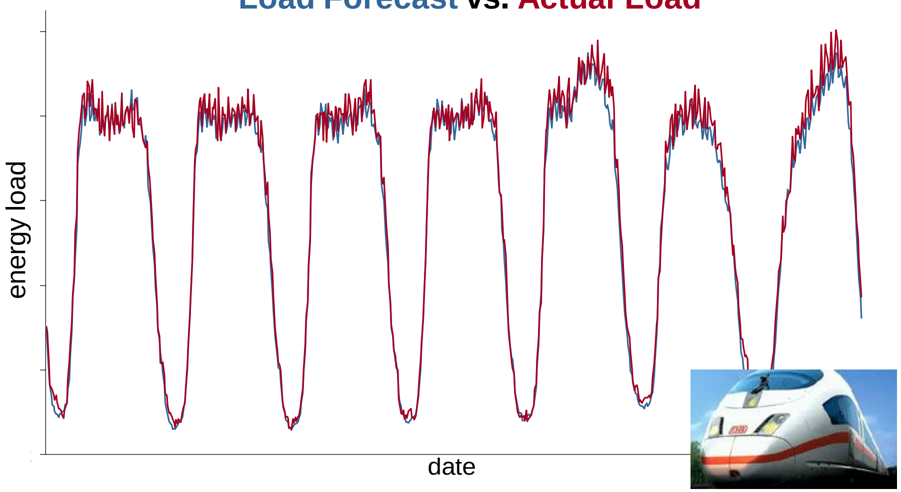
A nonlinear coordinate transformation from a high dimensional state space on to a manifold can be designed in form of a deep stacked auto-encoder. This might be done using pairs (E, E´) and (F, F´) or (E, E^T) and (F, F^T) to lower the parameterization of the coordinate transformations.



SIEMENS

Effective Load Forecasting with Recurrent Neural Networks

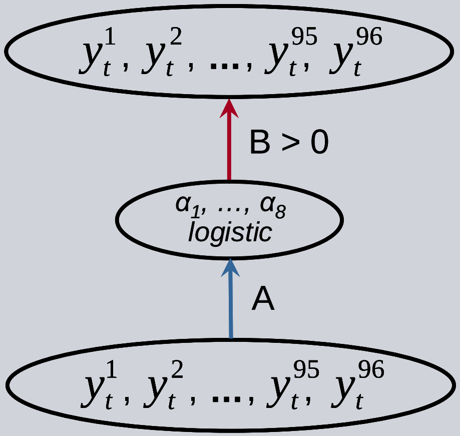
Load Forecasting for DB Energy



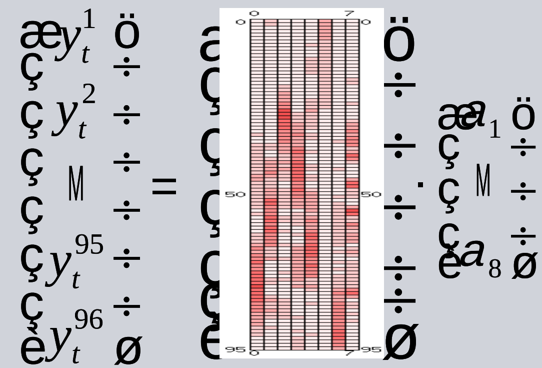
Accuracy of load forecast is 97,95% (Benchmark 95%)

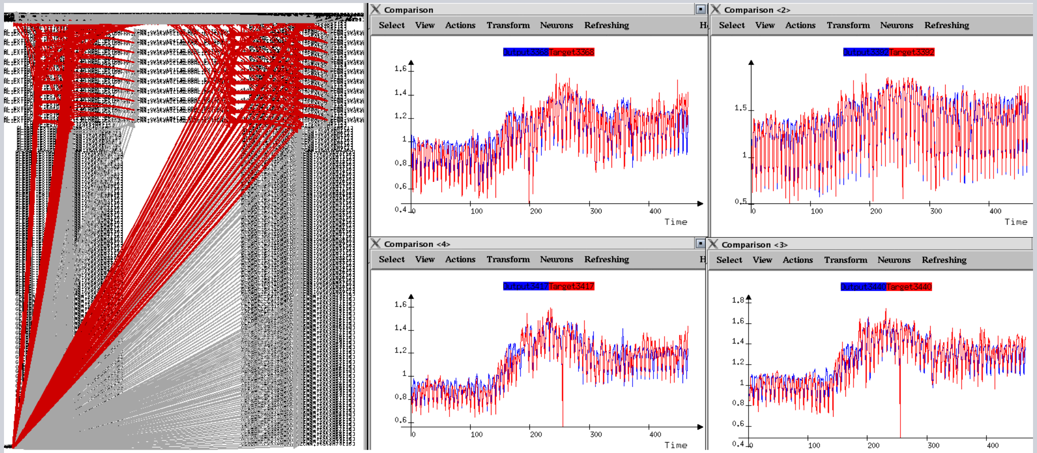
- **Task:** Predict the upcoming energy demand on a 15 min. time grid up to 5 days ahead.
- **Difficulty:** Incorporate the impact of external influences on the energy demand.

Compression of Electrical Load Curves

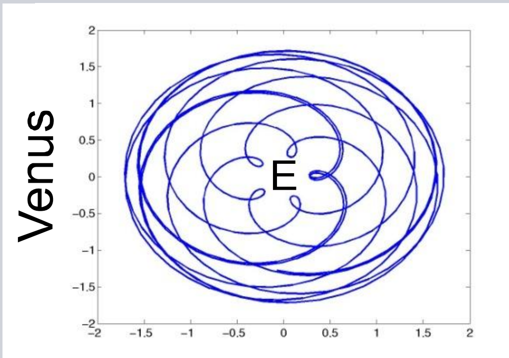


Daily load is represented by a superposition of intraday sub-load curves

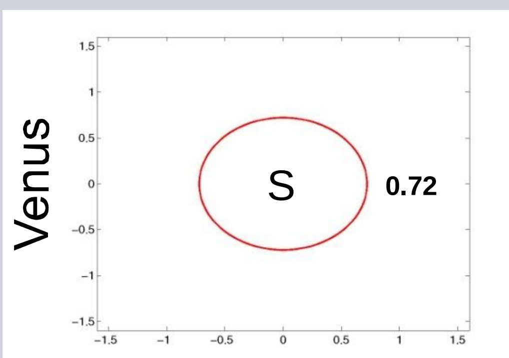




From Ptolemaeus to Copernicus: Optimal State Space Reconstruction



$$\begin{tikzcd} y_t \arrow[r, "F"] \arrow[d, "g"'] & y_{t+1} \\ x_t \arrow[r, "f"] & x_{t+1} \arrow[u, "h"'] \end{tikzcd}$$



The observed dynamics y_t may be measured in a non-optimal coordinate system. This can result in a very inefficient description

$$y_{t+1} = F\,(y_t).$$

In economics the measurement of a dynamics is tricky, because the units themselves change over time.

Find coordinate transformations g, h, such that the transformed trajectory x_t is easier to forecast.

Technically, the trajectory x_t should have a lower curvature than the original one.

Smoothing by Unfolding in Space and Time

Idea: Separate the system identification $y_{t+1} = F(y_t)$ into $y_{t+1} = h \circ f \circ g\,(y_t)$;

$g()$, $h()$ are transformations at a given time; $f()$ is the dynamics over time.

$$
\begin{tikzcd}
y_t \arrow[r, "F"] \arrow[d, "g"] & y_{t+1} \\
x_t \arrow[r, "f"] & x_{t+1} \arrow[u, "h"']
\end{tikzcd}
$$

Phase 1: Search for an unfolding $g : Y \to S$ and folding $h : S \to Y$ such that

$$\left\| y_t - h \circ g(y_t) \right\| \to \min_{g,h}$$

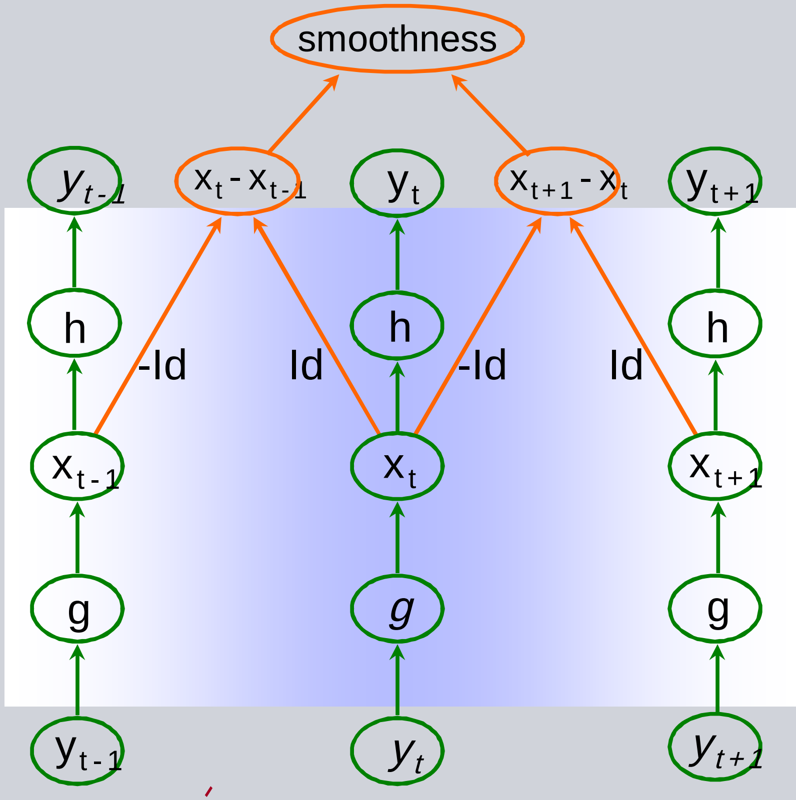
and the flow $x_t = g(y_t)$ is ***smooth over time***.

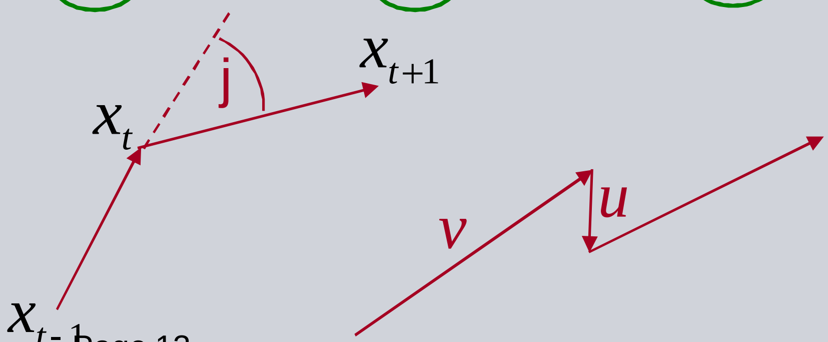
Phase 2: Search for a dynamics $f : S \to S$, using fixed $g()$, $h()$ such that

$$\left\| y_{t+1} - h \circ f \circ g(y_t) \right\| \to \min_{f}$$

SIEMENS

Optimality Conditions of Normalized Curvature





Smoothness Penalty (normalized curvature):

$$P = \frac{1}{T}\sum_{t=1}^{T} \frac{\left((x_{t+1} - x_t) - (x_t - x_{t-1})\right)^2}{(x_{t+1} - x_t)^2 + (x_t - x_{t-1})^2} \rightarrow \min$$

Lemma: With $u = (x_{t+1} - x_t)$ & $v = (x_t - x_{t-1})$ we get

$$(1) \quad P = \frac{1}{T}\sum_{t=1}^{T} \left(1 - 2\frac{|u| \cdot |v|}{|u|^2 + |v|^2}\cos\varphi\right) \rightarrow \min$$

$$(2) \quad \text{range of P}: \quad 0 \leq \text{smooth} < 1 < \text{zigzagging} \leq 2$$

Features of the above Smoothness Penalty:

1: $P \rightarrow min$: enforces angle smoothness

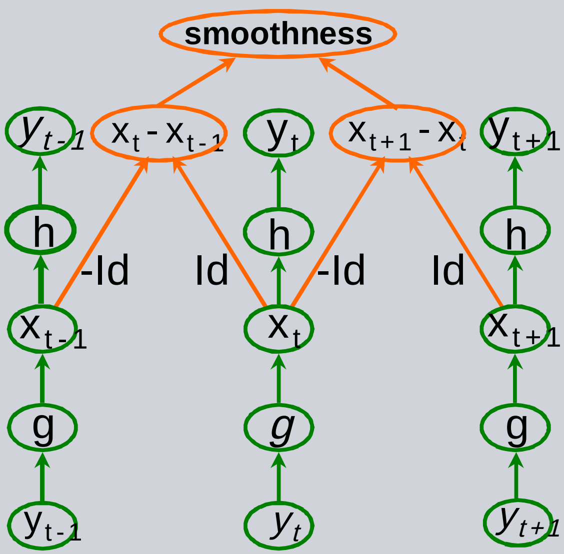
2: $P \rightarrow min$: enforces constant velocity

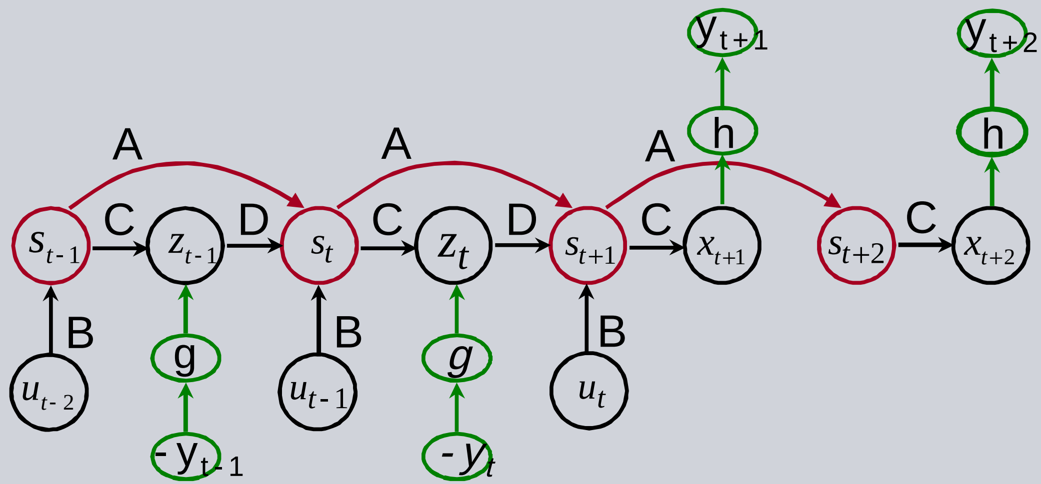
3: $P \rightarrow min$: enforces partly noise cancellation

SIEMENS

Smooth Unfolding in Space & Time for Error Correction Networks I

First, the observables are transformed such that the observed dynamics shows up smoother in the new coordinate system. This might generate a higher dimensional space of indicators. Then the smoothed dynamics is identified and forecasted. Finally the smoothed variables are transformed back to interpretable variables.

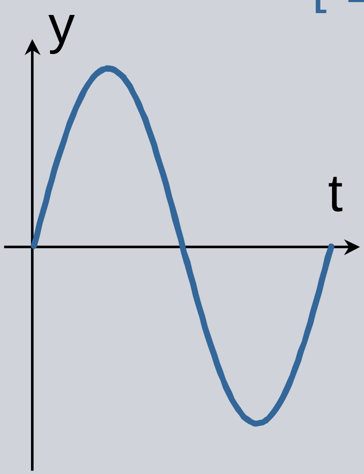




Unfolding of an Oscillator

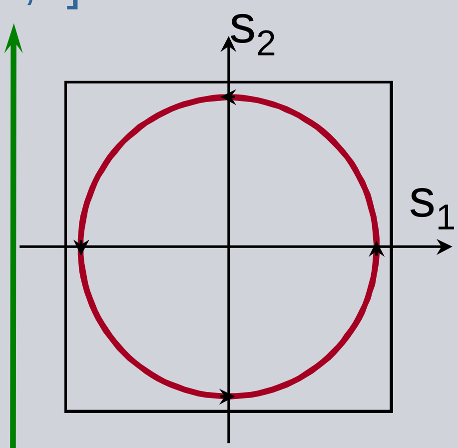
$[-1,1]^2$ → $[-1,1]^2$

$$y_{t+1} = \cos\left(\frac{2\pi}{T}\right) y_t - \sin\left(\frac{2\pi}{T}\right) \operatorname{sgn}\left(y_t - y_{t-1}\right)\sqrt{1 - y_t^2}$$



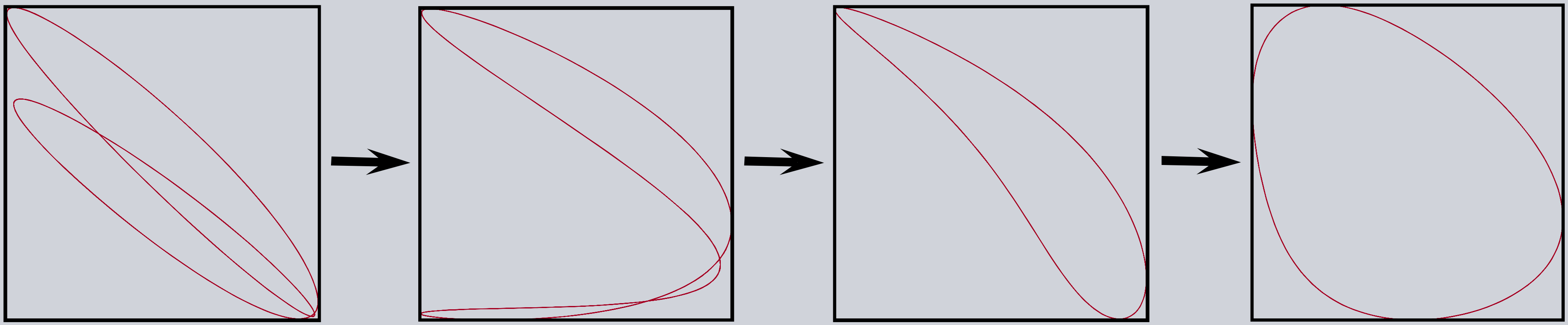
$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}_t = \begin{pmatrix} y_t \\ \operatorname{sgn}(y_t - y_{t-1})\sqrt{1 - y_t^2} \end{pmatrix}$$

$$y_t = (x_1)_{t+1}$$



$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}_{t+1} = \begin{pmatrix} \cos\left(\frac{2\pi}{T}\right) & -\sin\left(\frac{2\pi}{T}\right) \\ \sin\left(\frac{2\pi}{T}\right) & \cos\left(\frac{2\pi}{T}\right) \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}_t$$

$[-1,1]^2$ → $[-1,1]^2$

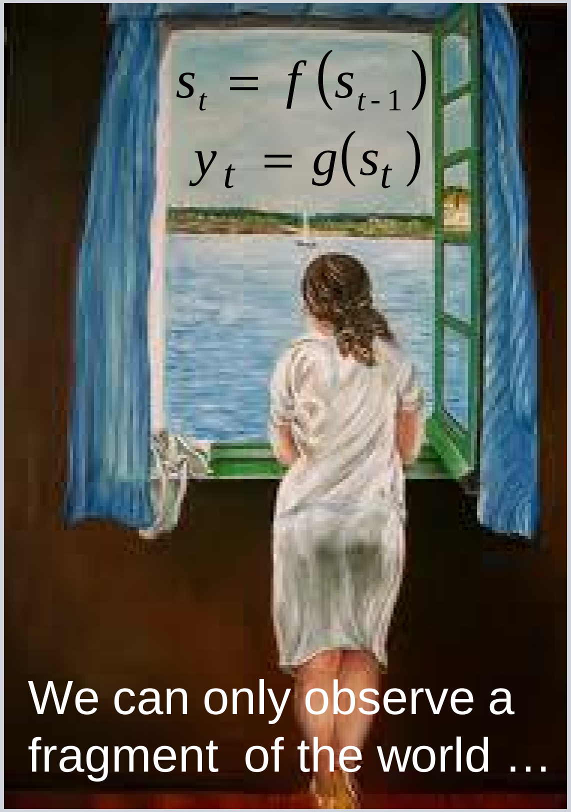


Modeling Closed (Large) Dynamical Systems

with

Recurrent Neural Networks

Modeling Closed Dynamical Systems with Recurrent Neural Networks (*HCNN*)

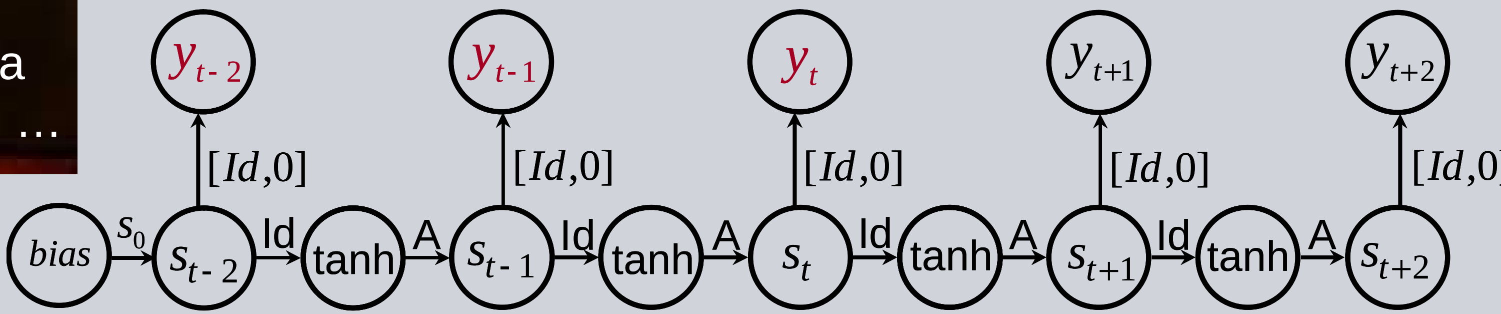


$$s_t = A\tanh(s_{t-1}) \quad , s_0$$ state transition

$$y_t = [Id, 0]\, s_t$$ output equation

$$\sum_{t=1}^{T} \left(y_t - y_t^d\right)^2 \to \min_{A, s_0}$$ identification

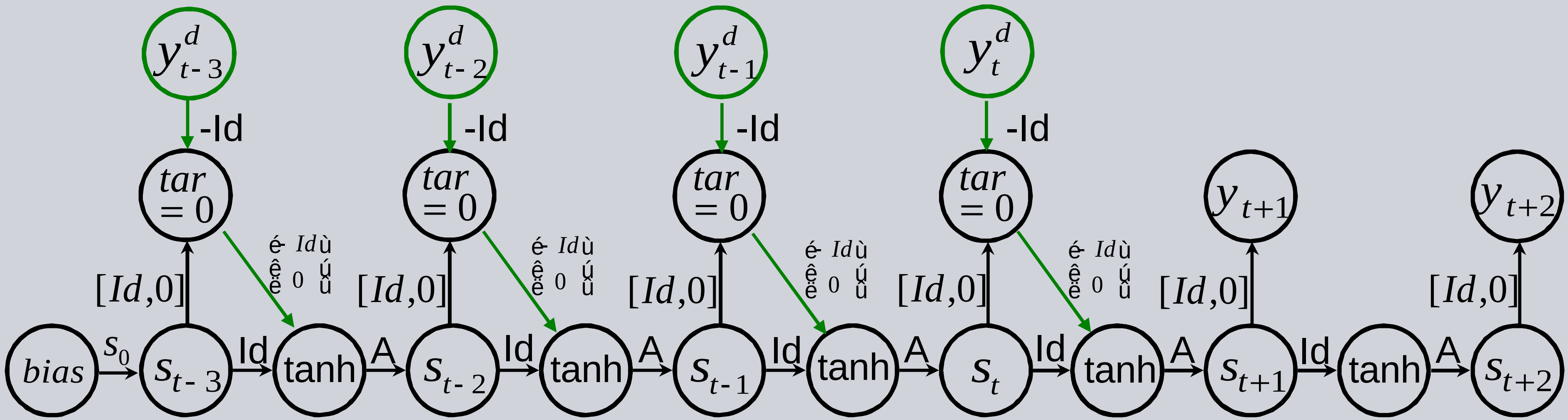
The model is unfolded along ***history*** → only 1 training example



… but to understand the dynamics of the observables, we have to reconstruct at least a part of the hidden states of the world. Forecasting is based on observables and hidden states.

The Role of Observations in the Identification of Dynamical Systems

Embed the original architecture into a larger architecture, which is easier to learn. After the training, the extended architecture has to converge to the original model.

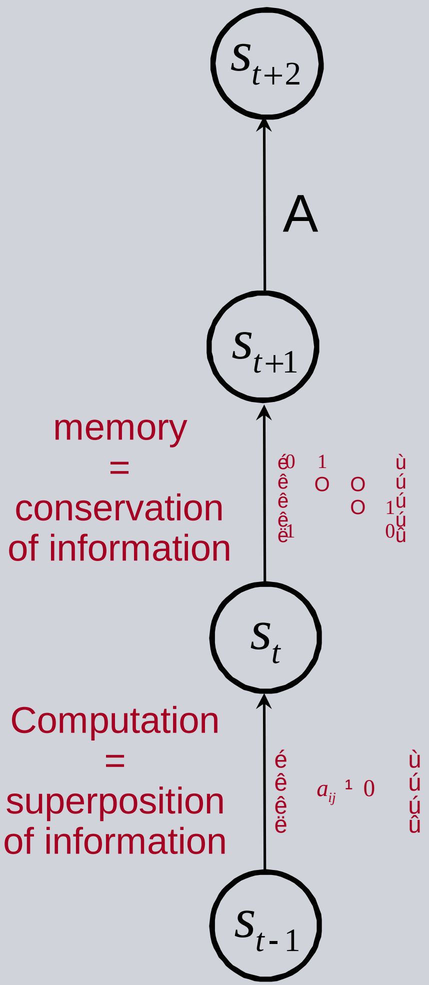


- The essential task is NOT to reproduce the past observations, but to identify related hidden variables, which make the dynamics of the observables reasonable.
- Use an architectural teacher-forcing (**ATF**) to support the learning of the HCNN. Replace expectations y_t by observations y_t^d: $r_t^{upper} = y_t - (y_t - y_t^d) = y_t^d$ & $e_t^{upper} = e_{t+1} - e_{t+1} + error_t \approx y_t - y_t^d$
- The state flow decomposes in 1) known observables, 2) reconstructed hidden variables and 3) unidentifiable random variables, which act as a net-internal regularization.

SIEMENS

Modeling of Dynamical Systems on Different Time Scales

Structure & Function



To avoid numerical explosions, large recurrent systems cannot be fully connected.

Backpropagation is able to learn systems with around 50 nonzero weights per column.

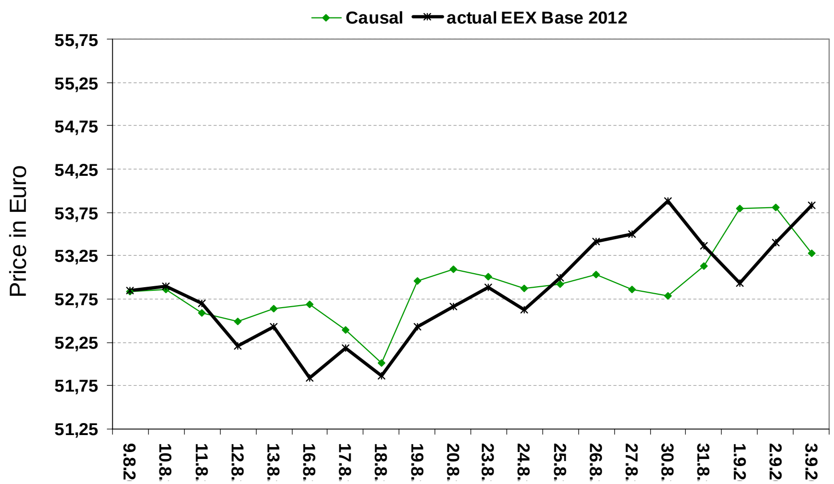
$$sparsity \approx \frac{50}{\dim(A)}$$

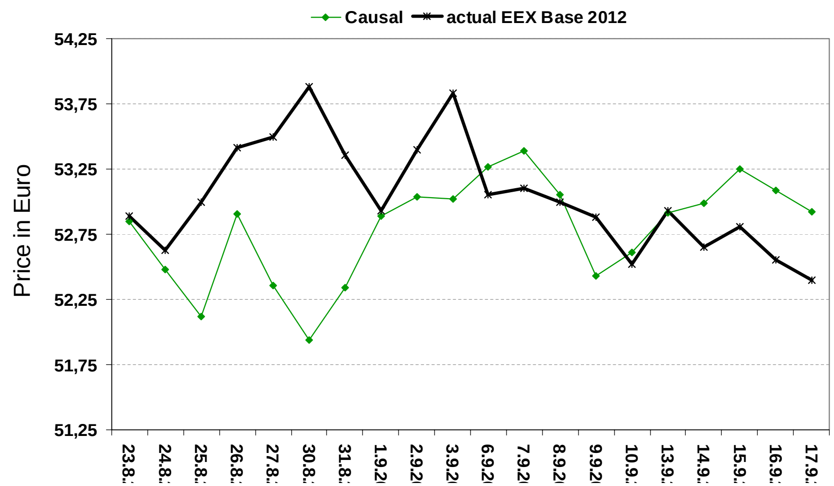
Sparsity is not only a necessary condition for large systems, it describes a tradeoff between memory and computational power.

A random sparse matrix A allows the modeling of dynamical systems on different time scales.

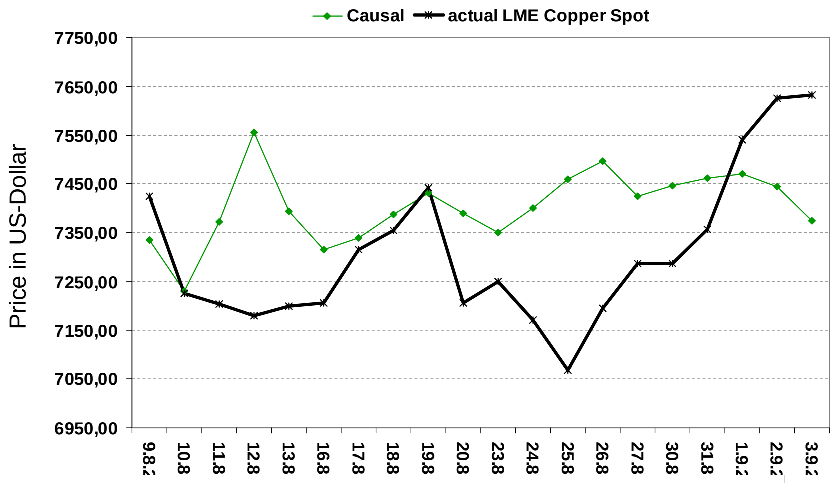
20 Days Price Forecasting with Causal Neural Networks

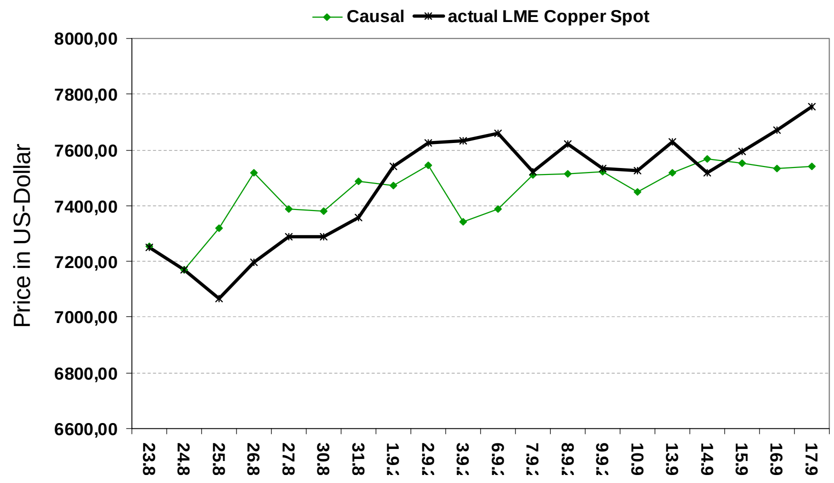
EEX Base Market





LME Copper Market





2010-08-09 – 2010-09-03

2010-08-23 – 2010-09-17

Identification of Goal Oriented Dynamical Systems

When we are able to identify the goals of the agents we can try to explain the dynamics backward from their goals.

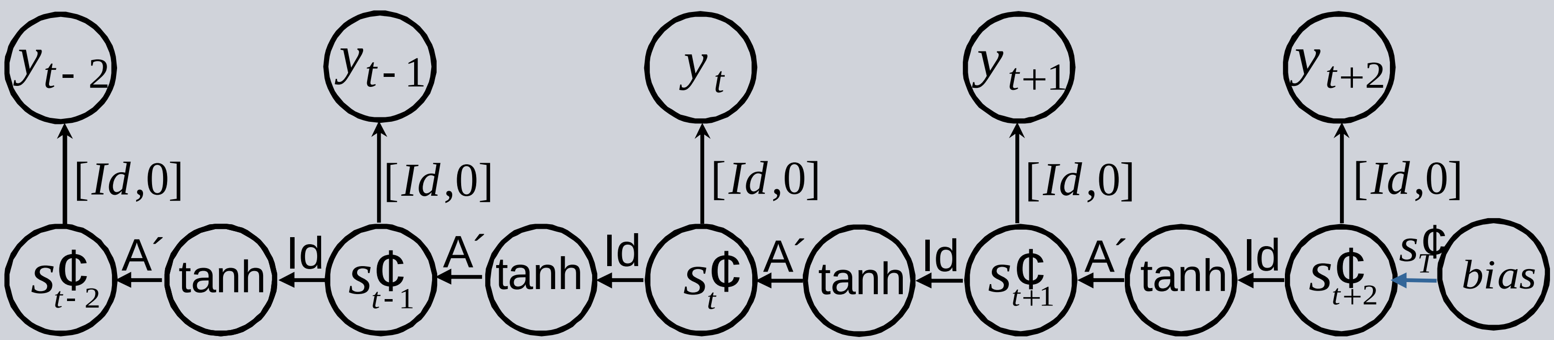
$$s'_t = A' \tanh\left(s'_{t+1}\right)\ ,\ s'_T \qquad \text{state transition}$$

$$y_t = [Id, 0]\, s'_t \qquad \text{output equation}$$

$$\sum_{t=1}^{T} \left(y_t - y_t^d\right)^2 \to \min_{A', s'_T} \qquad \text{identification}$$

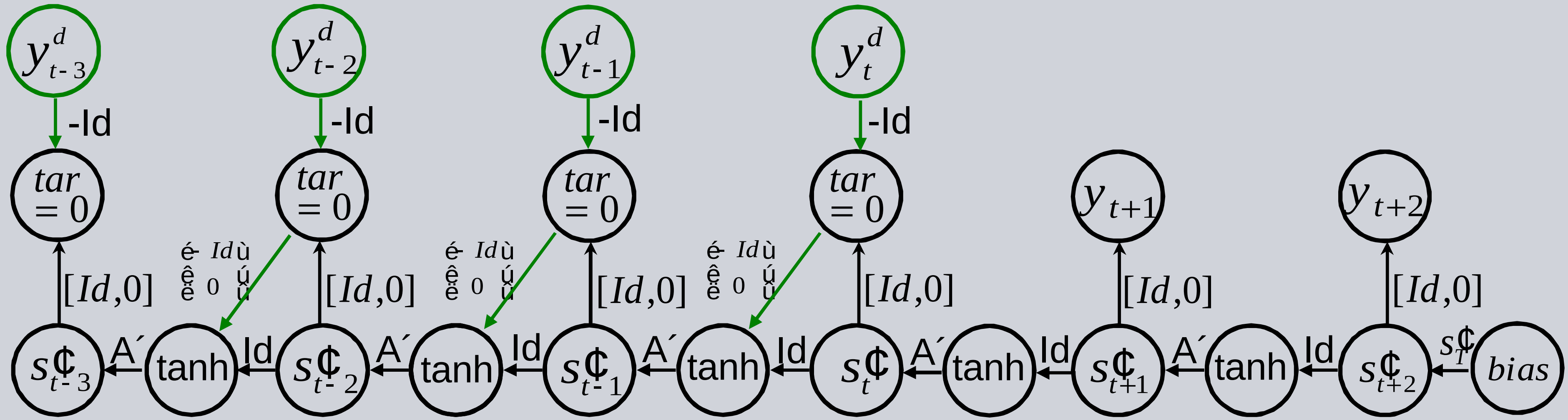
Causal nets are given by an initial state s_0 & matrix A

Retro-causal nets are given by a final state s_T & matrix A´



The Identification of Retro-Causal Dynamical Systems

Embed the original architecture into a larger architecture, which is easier to learn. After the training, the extended architecture has to converge to the original model.

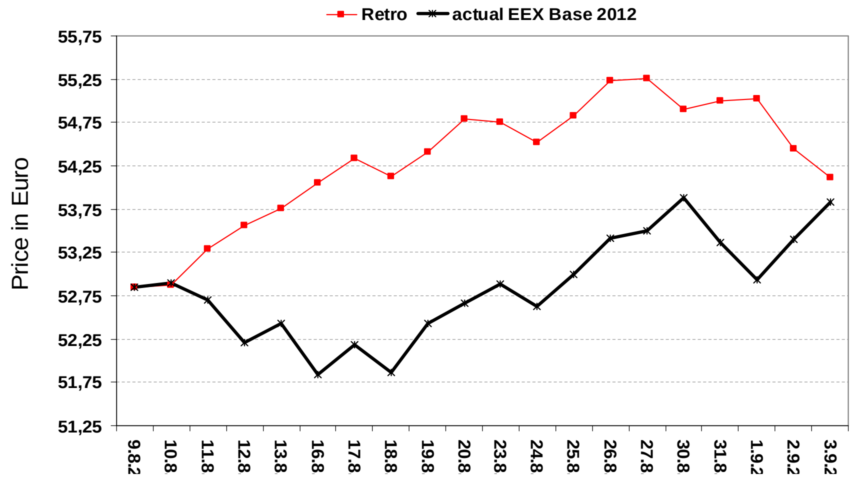


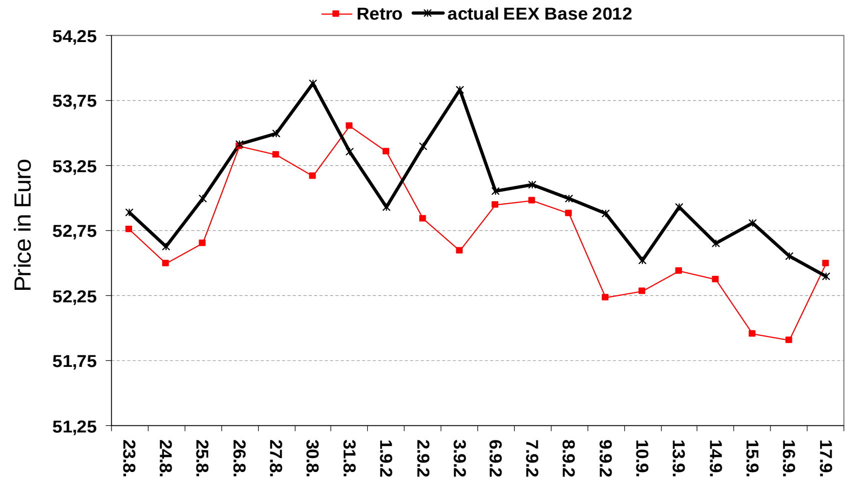
- The essential task is NOT to reproduce the past observations, but to identify related hidden variables, which make the dynamics of the observables reasonable.
- Use a form of an architectural teacher-forcing (**ATF**) to support the learning of the HCNN. Replace expectations y_t by observations y^d_t $\rightarrow$ $r^{upper}_t = y_t - (y_t - y^d_t) = y^d_t$
- Observations of states are a cut in the flow of the forward dynamics as well as in the backward error flow $e^{upper}_t = e_{t+1} - e_{t+1} + error_t \approx y_t - y^d_t$

SIEMENS

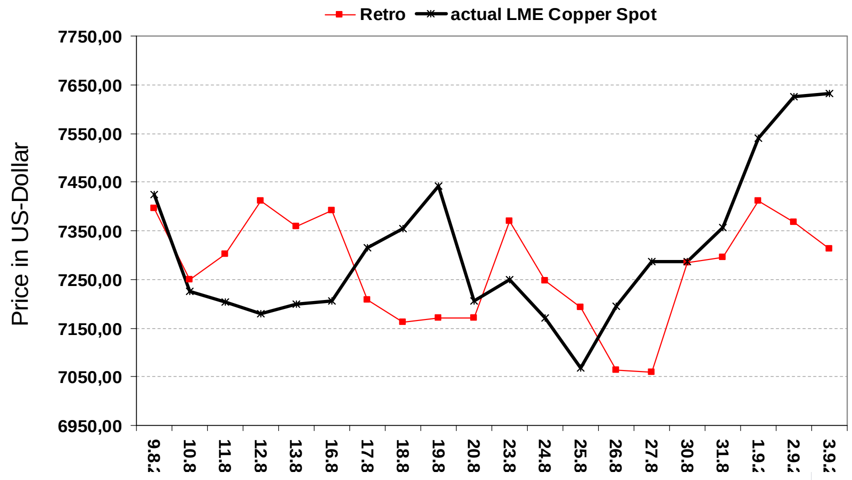
20 Days Price Forecasting with Retro-Causal Neural Networks

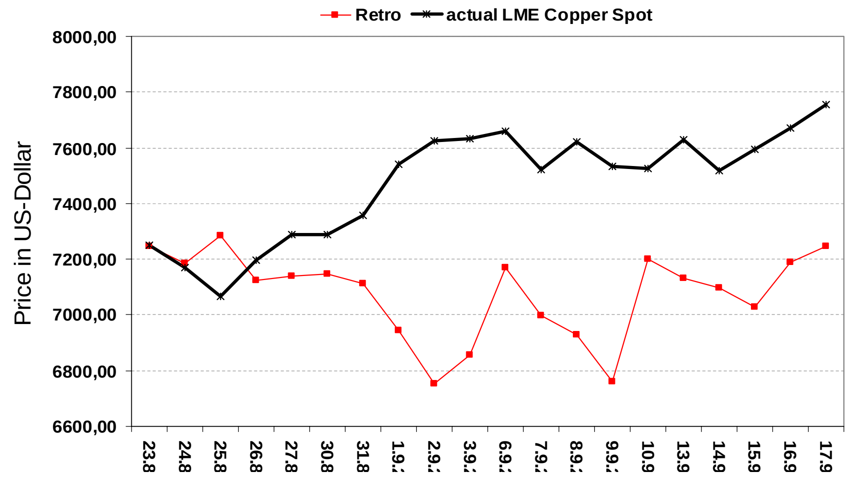
EEX Base Market





LME Copper Market



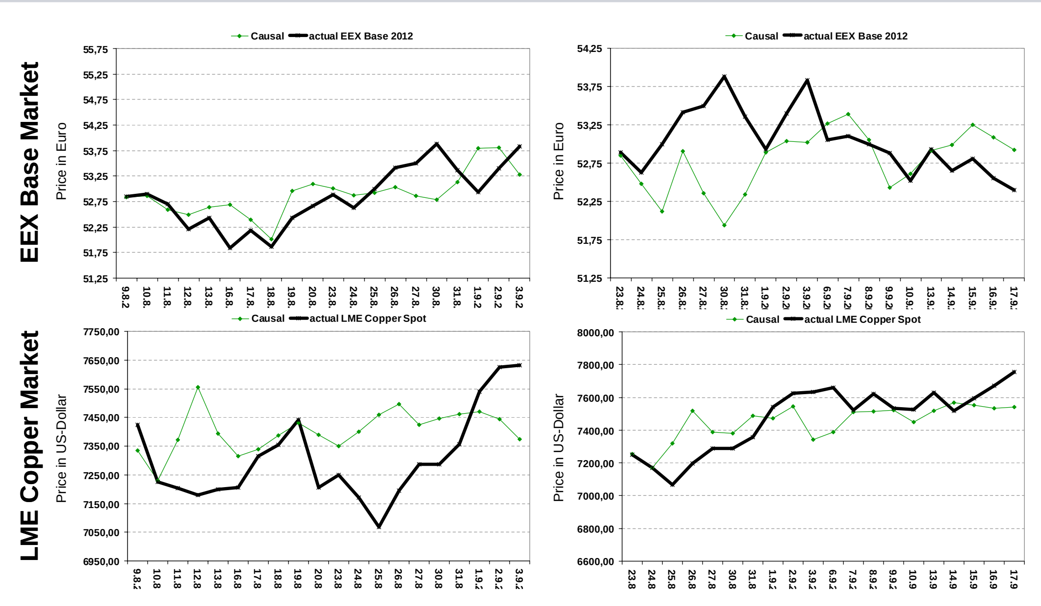


2010-08-09 – 2010-09-03

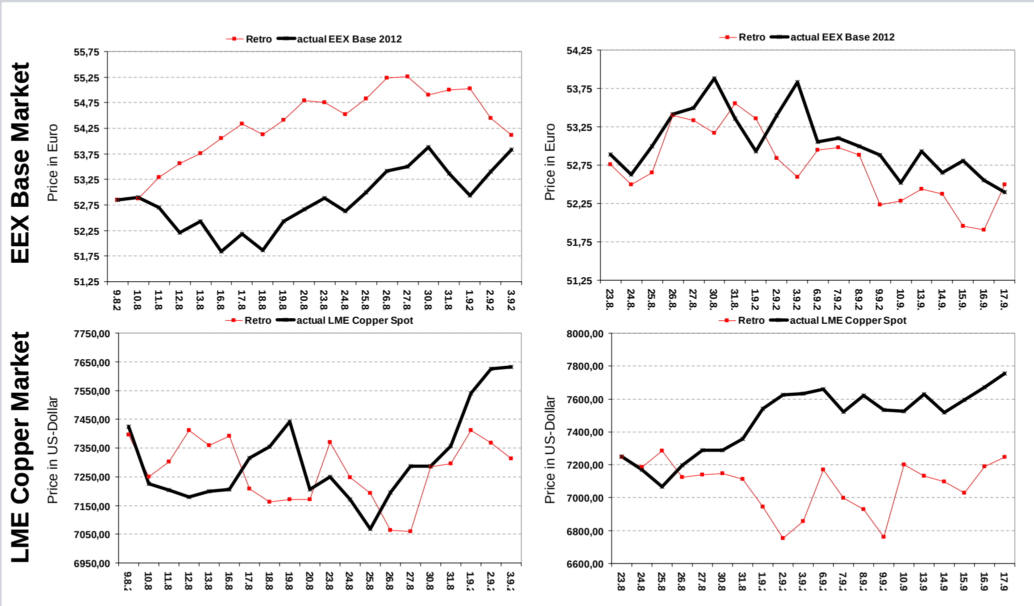
2010-08-23 – 2010-09-17

Comparison of Causal Forecasts and Retro-Causal Forecasts

Causal Forecasts



Retro-Causal Forecasts



Insights: Causal and retro-causal forecasting is complementary.

Ex ante there is no chance to know which approach is the appropriate.

We should find a way to combine the best of both approaches.

Combining Causal & Retro-Causal Neural Networks (CRCNN)

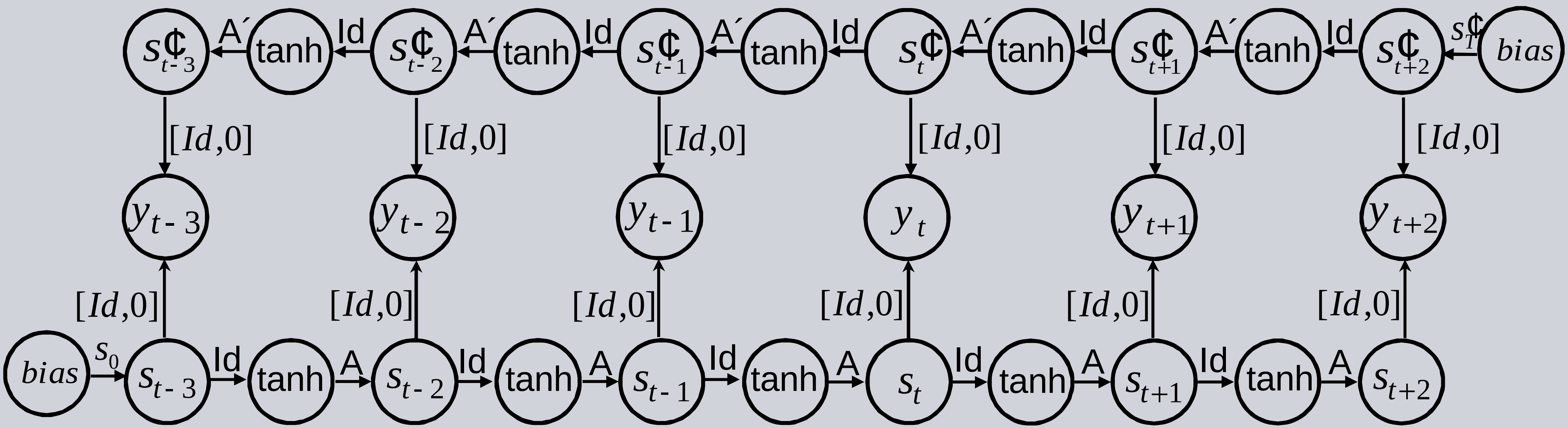
- We explain observations by a symmetric superposition of causal & retro-causal subnets.
- Both subnets are universal learners, but the more appropriate branch learns faster and reduces the error flow of the opposite branch.
- In non-unique optimization, we have to have attention on the path to the optimum!!!

$$s_t = A\tanh(s_{t-1}) \quad , s_0 \qquad \text{causal transition}$$

$$s'_t = A'\tanh(s'_{t+1}) \quad , s'_T \qquad \text{retro transition}$$

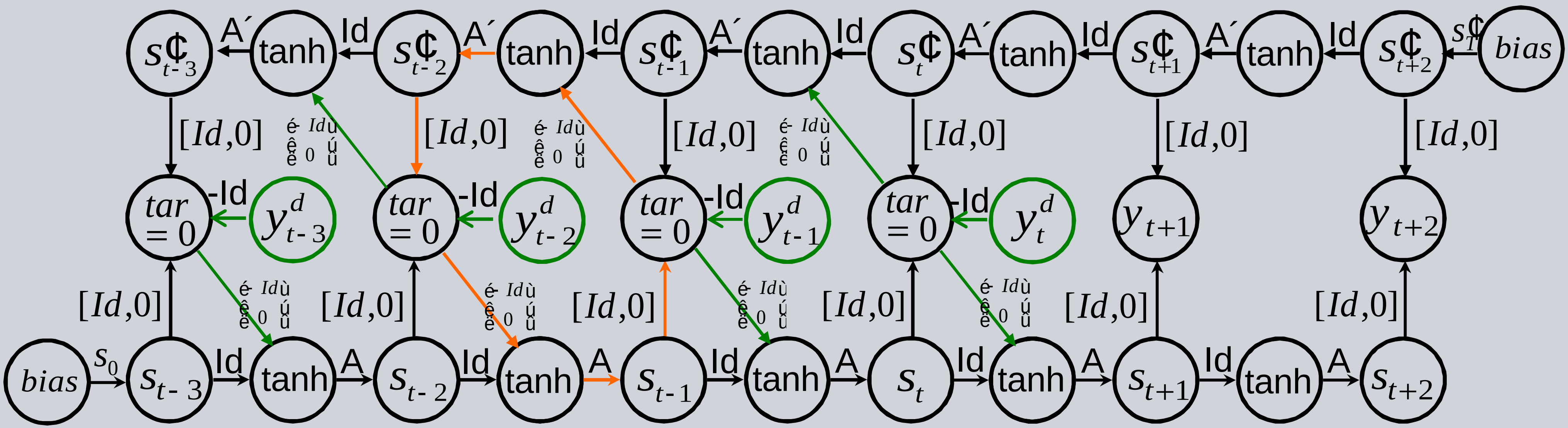
$$y_t = [Id,0]s_t + [Id,0]s'_t \qquad \text{output equation}$$

$$\sum_{t=1}^{T} \left(y_t - y_t^d\right)^2 \to \min_{A,s_0,A',s'_T} \qquad \text{identification}$$



System Identification of Causal-Retro-Causal Networks

Problem 1: Obviously we do not know the future data for the observables.
Thus skip the teacher forcing beyond present time.



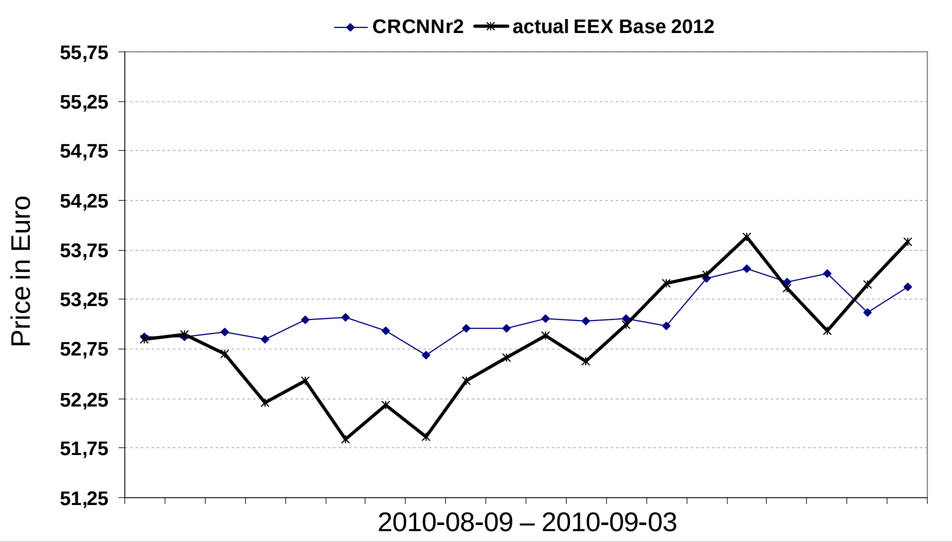
Problem 2: At every time step the network has a fix point loop (equality constraint), defining a manifold.

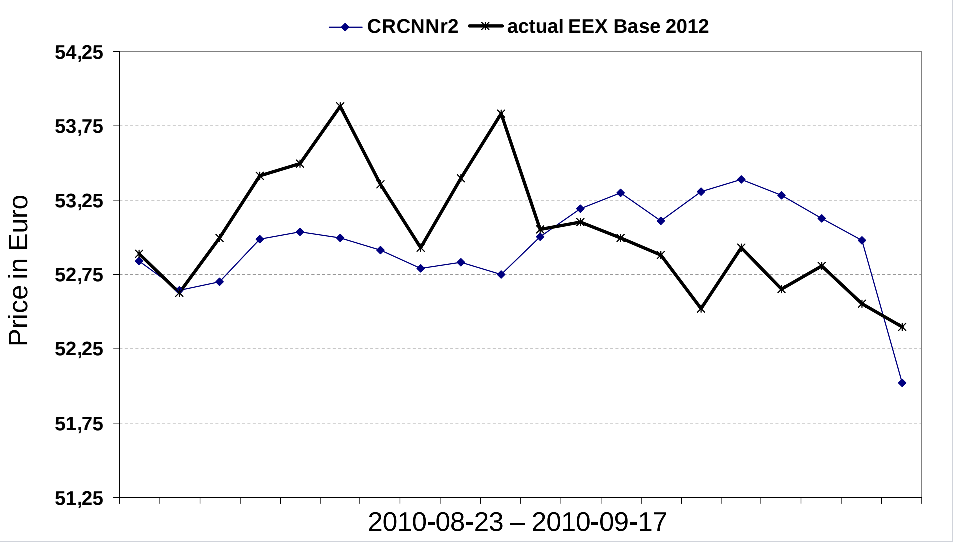
In a joint effort we have to identify the dynamics and the manifold, which is implicitly identified by causal-retro-causal architecture.

SIEMENS

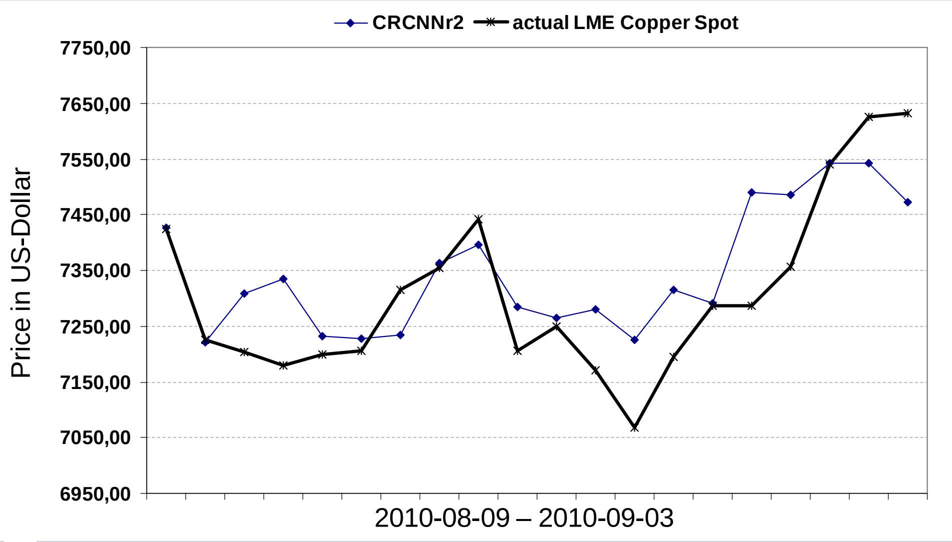
Causal-Retro-Causal Neural Networks (CRCNN)

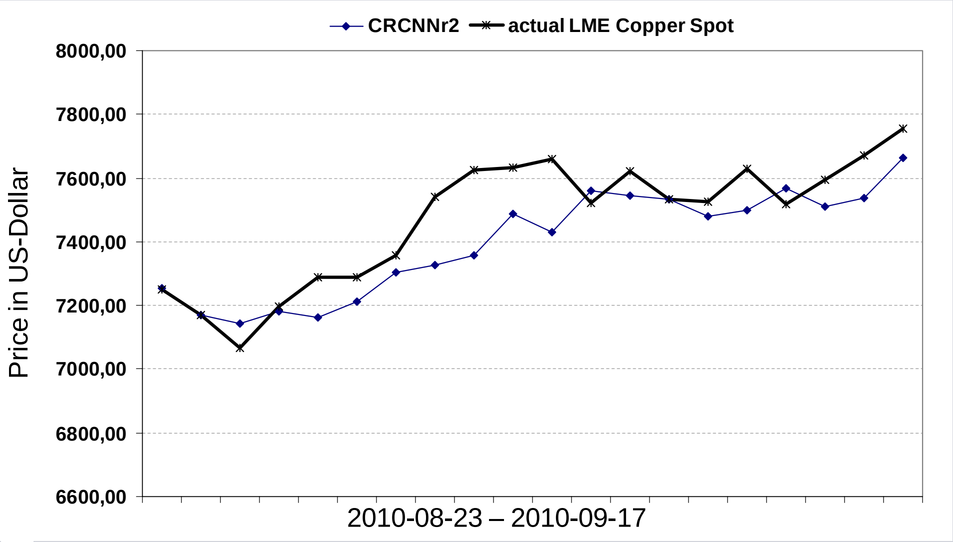
EEX Base Market





LME Copper Market



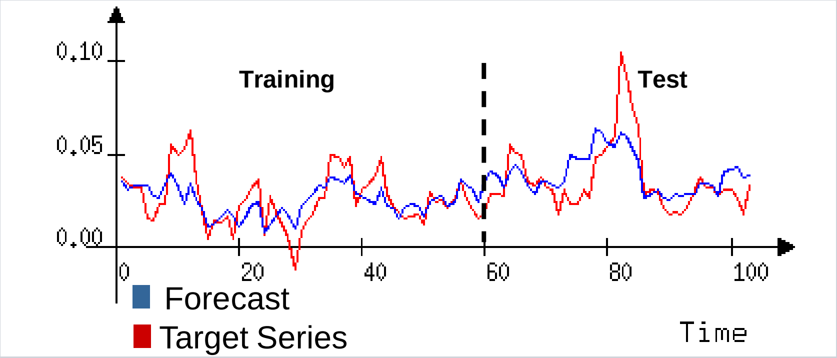


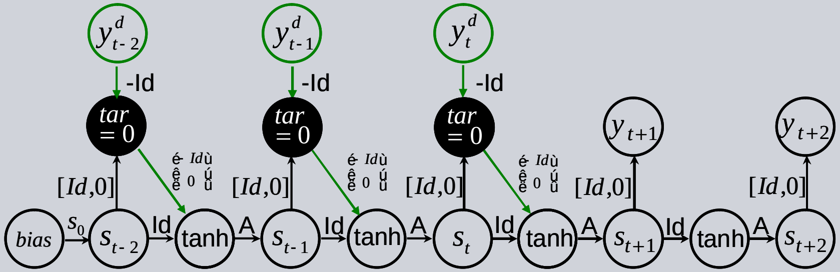
 CT RTC BAM

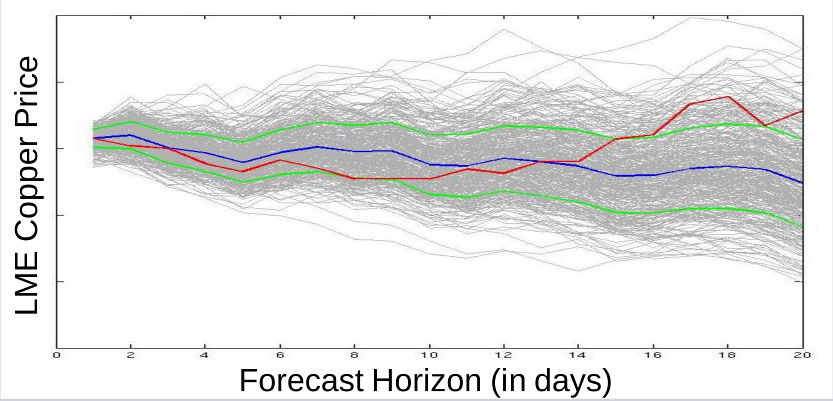
Uncertainty in Forecasting

SIEMENS

Approaches to Model Uncertainty in Forecasting







1 Measure uncertainty as volatility (variance) of the target series. The underlying forecast model is a constant. Thus sin(ωt) can be highly uncertain!??

2 Build a forecast model. The error is interpreted as uncertainty in form of additive noise. The width of the uncertainty channel is constant over time.

3 Describe uncertainty as a diffusion process (random walk). The diffusion channel widens over time, e.g. scaled by the one-step model error.

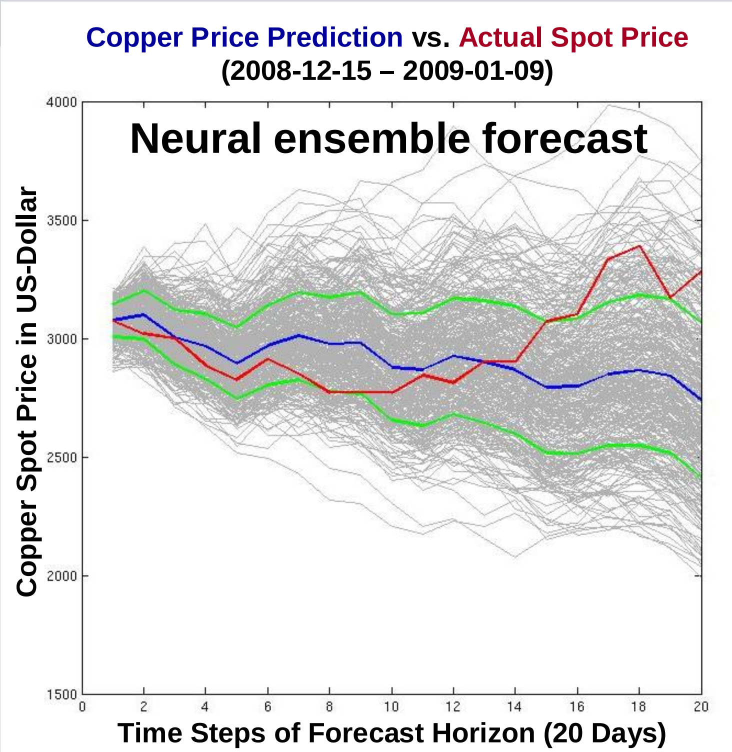
For large systems 2 & 3 fail: We have to learn to zero error → the uncertainty channel disappears.

4 One large model doesn't allow to analyze forecast uncertainty, but an ensemble forecast shows the characteristics of an uncertainty channel: Given a finite set of data, there exist many perfect models of the past data, showing different future scenarios caused by different estimations of the hidden states.

SIEMENS

From Model Uncertainty to Forecast Uncertainty (Risk)

Obviously, the ensemble distribution describes <u>model uncertainty</u>. What do we learn about <u>forecast uncertainty</u>?



Conjecture for large recurrent neural networks:
forecast uncertainty = model uncertainty

Concept of proof:

forecast uncertainty ≥ model uncertainty

Given past observations & prior information, every ensemble member is a reasonable forecast. All of the large RNNs fit the past perfectly, but we do not know which one is the true further development.

model uncertainty ≥ forecast uncertainty

For large recurrent neural networks we have:

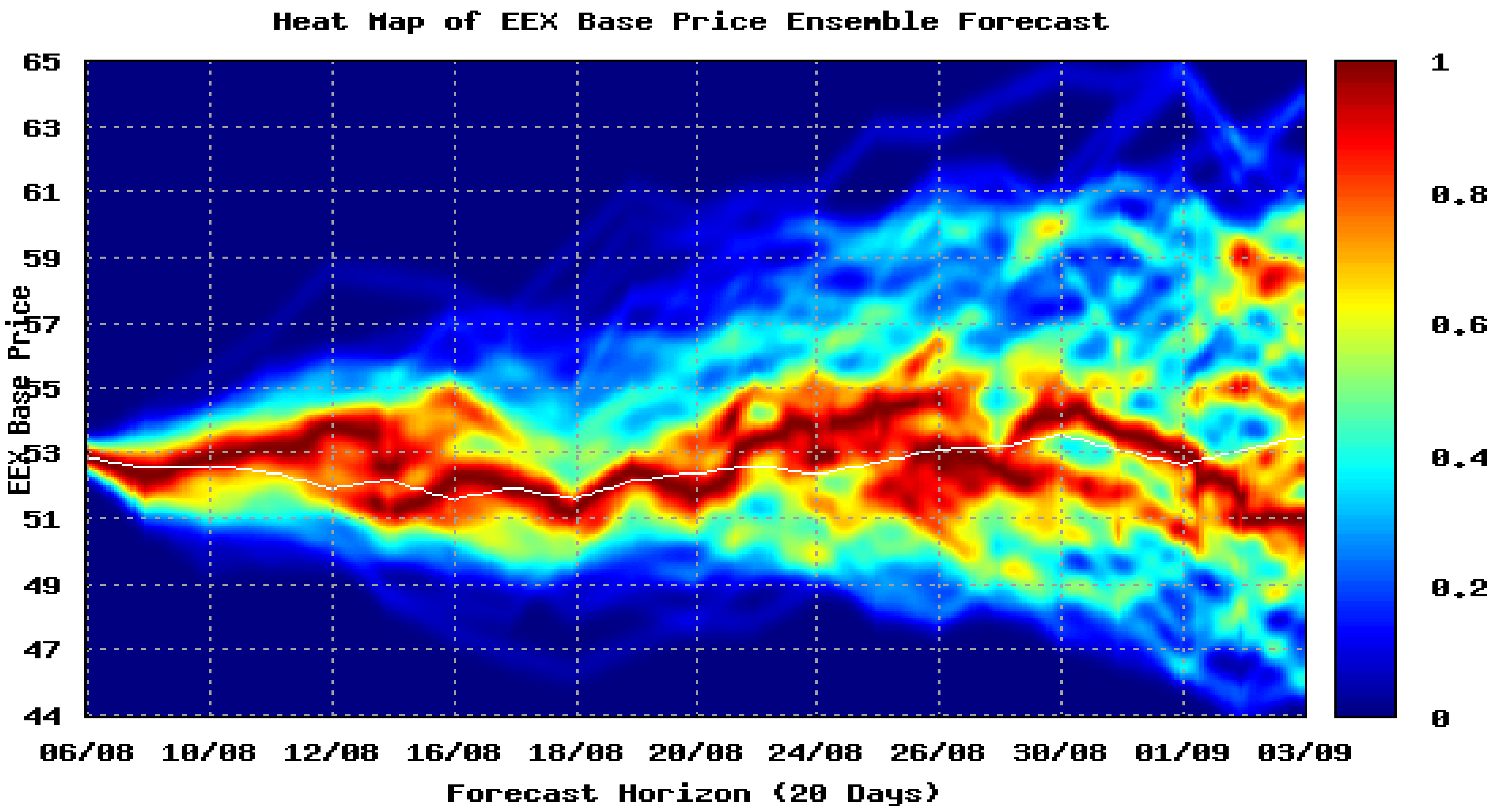
(a) Because of the universal approximation the ensemble contains all black swans.

(b) The distribution is independent of the modeling details → it contains information.

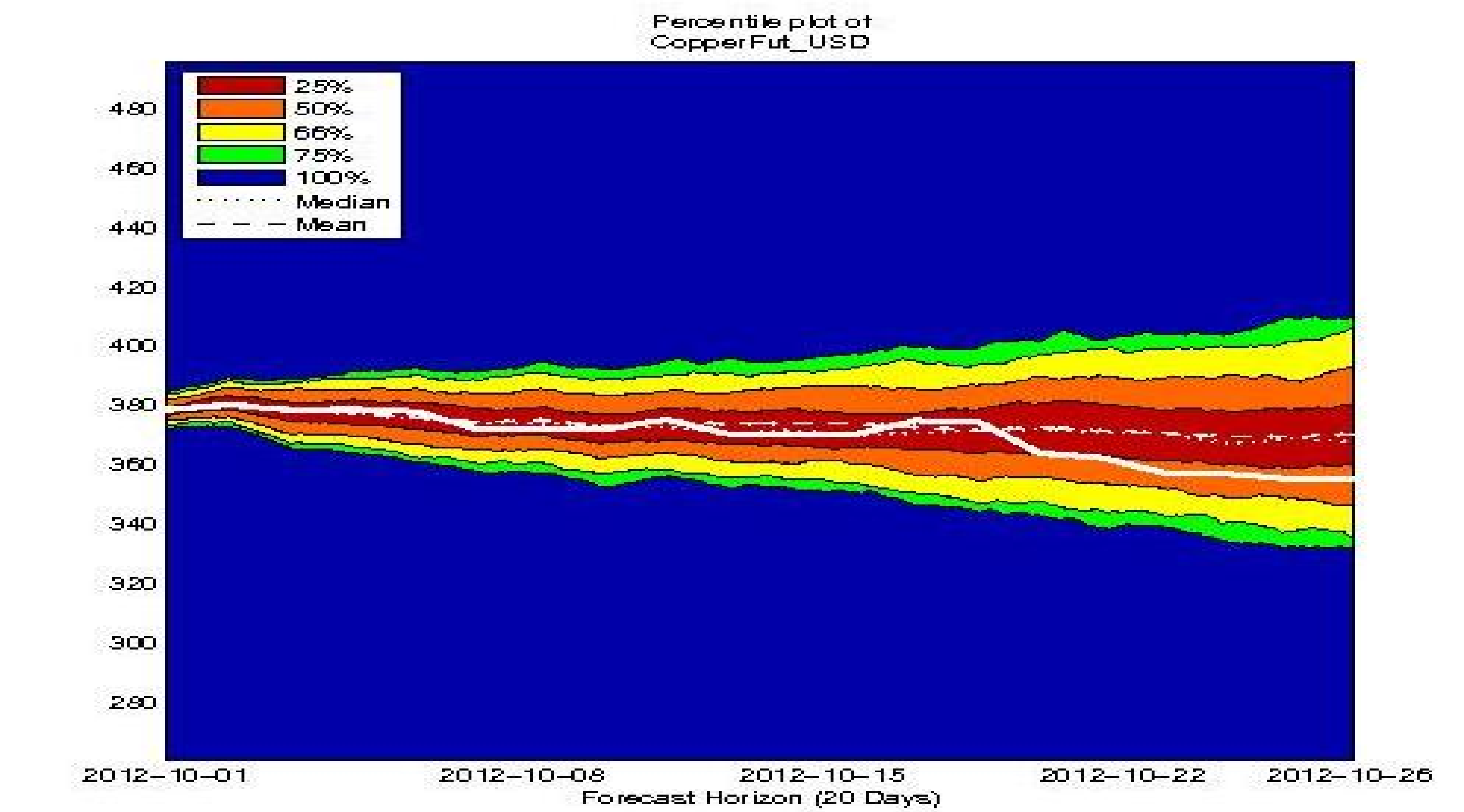
(c) The form of the distribution depends on data and an interacting decision dynamics.

SIEMENS

EEX Electricity Price Heatmap



Copper Price Forecasting in Form of Quantile Ranges



Commodity Price Forecasting Study over 55 Weeks

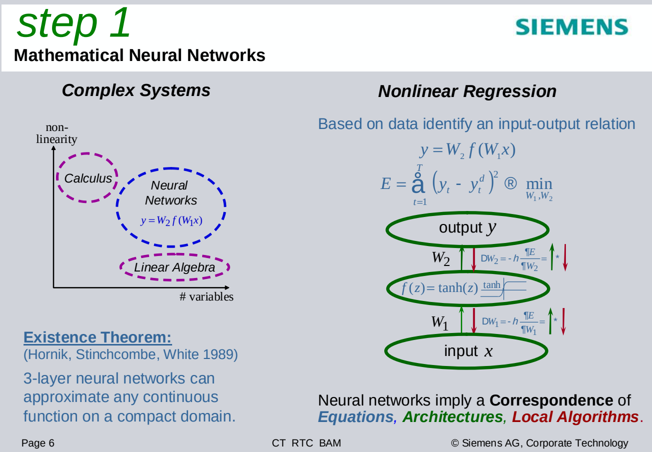
ensemble / target	25% Quantile	50% Quantile	66% Quantile	75% Quantile	Procurement	Trading
CAUSAL_a17	17.81%	40.78%	54.53%	63.75%	-7.33	-1.26
RETRO_a17	28.75%	57.34%	74.06%	84.06%	-4.33	-20.68
CRCNN2_a17	38.59%	69.06%	81.56%	88.28%	9.98	25.23
CRCNN4_a17	39.38%	65.16%	79.84%	85.94%	4.14	14.08
CRCNN6_a17	36.72%	68.13%	80.63%	86.72%	6.93	10.99
CRTNN_a17	23.75%	45.94%	59.38%	68.44%	-13.33	-24.40
CRTNN_a29	27.19%	50.63%	65.47%	74.06%	-14.42	-16.48
CRTNN_a33	28.28%	52.19%	65.47%	75.00%	-9.57	-3.71

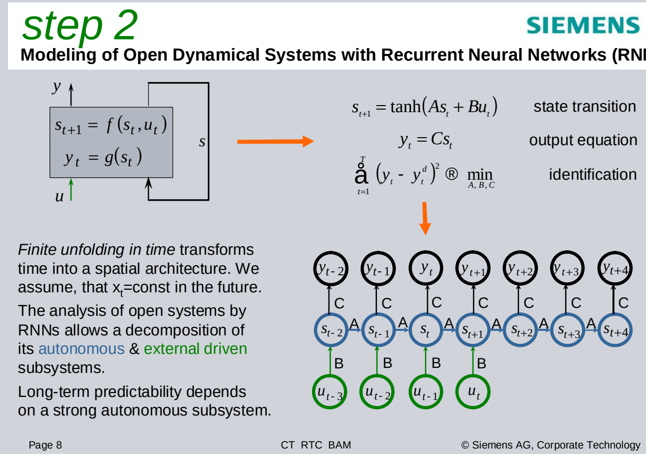
The confidence intervals are measured on basis of the ensemble width. They should contain the same number of target values (ensemble size 200 models).

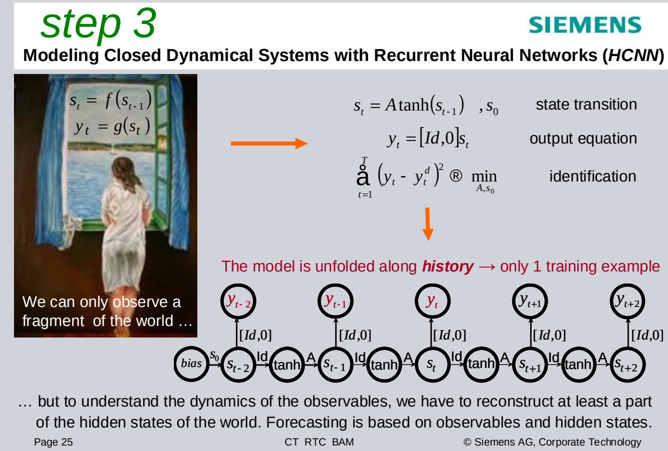
Insight: In CRTNNs the network sparsity acts as a diffusion parameter. Thus the diffusion functionality allows the identification of the sparsity structure.

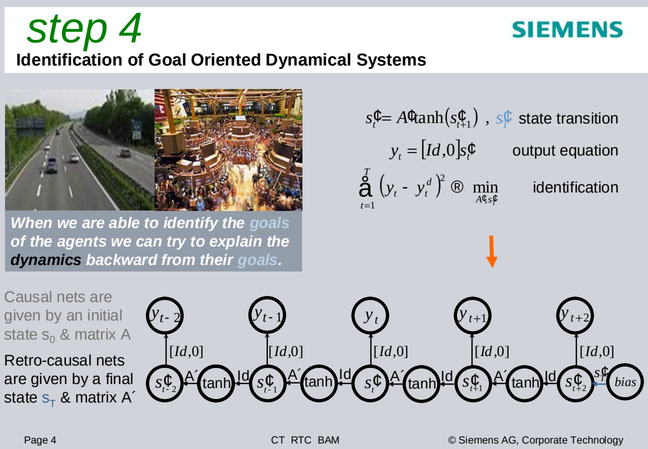
SIEMENS

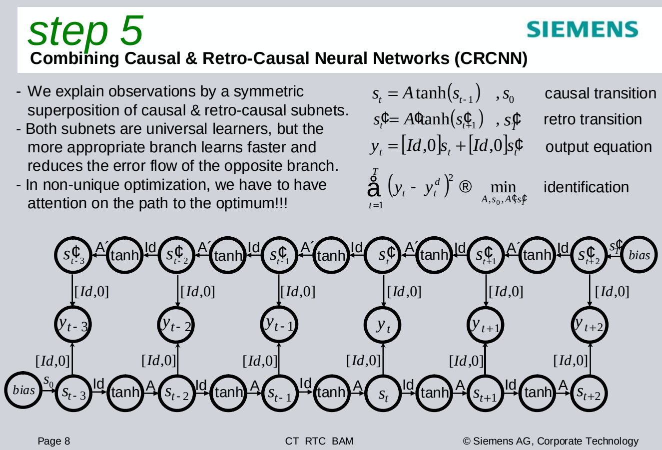
Is Causality a Useful Concept ?











1 FFNNs have a clear causality concept, problems: nonlin, dim;

2 Small RNNs introduce lagged causalities;

3 HCNNs have a time direction, no chicken – egg problems, reconstructed hidden states play an essential role;

4 RetroNNs have an inverted time direction;

5 CRCNNs are fully symmetric;

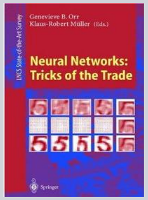
In general, *causality* (as an intellectual concept) & *predictability* (as a mathematical description) are mutually exclusive.

See also YouTube (keywords: Zimmermann, Siemens)

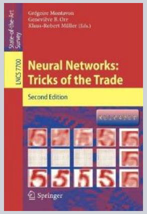
SIEMENS

Selected References

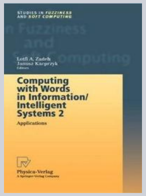
Zimmermann, H.G.: Neuronale Netze als Entscheidungskalkül, In: Neuronale Netze in der Ökonomie; Eds.: Rehkugler, H., Zimmermann, H.G., p. 1-87; Franz Vahlen; 1994; ISBN 3-8006-1871-0



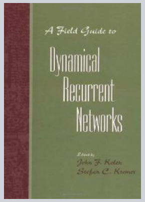
Neuneier, R.; Zimmermann, H.G.: How to Train Neural Networks, In: Neural Networks: Tricks of the Trade; Eds.: Orr G. B., Müller K. R.; p. 373-423; Springer, 1998; ISBN 3-540-65311-2



Zimmermann, H.G.; Tietz, C.; Grothmann, R.: Forecasting with Recurrent Neural Networks, In: Neural Networks: Tricks of the Trade, **Second Edition**; Eds.: Montavon, G; Orr G. B., Müller K.R.; p. 687-707; Springer, 2012; ISBN 978-3-642-35288-1

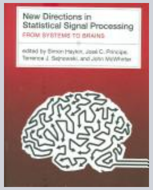


Siekmann, S.; Neuneier, R.; Zimmermann, H.G.; Kruse R.: Neuro Fuzzy Systems for Data Analysis, In: Computing with Words in Information / Intelligent Syst. 2; Eds.: Zadeh L. A., Kacprzyk J.;p. 35-74; Physica Verlag; 1999



Zimmermann, H.G.; Neuneier, R.: Neural Network Architectures for the Modeling of Dynamical Systems, In: A Field Guide to Dynamical Rec. Networks; Eds.: Kolen J. F., Kremer S. C.; IEEE Press; 2000

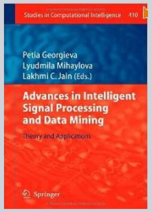
Zimmermann, H.G.; Neuneier, R.; Grothmann, R.: Modeling of Dynamical Systems by Error Correction Neural Networks, In: Modeling and Forecasting Financial Data, Techniques of Nonlinear Dynamics; Eds. Soofi, A. and Cao, L., Kluwer, 2002



Zimmermann, H.G, Grothmann, R., Schäfer, A.M., Tietz, Ch.: Identification and Forecasting of Large Dynamical Systems by Dynamical Consistent Neural Networks, in: Haykin, S.; et. al. [Eds.]: New Directions in Statistical Signal Processing: From Systems to Brain, MIT, 2006.

Zimmermann, H.-G., Grothmann, R., Tietz Ch and von Jouanne-Diedrich, H: Market Modeling, Forecasting and Risk Analysis with Historical Consistent Neural Networks, in: Proceedings of the International Conference on Operations Research, Munich 2010, Springer, 2011

Zimmermann, H.-G., Grothmann, R., Tietz, Ch.: Forecasting market prices with causal-retro-causal neural networks. In: Lüthi, H.-J. et. al (eds.): OR Proceedings 2011: Selected Papers of the Int. Conference on Operations Research (OR 2011), Zurich, Springer, 2012



Zimmermann, H.G, Grothmann, R., Tietz, Ch.: Historical Consistent Neural Networks: New Perspectives on Market Modelling, Forecasting and Risk Analysis, in: Georgieva P. et al. (Eds): Advances in Intelligent Signal Processing and Data Mining: Theory and Applications, Springer 2012