ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/278654704

Market Modeling, Forecasting and Risk Analysis with Historical Consistent Neural Networks

Chapter · June 2011

DOI: 10.1007/978-3-642-20009-0_84

CITATIONS
8

READS
763

4 authors, including:

Hans Georg Zimmermann
Fraunhofer-Gesellschaft zur Förderung der angewandten Forschung e.V.
85 PUBLICATIONS **1,289** CITATIONS

SEE PROFILE

Ralph Grothmann
Siemens
36 PUBLICATIONS **305** CITATIONS

SEE PROFILE

Holger K. von Jouanne-Diedrich
Technical University Aschaffenburg
12 PUBLICATIONS **27** CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:

Project Advanced Adaptive Architectures for Asset Allocation View project

Project Project TANGO@Siemens View project

All content following this page was uploaded by Hans Georg Zimmermann on 14 March 2021.

The user has requested enhancement of the downloaded file.

Market Modeling, Forecasting and Risk Analysis with Historical Consistent Neural Networks

Hans-Georg Zimmermann, Ralph Grothmann, Christoph Tietz and Holger von Jouanne-Diedrich

1 Introduction

Business management requires precise forecasts in order to enhance the quality of planning throughout the value chain. Furthermore, the uncertainty in forecasting has to be taken into account.

Neural networks (NN) offer significant benefits for dealing with the typical challenges associated with forecasting. With their universal approximation properties, NN make it possible to describe non-linear relationships between a large number of factors and multiple time scales[Hay08]. In contrast, conventional econometrics (such as ARMA, ARIMA, ARMAX) remain confined to linear systems[Wei90]. A wide range of models is discussed within the class of neural networks. For example, in terms of the data flow, it is possible to draw a distinction between feedforward and (time) recurrent NNs[CK01]. In this paper we focus on recurrent NN.

It is noteworthy that any NN equation can be expressed as an architecture which represents the individual layers in the form of nodes and the connections between the layers in the form of links. This relationship will be described hereinafter as the correspondence principle between equations, architectures and the local algorithms associated with them. The use of local algorithms provides an elegant basis for the expansion of the NN towards the modeling of large systems.

In this article, we present a new type of recurrent NN, called *historical consistent neural network* (HCNN). HCNNs allow the modeling of highly-interacting non-linear dynamical systems across multiple time scales. HCNNs do not draw any distinction between inputs and outputs, but model observables embedded in the dynamics of a large state space.

In the following, sec. 2 is dedicated to the theoretical foundations of HCNNs. Sec. 3 reports on the application of HCNN to analyze the risk in financial markets, while sec. 4 summarizes the primary findings and points to practical applications.

Siemens AG, Corporate Technology, Otto-Hahn-Ring 6, 81730 Munich, e-mail: Hans_Georg.Zimmermann@siemens.com

2 Historical Consistent Neural Networks (HCNN)

2.1 Modeling open dynamic systems with RNN

To derive the HCNN, let us first consider a simple recurrent neural network (RNN). Our guideline for the development is the correspondence principle of NN (see sec. 1). We start from the assumption that a time series y_τ is created by an open dynamic, which can be described in discrete time τ using a state transition and output equation[Hay08]:

$$\text{state transition } s_\tau = f(s_{\tau-1}, u_\tau) \quad (1)$$
$$\text{output equation } y_\tau = g(s_\tau) \quad (2)$$
$$\text{system identification } E = \frac{1}{T}\sum_{\tau=1}^{T}\left(y_\tau - y_\tau^d\right)^2 \to \min_{f,g} \quad (3)$$

The time-recurring state transition equation $s_\tau = f(s_{\tau-1}, u_\tau)$ describes the current state s_τ dependent on the previous system state $s_{\tau-1}$ and the external influences u_τ.

Without loss of generality we can approximate the state space model with a RNN[SZ06, ZGN02]:

$$\text{state transition } s_\tau = \tanh(As_{\tau-1} + Bu_\tau) \quad (4)$$
$$\text{output equation } y_\tau = Cs_\tau \quad (5)$$

To cover the complexity of the original task, the RNN has to incorporate an increased state dimension. We use the technique of finite unfolding in time[Hay08] to transform the temporal equations into a spatial architecture (see Fig. 1).

The training of the RNN can be conducted using the error-back-propagation-through-time algorithm[Hay08]. The underlying idea here is that any RNN can be reformulated into an equivalent feedforward neural network, if matrices A, B and C are identical in the unfolded architecture (shared weights). For algorithmic solution methods, the reader is referred to the overview article by B. Pearlmatter[Pearl01].

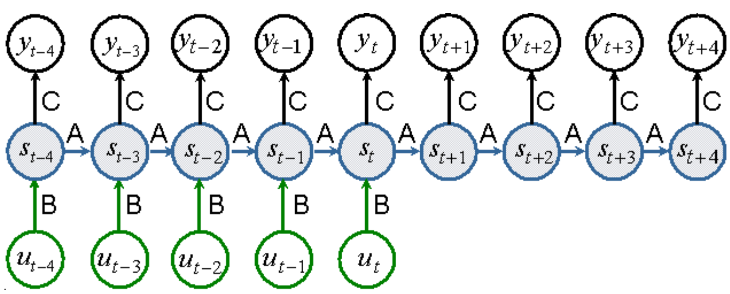

Fig. 1 Unfolded in time recurrent neural network (RNN)[ZGN02]. Note, that the hidden state clusters s_τ are equipped with a tangent hyperbolic tanh() activation function.

2.2 Modeling of closed dynamic systems with HCNN

The RNN is used to model and forecast an open dynamic system using a non-linear regression approach. Many real-world technical and economic applications must however be seen in the context of large systems in which various (non-linear) dynamics interact with each other in time. Projected on a model, this means that we do not differentiate between inputs and outputs but speak about observables. Due to the partial observability of large systems, we need hidden states to be able to explain the dynamics of the observables. Observables and hidden variables should be treated by the model in the same manner. The term observables embraces the input and output variables (i. e. $Y_\tau := (y_\tau, u_\tau)$). If we are able to implement a model in which the dynamics of all of the observables can be described, we will be in a position to close the open system.

Motivated by these modeling principles, we develop the HCNN as follows:

$$\text{state transition } s_\tau = \tanh(As_{\tau-1}) \tag{6}$$

$$\text{output equation } Y_\tau = [Id, 0]s_\tau \tag{7}$$

$$\text{system identification } E = \sum_{\tau=t-m}^{t} \left(Y_\tau - Y_\tau^d\right)^2 \to \min_A \tag{8}$$

The HCNN (Eq. 6 and 7) describes the dynamics of all observables by the sequence of states s_τ using a single transition matrix A. The observables ($i = 1, \ldots, N$) are arranged as the first N neurons of a state s_τ, whereas the hidden variables are represented by the subsequent neurons. The connector $[Id, 0]$ is a fixed matrix of appropriate size which reads out the observables. The HCNN is unfolded across the entire time path, i. e. we learn the unique history of the system. The HCNN architecture is depicted in Fig. 2.

Teacher Forcing (TF)[WZ89, Pearl01] was originally introduced as an extension to the algorithmic training procedures of RNNs. In contrast, we formulate TF as a part of the NN architecture, which allows us to learn the NN using standard error-backpropagation-through-time. Fig. 3 deals with the resulting HCNN architecture.

Let us explain the TF mechanism in the extended HCNN architecture (see Fig. 3): In every time step $\tau \leq t$ the expected values for all observables Y_τ are replaced by the observations Y_τ^d using an intermediate (hidden) layer r_τ. Up to present time ($\tau = t$), the output layers hold the difference between expectations Y_τ and observations Y_τ^d. The output layers are given fixed target values of zero. This causes the HCNN to

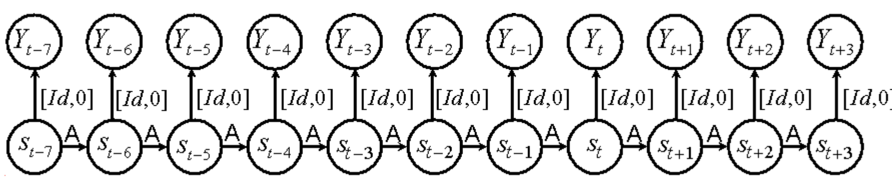

Fig. 2 Architecture of the Historical Consistent Neural Network (HCNN)

learn the expectations Y_τ to compensate for the observations $-Y_\tau^d$. The content of the output layer $(Y_\tau - Y_\tau^d)$, is negated and transferred to the first elements of r_τ using the fixed $[-Id,0]'$ connector. In addition, the expected values Y_τ are transferred from the internal state s_τ to r_τ. The net effect is that the expected values Y_τ on the first N components of the state s_τ are replaced by the corresponding observations Y_τ^d (i. e. $r_{\tau=1,\ldots,N} = Y_\tau - (Y_\tau - Y_\tau^d) = Y_\tau^d$, see Fig. 3). Since all additional connectors for the mechanism introduced are fixed and are used only to transfer data in the network, TF does not lead to a larger number of free network parameters.

In the future part the HCNN is iterated exclusively on the basis of expected values. This turns an open system into a closed dynamical system and we do not need to make the assumption of an constant environment as for dynamical systems. The usage of TF does not reintroduce an input / output modeling, since we replace the expected value of the observables with their actual observations. For sufficiently large HCNNs and convergence of the output error to zero, this architecture converges towards the fundamental HCNN architecture shown in Fig. 2.

Eq. 9 to 11 show the state transition and output equation derived from the HCNN architecture depicted in Fig. 3.

$$\text{state transition } \forall\tau \le t \;\; s_\tau = \tanh\left(A\left(s_{\tau-1} - [Id,0]^T\left(Y_\tau - Y_\tau^d\right)\right)\right) \tag{9}$$

$$\forall\tau > t \;\; s_\tau = \tanh(As_{\tau-1}) \tag{10}$$

$$\text{output equation } \forall\tau \in t \;\; Y_\tau = [Id,0]s_\tau \tag{11}$$

The HCNN models the dynamics of all observables and hidden variables in parallel. Thus, a high-dimensional state transition matrix A is required. For numerical stability we recommend to initialize matrix A with a (random) sparse connectivity. The degree of sparsity is chosen with respect to the metaparameters connectivity and memory length (for details see [ZGST06]).

If we repeat the system identification we will get an ensemble of solutions. All solutions have a model error of zero in the past, but show a different behavior in the future. The reason for this lies in different ways of reconstructing the hidden variables from the observations and is independent of different random sparse initializations. Since every model gives a perfect description of the observed data, we can use the simple average of the individual forecasts as the expected value, assuming that the distribution of the ensemble is unimodal.

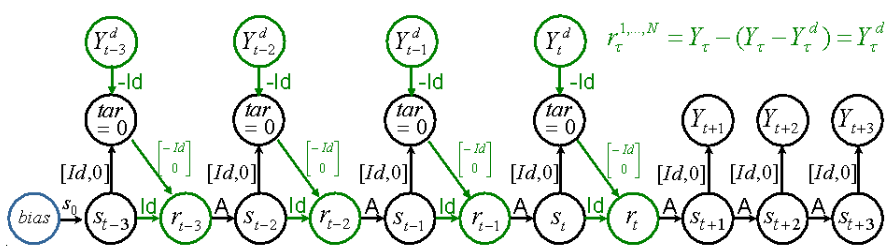

Fig. 3 HCNN incorporating a Teacher Forcing (TF) mechanism

3 Risk Analysis in Financial Markets

The latest financial crisis has triggered a far-reaching discussion on the limitations of quantitative forecasting models and made investors very conscious of risk[Foel09]. Risk management frequently considers the probability distribution of market prices / returns[Hull01]. In order to understand risk distributions, traditional risk management uses historical simulations which require strong model assumptions. Risk is understood as a random walk, in which the diffusion process is calibrated by the observed past error of the underlying model[MFE05].

For our approach this concept fails, because the (past) residual error of the HCNN is zero. Our risk concept is based on the partial observability of the world, leading to different reconstructions of the hidden variables and thus, different future scenarios. Since all scenarios are perfectly consistent with the history, we do not know which of the scenarios describes the future trend best and risk emerges.

Our approach directly addresses the model risk. For HCNN modeling we claim that the model risk is equal to the forecast risk. The reasons can be summarized as follows: First, HCNNs are universal approximators, which are therefore able to describe every future market scenario. Second, the form of the ensemble distribution is caused by underlying dynamical equations, which interpret the market dynamics as the result of interacting decisions[Zim94]. Third, in experiments we have shown that the ensemble distribution is independent from the details of the model configuration, iff we use large models and large ensembles.

The diagram below (Fig. 4, left) shows our approach applied to the Dow Jones Industrial Index (DJX). For the ensemble, a HCNN was used to generate 250 individual forecasts for the DJX. For every forecast date, all of the individual forecasts for the ensemble represent the empirical density function, i. e. a probability distribution over many possible market prices at a single point in time (see Fig. 4, right). It is noticeable that the actual development of the DJX is always within the ensemble channel (see grey lines, Fig. 4, left). The expected value for the forecast distribution is also an adequate point forecast for the DJX (see Fig. 4, right).

It is our intention to use the ensemble distribution of the HCNN in a future project for the evaluation of call and put options.

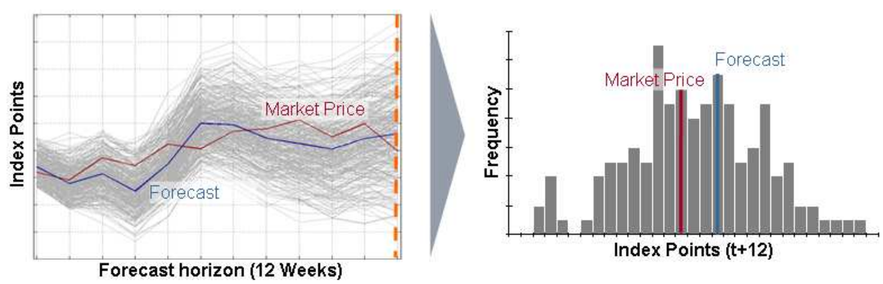

Fig. 4 HCNN ensemble forecast for the Dow Jones Index (12 week forecast horizon), left, and associated index point distribution for the ensemble in forecast time step $t+12$ weeks, right.

4 Conclusion and Outlook

The joint modeling of hidden and observed variables in large recurrent neural networks provides new prospects for planning and risk management. The ensemble approach based on HCNN offers an alternative approach to forecasting of future probability distributions. HCNNs give a perfect description of the dynamic of the observables in the past. However, the partial observability of the world results in a non-unique reconstruction of the hidden variables and thus, different future scenarios. Since the genuine development of the dynamic is unknown and all paths have the same probability, the average of the ensemble may be regarded as the best forecast, whereas the bandwidth of the distribution describes the market risk.

Today, we use HCNN forecasts to predict prices for energy and precious metals to optimize the timing of procurement decisions. Work currently in progress concerns the analysis of the properties of the ensemble and the implementation of these concepts in practical risk management and financial market applications.

All NN architectures and algorithms are implemented in the Simulation Environment for Neural Networks (SENN), a product of Siemens Corporate Technology.

References

[CK01] Calvert D. and Kremer St.: Networks with Adaptive State Transitions, in: Kolen J. F. and Kremer, St. (Ed.): A Field Guide to Dynamical Recurrent Networks, IEEE, 2001, pp. 15-25.

[Foel09] Föllmer, H.: Alles richtig und trotzdem falsch?, Anmerkungen zur Finanzkrise und Finanzmathematik, in: MDMV 17/2009, pp. 148-154

[Hay08] Haykin S.: Neural Networks and Learning Machines, 3rd Edition, Prentice Hall, 2008.

[Hull01] Hull, J.: Options, Futures & Other Derivative Securities. Prentice Hall, 2001.

[MFE05] McNeil, A., Frey, R. and Embrechts, P.: Quantitative Risk Management: Concepts, Techniques and Tools, Princeton University Press, Princeton, New Jersey, 2005.

[Pearl01] Pearlmatter B.: Gradient Calculations for Dynamic Recurrent Neural Networks, in: A Field Guide to Dynamical Recurrent Networks, Kolen, J.F.; Kremer, St. (Ed.); IEEE Press, 2001, pp. 179-206.

[SZ06] Schäfer, A. M. und Zimmermann, H.-G.: Recurrent Neural Networks Are Universal Approximators. ICANN, Vol. 1., 2006, pp. 632-640.

[Wei90] Wei W. S.: Time Series Analysis: Univariate and Multivariate Methods, Addison-Wesley Publishing Company, N.Y., 1990.

[Wer74] Werbos P. J.: Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences, PhD Thesis, Harvard University, 1974.

[WZ89] Williams R. J. and Zipser, D.: A Learning Algorithm for continually running fully recurrent neural networks, Neural Computation, Vol. 1, No. 2, 1989, pp. 270-280.

[ZGN02] Zimmermann, H. G., Grothmann, R. and Neuneier, R.: Modeling of Dynamical Systems by Error Correction Neural Networks. In: Soofi, A. und Cao, L. (Ed.): Modeling and Forecasting Financial Data, Techniques of Nonlinear Dynamics, Kluwer, 2002.

[ZGST06] Zimmermann, H. G., Grothmann, R., Schäfer, A. M. and Tietz, Ch.: Modeling Large Dynamical Systems with Dynamical Consistent Neural Networks, in New Directions in Statistical Signal Processing: From systems to brain. Haykin, S., Principe, J. C., Sejnowski, T. J., and McWhirter, J. (Ed.), MIT Press, Cambridge, Mass., 2006.

[Zim94] Zimmermann, H. G.: Neuronale Netze als Entscheidungskalkül. In: Rehkugler, H. und Zimmermann, H. G. (Ed.): Neuronale Netze in der Ökonomie, Grundlagen und wissenschaftliche Anwendungen, Vahlen, Munich 1994.

View publication stats