ESANN 2017 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 26-28 April 2017, i6doc.com publ., ISBN 978-287587039-1. Available from http://www.i6doc.com/en/.

Predicting Time Series with Space-Time Convolutional and Recurrent Neural Networks

Wolfgang Groß[1,2], Sascha Lange[1], Joschka Bödecker[3] and Manuel Blum[1,3] *

1- PSIORI - Freiburg im Breisgau - Germany

2- University of Osnabrück - Institute of Cognitive Science - Germany

3- University of Freiburg - Machine Learning Lab - Germany

Abstract. Convolutional neural networks (CNNs) with their ability to learn useful spatial features have revolutionized computer vision. The network topology of CNNs exploits the spatial relationship among the pixels in an image and this is one of the reasons for their success. In other domains deep learning has been less successful because it is not clear how the structure of non-spatial data can constrain network topology. Here, we show how multivariate time series can be interpreted as space-time pictures, thus expanding the applicability of the tricks-of-the-trade for CNNs to this important domain. We demonstrate that our model beats more traditional state-of-the-art models at predicting price development on the European Power Exchange (EPEX). Furthermore, we find that the features discovered by CNNs on raw data beat the features that were hand-designed by an expert.

1 Introduction

The recent success of deep learning methods [1] is mainly due to the ability of deep networks to discover useful features in high-dimensional data [2, 3]. Especially in the field of computer vision, convolutional neural networks (CNNs) are well known to learn relevant features that radically improved the performance of modern computer vision systems. In fact, CNNs are now considered the gold standard for many computer vision tasks.

The success in this specific domain is due to two properties of image data that are known a priori: there are (a) high correlations between dimensions (pixels) and (b) the correlations are spatially organized, i.e. pixels closer to each other are more highly correlated. These properties are reflected in the topology of deep neural networks: local processing in the first layers and quick reduction of redundancy, the fundamental ideas behind convolutional neural networks (CNN) and also of autoencoders.

These principles have also been applied successfully to some none-image domains [4, 5, 6]. However, it is an open question how to transfer them to domains where the "spatial" relationship between dimensions is less obvious. For time series data one such relation is obvious, namely the temporal correlation among measurements. Previous work has exploited this relation with the traditional

*This research was partly funded by the Addison Fischer Institute for Research and Ethical Use of Artificial Intelligence. The authors would like to thank Frank Jäkel for helpful comments and editing.

ESANN 2017 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 26-28 April 2017, i6doc.com publ., ISBN 978-287587039-1.
Available from http://www.i6doc.com/en/.

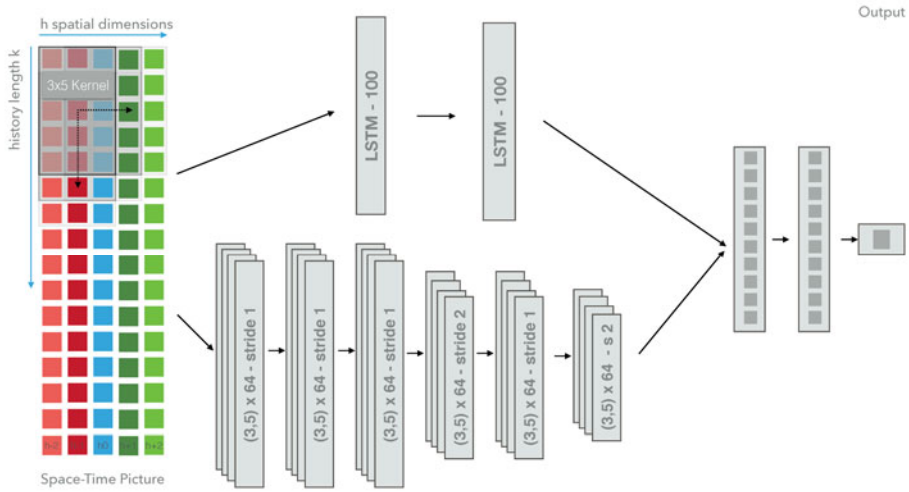

Fig. 1: The Space-Time Convolutional and Recurrent Neural Network (STaR) architecture. Related time series are arranged in a space-time picture and fed to the input layer of STaR.

sliding-window approach, recurrent neural networks [4, 7], and temporal receptive fields [8, 9]. However, as real-world time series often have very low signal-to-noise ratio, improvement over traditional shallow models has been limited.

Our key novel idea in this paper is to extract additional information from related time series by treating multivariate time series as a space-time picture. This space-time picture and our model are depicted in Fig. 1. Discrete time flows from top to bottom and the different channels of the time series are arranged from left to right. As usual, the input neurons provide a sliding window for a history of k values of the time-series. Importantly, the receptive fields range over time *and* space (channels). Note that it is necessary to be able to arrange channels along a "spatial" dimension for the spatio-temporal receptive fields to be meaningful, i.e. neighbouring channels should be related to each other.

This is, for example, naturally the case for prices at the European Power Exchange (EPEX). Electricity is traded per hour, e.g. you can buy or sell contracts that promise to provide electricty for tomorrow 2pm, 3pm, or 4pm. Market prices for these three hourly contracts change over time depending on predicted energy production and consumption. Hence, in this example there are three time-series (one for the price development of each contract) with a one-dimensional "spatial" structure of the 2pm contract being closer to the 3pm contract than the 4pm contract.

In the experimental section we demonstrate that for EPEX price predictions our space-time convolutional neural network outperforms non-convolutional methods and networks that only exploit temporal correlations. In addition, we present a new space-time convolutional and recurrent (STaR) network architecture that

ESANN 2017 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 26-28 April 2017, i6doc.com publ., ISBN 978-287587039-1.
Available from http://www.i6doc.com/en/.

improves the performance even further. We show that this network learns useful features across space and time and we show that these features are better than those designed by an expert.

2 Model

Along this line of argument, we present and later evaluate three deep architectures exploting the space-time picture. We show how different components—(1) space-time convolutions, (2) recurrency on the time component alone, and (3) a combination of both—contribute to the performance.

Space-Time convolutional neural network (ST-CNN). The space-time convolutional neural network (ST-CNN) is a short-term feature detector on the space-time picture. Our CNN follows the suggestions of [10] for the image domain, with some trivial parameter adaptations made to the time series domain. Every convolutional layer is followed by a ReLU nonlinearity and a dropout layer. The network is trained with the ADAM back-propagation algorithm on mini batches. The size of the input is 100 in history length (k) and 3, 5 or 7 (h) in the spatial dimension.

Recurrent neural network (RNN). We use long short-term memory cells (LSTM) [11] in the recurrent neural network. LSTM cells use gates to improve the training of the model. With their recurrent connections LSTMs have the potential of learning better long-term features than is possible with size-limited convolutional kernels. Following the recurrent layers there are two dense "classification" layers (as in ST-CNN) with 100 units each, ReLU nonlinearities, and a dropout between the layers as the ST-CNN. The LSTM layers have 100 LSTM units and a hyperbolic tangent nonlinearity. The network is also trained with the ADAM back propagation algorithm on the same input.

Space-Time convolutional and Recurrent Neural Network (STaR). The novel STaR architecture combines the ST-CNN and the RNN with LSTM cells from above into one hybrid model by copying the input and concatenating the networks' output. Figure 1 shows the architecture of STaR. This architecture choice loosely follows the idea for parallel feature detectors as presented in [12, 13]. In these earlier works, copies of the same CNN are used to detect features of different inputs, and are concatenated in the end. In contrast, we use different feature detectors on the same input to learn different feature representations.

Benchmark models. In addition to the three deep architectures for time series prediction, we used several benchmark models for comparison. These models have hand-engineered features in order to compare these features to the features learned by the deep architectures. The models that are used are as follows: a shallow neural network, a random forest, and a linear model.

ESANN 2017 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 26-28 April 2017, i6doc.com publ., ISBN 978-287587039-1.
Available from http://www.i6doc.com/en/.

3 Experiments

The European power exchange (EPEX) is a market place to trade power (MWh) in the form of contracts. A contract is an hourly liability to deliver or distribute power into the network. Hence, there are 24 unique related contracts each day, one for every hour. As we know that successive hours (i.e. neighboring contracts) have similar price developments—because of similar supply and demand over time—we model their relationship as the "spatial" dimension and learn combined features.

Baseline with human feature engineering. We benchmark all models against traditional linear and non-linear machine learning methods with features engineered by a human. These features have been provided by a domain expert with years of trading experience. These expert features even make use of additional information from the order books and include features like the price, its movement, spread among ask and bid side, spread to prices of different contracts, and the order book depth.

Experimental setting. In a trading setting the direction of movement is arguably of greater value than an exact price prediction and also easier to evaluate and compare. Therefore we define the problem as a classification problem with the goal to predict the direction of price at $\Delta t = 15$ minutes every 20 seconds. We report accuracy on three target classes: price stays the same in a margin of $\pm$ 0.40 €, price rises over this margin, or the price falls below this margin. The three classes are roughly equally distributed in the data set. We use 10-fold cross validation to generate training and test set and report the mean test score over all folds.

3.1 Experiments with feature engineering

In order to establish a challenging baseline we tested several shallow models on the expert features (see Table 1).

NN shallow	**Linear Model**	**Random Forest**	**Expert Model**
46.4 %	45.7 %	37.5 %	42.3 %

Table 1: Baseline Model Comparison. NN-shallow is a neural network with only two hidden layers. Expert is a manually-tuned decision tree built by a domain expert.

Chance level is roughly one third and all models perform better than that. The expert's model is a form of decision tree with carefully hand-selected features and boundaries. Out of the approximately 70 available hand-crafted features the expert model only uses a small subset (less than 10). The other models only used a similar small subset of the features that was optimized for each method. This

ESANN 2017 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 26-28 April 2017, i6doc.com publ., ISBN 978-287587039-1.
Available from http://www.i6doc.com/en/.

shows that there is a good deal of useful information in the features. If we can learn features as good as those of the expert, we can use this same shallow NN to make predictions. If the predictive performance is better than the performance of the shallow NN with expert features, we can conclude that deep models learn better feature representations.

3.2 Deep models without feature engineering

In the following experiment only the raw dataset is used as input and none of the expert features were used. We compare the deep models introduced in section 2 against each other with variations in the network architecture in order to show which parts of the architecture are beneficial for feature learning.

Time CNN	**ST-CNN 6L-3Ch**	**ST-CNN 6L-5Ch**	**ST-CNN 9L-5Ch**	**ST-CNN 6L-7Ch**
38.6 %	41.9 %	41.8 %	43.9 %	44.3 %
NN	**RNN 1L**	**RNN 2L**	**STaR NN**	**STaR Linear**
42.1 %	45.3 %	44.3 %	**48.3 %**	43.2 %

Table 2: Model comparison on the raw dataset without feature engineering. The table shows the results of the benchmark dense neural network and several versions of the ST-CNN are compared with different number of layers (L) and different number of channels (Ch) in the input data. Also, for the RNN two results are reported; with one and with two LSTM layers. Time CNN is the standard approach with convolution only over one time dimension.

By comparing these results to Table 1 we can conclude that the learned features are indeed useful for the classification task. The results also show that both the ST-CNN and RNN architectures are better than the vanilla dense NN. Note that combining these approaches improves the overall performance. It is known that CNNs and RNNs have different biases [14]. Using both in an ensemble and combining them may thus explain the improved performance. The STaR network shows the overall best performance and is therefore able to learn features that are better than the expert features.

As the network working on the single time series produces the worst results (Time CNN), we can also conclude that using neighboring channels improves the predictive power of a ST-CNN (3Ch, 5Ch, 7Ch), hence backing the idea of the space-time picture as a useful representation of multivariate time series.

4 Conclusion and outlook

We presented a novel approach for classifying time series with deep neural networks. Our approach makes use of the information contained in the "spatial" relations between time series. The STaR network architecture uses both CNNs

ESANN 2017 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 26-28 April 2017, i6doc.com publ., ISBN 978-287587039-1. Available from http://www.i6doc.com/en/.

and RNNs and has been trained end-to-end. The network learns feature representations of at least the same quality as those provided by a domain expert. We conclude that insights from applying deep learning in computer vision domain can also be applied fruitfully to time series prediction problems. As deep neural networks are exceptionally good at finding complex features in high-dimensional data, better than state-of-the-art performance can be achieved without elaborate feature engineering. We have already applied this method to multivariate prediction problems in the domain of financial time series and machine data in the automotive sector. In these cases, the spatial relationship among channels is less obvious but a suitable ordering can be constructed from the data or expert insight.

References

[1] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller. Playing atari with deep reinforcement learning. *CoRR*, abs/1312.5602, 2013.

[2] G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313:504 – 507, 2006.

[3] S. Lange and M. Riedmiller. Deep auto-encoder neural networks in reinforcement learning. In *The 2010 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, July 2010.

[4] H. Sak, A. W. Senior, and F. Beaufays. Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition. *CoRR*, abs/1402.1128, 2014.

[5] M. Watter, J.T. Springenberg, J. Boedecker, and M. Riedmiller. Embed to control: A locally linear latent dynamics model for control from raw images. *CoRR*, abs/1506.07365, 2015.

[6] N. Chen, J. Bayer, S. Urban, and P. van der Smagt. Efficient movement representation by embedding dynamic movement primitives in deep autoencoders. In *Proc. 2015 IEEE-RAS International Conference on Humanoid Robots*, 2015.

[7] J. Boedecker, O. Obst, J.T. Lizier, N. M. Mayer, and M. Asada. Information processing in echo state networks at the edge of chaos. *Theory in Biosciences*, 131(3):205–213, 2012.

[8] Y. Zheng, Q. Liu, E. Chen, Y. Ge, and J. L. Zhao. *Web-Age Information Management: 15th International Conference, WAIM 2014, Macau, China, June 16-18, 2014. Proceedings*, chapter Time Series Classification Using Multi-Channels Deep Convolutional Neural Networks, pages 298–310. Springer International Publishing, Cham, 2014.

[9] Z. Cui, W. Chen, and Y. Chen. Multi-Scale Convolutional Neural Networks for Time Series Classification. *CoRR*, abs/1603.06995, March 2016.

[10] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller. Striving for simplicity: The all convolutional net. *CoRR*, abs/1412.6806, 2014.

[11] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, November 1997.

[12] S. E. Reed, Y. Zhang, Y. Zhang, and H. Lee. Deep visual analogy-making. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 1252–1260. Curran Associates, Inc., 2015.

[13] M. Oberweger, P. Wohlhart, and V. Lepetit. Training a feedback loop for hand pose estimation. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 3316–3324, Dec 2015.

[14] K. J. Geras, A. Mohamed, R. Caruana, G. Urban, S. Wang, Ö. Aslan, M. Philipose, M. Richardson, and C. A. Sutton. Blending lstms into cnns. *CoRR*, abs/1511.06433, 2016.