ESANN 2020 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 2-4 October 2020, i6doc.com publ., ISBN 978-2-87587-074-2.
Available from http://www.i6doc.com/en/.

A Distributed Neural Network Architecture for Robust Non-Linear Spatio-Temporal Prediction

Matthias Karlbauer[1]*, Sebastian Otte[1],
Hendrik P.A. Lensch[2], Thomas Scholten[3], Volker Wulfmeyer[4], and Martin V. Butz[1]

1- University of Tübingen - Neuro-Cognitive Modeling Group
Sand 14, 72076 Tübingen - Germany

2- University of Tübingen - Computer Graphics
Maria-von-Linden-Straße 6, 72076 Tübingen - Germany

3- University of Tübingen - Soil Science and Geomorphology
Rümelinstraße 19-23, 72070 Tübingen - Germany

4- University of Hohenheim - Institute for Physics and Meteorology
Garbenstraße 30, 70599 Stuttgart - Germany

Abstract.
DISTANA – a distributed spatio-temporal artificial neural network architecture – learns to model and predict spatio-temporal time series dynamics. It learns in a parallel, spatially distributed manner while employing a mesh of recurrent, neural prediction kernels (PKs). Individual PKs predict the local data stream and exchange information laterally. DISTANA essentially assumes that generally applicable causes, which may be locally modified, generate the observed data. We show that DISTANA scales and generalizes to large problem spaces, can approximate complex dynamics, and is robust to overfitting, outperforming other competitive ANNs.

1 Introduction

Modeling and predicting non-linear, spatio-temporal dynamics is challenging for current pattern recognition systems [1]. Representative dynamics include, for example, brain activities [2], video streams [3], traffic flow [4], and weather and climate progressions [5, 6]. The major challenge is to infer, model, and predict the *underlying causes* that generate the perceived data stream. A key property, which all spatio-temporal processes have in common, is that the same underlying causal principles—such as physics when observing natural processes—apply irrespective of time or location. As a result, similar dynamics will be observable repeatedly at different spatial locations and points in time.

DISTANA actively searches for these underlying causes in spatially distributed time series data. It learns a *predictive*, spatio-temporal, neural network *kernel* (PK), which is applied to all nodes of a mesh. Thus, all nodes apply the same operations at different locations. This enables efficient computation in and learning from all nodes in parallel. Moreover, it predisposes DISTANA to

*This work received funding from the German Research Foundation (DFG) under Germany's Excellence Strategy – EXC-Number 2064/1 – Project Number 390727645. Moreover, we thank the International Max Planck Research School for Intelligent Systems (IMPRS-IS) for supporting Matthias Karlbauer.

ESANN 2020 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 2-4 October 2020, i6doc.com publ., ISBN 978-2-87587-074-2. Available from http://www.i6doc.com/en/.

identify the universal, recurring causes of the observed pattern dynamics. Compared to seven other ANN models, including (temporal) convolutional neural networks (CNNs, TCNs), recurrent neural networks (RNNs), and combinations of both (e.g. ConvLSTM), DISTANA reaches both higher accuracy and robustness at approximating circularly propagating waves. Moreover, it is critically less prone to overfitting and bears the potential to handle heterogeneously distributed sensor meshes. In the near future we will apply DISTANA to related, but more challenging real-world problems, such as modeling the partially chaotic processes that generate our weather and climate.

2 DISTANA

While CNNs can efficiently and accurately process spatially distributed information such as images, RNNs—and long short-term memory cells (LSTMs) [7] in particular—are designed to handle time series data. Recently, Shi et al. [6] proposed ConvLSTM—a convolution-gating architecture, which combines CNNs and LSTMs, thus processing spatial and temporal information simultaneously. GridLSTM [8], on the other hand, extends LSTMs to process not only temporal but also spatial data dimensions sequentially.

DISTANA belongs to a third related class of architectures, which is referred to as graph neural networks (GNNs) [9]. GNNs treat graph vertices and edges in two different neural network components. Unlike earlier GNNs, however, DISTANA integrates LSTM structures, projects the graph, i.e. its mesh, onto a metrical space, and assumes universal causes underlying the observable spatio-temporal data.

DISTANA consists of a PK network, which generates dynamic predictions at each desired spatial location. Multiple PK instances, which share their respective weights, are applied in a sensor mesh, enabling their parallel invocation. Each PK instance receives (1) dynamic input, which is subject to prediction and changes over time, (2) static information, which stays constant and characterizes the location of each PK, and (3) lateral input from neighboring PKs. Typically a PK contains recurrent connections.

3 Experiments

In two experiments, which differ in the data sets used, several ANN architectures including fully connected networks, CNNs, and RNNs are compared with DISTANA. We model a wave-like spatio-temporal process (cf. Figure 1) distributed in a 16×16 mesh. Train and test errors are mean squared errors between network output and target values, which are the dynamic inputs (e.g. wave height) at the next time step. The test error is calculated over 65 time steps of closed loop performance, where the network feeds itself with its own dynamic predictions from the previous time step. The closed loop begins after 15 steps of teacher forcing, which ground the recurrent activity in the network.

ESANN 2020 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 2-4 October 2020, i6doc.com publ., ISBN 978-2-87587-074-2. Available from http://www.i6doc.com/en/.

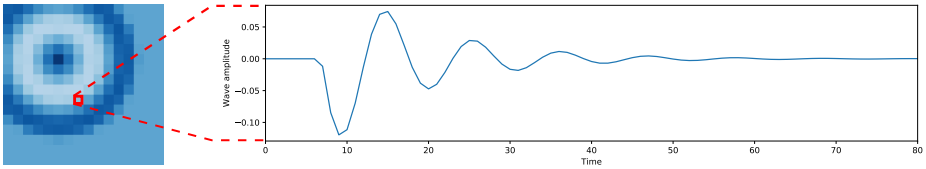

Fig. 1: Data set one. Left: exemplary circular wave. Right: activity pattern over time at one particular position in the two-dimensional wave field.

3.1 Data Set 1

Single sinusoidal waves are generated propagating outwards:

$$u(x,y,t) = \begin{cases} \sin(r_{x,y} - ct)\exp(-d(ct - r_{x,y})) & \text{if } r_{x,y} < ct \\ 0 & \text{else} \end{cases}, \qquad (1)$$

where $u(x,y,t)$ is the wave height of the field at a certain position and time, $\sin(r_{x,y} - ct)$ defines the oscillating wave height considering the distance to the wave center $r_{x,y}$ and the current time step t, and $\exp(-d(ct - r_{x,y}))$ causes waves to decay away from the wave origin over time (decay factor $d = 0.25$). Constant $c = 10$ is the wave velocity. Field values that have not been reached by the wave, yet, are set to zero. The waves are not reflected at the borders.

Table 1 shows the performance of all compared models at approximating these circular wave dynamics. Besides *train* and *test errors*, we report the number of *parameters* and the *inference time* of one sequence (consisting of 80 time steps) for each model. In order to rigorously test all models for their generalization abilities, we also trained them on one single sequence (one wave origin) and computed the test error on unseen sequences (test error *1-train-ex.*). Furthermore, to elaborate the models' abilities to approximate variable dynamics, we trained them on waves that travel with varying velocities (test error *var. wave*). *Spatial scalability*, reported in the model descriptions below, indicates whether a model can be directly applied to input fields of different resolutions.

Performances of the following models are compared:
Baselines: *Baseline* $t-1$ is the identity function; *Baseline zero* predicts zeros.
Fully Connected Networks: A naive and spatially not scalable approach to model the circular wave is a fully connected linear network (*FC-Linear*), with $16 \times 16 = 256$ cells, receiving the flattened input. A more elaborated model is *FC-LSTM*, which replaces the linear layer of *FC-Linear* by a 256-cell LSTM layer to facilitate temporal information processing.
CNN: To reduce the number of parameters, defining a spatially scalable model, numerous *CNN*s with different kernel sizes, a varying number of feature maps, and two convolutional layers were evaluated. The best results, which are reported here, were achieved by using a kernel size of 3×3 and one feature map.
Temporal Convolution Network: TCNs, as a spatially scalable approach, were applied with three 3D convolution layers, each with a $3 \times 3 \times 3$ kernel and $[1, 8, 1]$ feature maps. Other settings did not seem to improve performance.

ESANN 2020 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 2-4 October 2020, i6doc.com publ., ISBN 978-2-87587-074-2. Available from http://www.i6doc.com/en/.

Table 1: Performance measures of simple wave propagations on data set one.

Model (#pars)	Train error	Test error	Inf. time	1-train-ex.	Var. wave
Baseline $t-1$	-	3.59×10^{-5}	-	3.59×10^{-5}	5.90×10^{-5}
Baseline zero	-	8.88×10^{-5}	-	8.88×10^{-5}	5.84×10^{-4}
FC-Linear (65k)	2.36×10^{-4}	2.56×10^{-4}	**0.0051 s**	2.41×10^{-4}	1.91×10^{-3}
FC-LSTM (524k)	5.34×10^{-5}	1.38×10^{-2}	0.0113 s	2.14×10^{-3}	6.57×10^{-3}
CNN (20)	3.41×10^{-4}	2.22×10^{-4}	0.0115 s	1.37×10^{-3}	2.66×10^{-2}
TCN (2.3k)	1.17×10^{-5}	8.56×10^{-3}	0.0531 s	1.04×10^{-1}	3.09×10^{-2}
CLSTMC (768k)	6.28×10^{-5}	4.67×10^{-1}	0.0230 s	6.13×10^{-4}	2.71×10^{-1}
ConvLSTM1 (144)	1.83×10^{-5}	4.26×10^{-5}	0.0247 s	4.55×10^{-5}	5.85×10^{-4}
ConvLSTM8 (2.9k)	6.34×10^{-6}	$\mathbf{1.28 \times 10^{-6}}$	0.0298 s	1.29×10^{-2}	7.88×10^{-4}
GridLSTM (624)	7.95×10^{-5}	3.62×10^{-1}	5.8786 s	2.86×10^{-1}	1.35×10^{-1}
BiGridLSTM (1.8k)	$\mathbf{6.28 \times 10^{-6}}$	5.65×10^{-1}	11.9900 s	8.67×10^{-1}	4.55×10^{-2}
DISTANA4 (108)	4.18×10^{-5}	2.08×10^{-5}	0.0264 s	1.41×10^{-5}	2.17×10^{-4}
DISTANA26 (2.9k)	2.58×10^{-5}	1.48×10^{-5}	0.0326 s	$\mathbf{2.04 \times 10^{-5}}$	$\mathbf{9.99 \times 10^{-5}}$

CNN-LSTM-CNN (CLSTMC): *CNNs* were extended by inserting a fully connected LSTM layer—making it not spatially scalable—after a variable number of layers. Best results were achieved with one 3×3 convolution followed by a flat LSTM layer and a 3×3 transposed convolution with skip connection.
ConvLSTM: Two models of the spatially scalable ConvLSTM architecture, both with two layers and kernel size three, are reported: *ConvLSTM1* with one feature map in both layers, and *ConvLSTM8* with eight feature maps in the first layer, which are reduced to one in the second layer.
GridLSTM and BiGridLSTM: *GridLSTM* runs forward in time and space; *BiGridLSTM* processes data forward in time but bidirectionally over space. Both are spatially scalable.
DISTANA: DISTANA is spatially scalable. PKs consist of a two-neuron tanh layer, followed by a layer of either four or 26 LSTM cells and another two-neuron tanh layer. This yields, for example, $108 = (2 \cdot 2) + (2 \cdot 4 \cdot 4 + 4 \cdot 4 \cdot 4) + (4 \cdot 2)$ parameters for DISTANA4.

3.2 Data Set 2

To increase data complexity, a second set was created where waves are reflected at borders, such that wave fronts become interactive. We focus our analysis on the most promising architectures determined above. For wave data generation, the two-dimensional wave equation

$$\frac{\partial^2 u}{\partial t^2} = c^2 \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) \tag{2}$$

was solved using the second order central differences approach to obtain an equation for computing the state of the field at a desired position (x, y) in the subsequent time step $t + \Delta_t$

$$u(x, y, t + \Delta_t) = c^2 \Delta_t^2 (u_{xx} + u_{yy}) + 2u(x, y, t) - u(x, y, t - \Delta_t). \tag{3}$$

ESANN 2020 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 2-4 October 2020, i6doc.com publ., ISBN 978-2-87587-074-2. Available from http://www.i6doc.com/en/.

Table 2: Same evaluation as in Table 1 on data set two, including TCN, ConvLSTM and three variants of DISTANA.

Model (#pars)	Train error	Test error	Inf. time
Baseline $t-1$	-	5.83×10^{-3}	-
Baseline zero	-	1.07×10^{-2}	-
TCN (2.3k)	1.14×10^{-5}	2.11×10^{-1}	0.0707 s
ConvLSTM8 (2.9k)	3.52×10^{-6}	8.09×10^{-2}	0.0289 s
DISTANAv1 (146)	7.89×10^{-6}	8.77×10^{-3}	**0.0280 s**
DISTANAv2 (172)	$\mathbf{1.37 \times 10^{-6}}$	7.68×10^{-4}	0.0294 s
DISTANAv3 (200)	1.64×10^{-6}	$\mathbf{4.99 \times 10^{-4}}$	0.0301 s

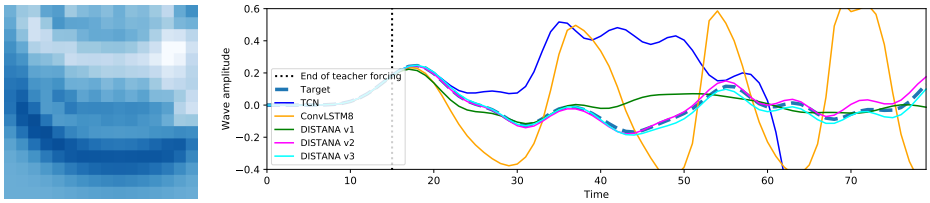

Fig. 2: Data set two. Left: exemplary circular wave with reflecting borders. Right: network dynamics generated by selected architectures.

The unfolding dynamics of higher complexity are much harder to predict (cf. Figure 2). None of the previously tested architectures was able to approximate the dynamics satisfactorily (cf. Table 2) yielding errors larger than the baselines. Accordingly, DISTANA was enhanced as follows:

DISTANA v1: The size of the preprocessing feed forward layer in the PK was increased from two to four neurons.

DISTANA v2: Enhances DISTANA v1 with eight, compared to one, lateral input neurons, which receive input from the eight neighboring PKs, respectively.

DISTANA v3: Enhances DISTANA v2 increasing the number of lateral output neurons from one to eight, dynamically routing individualized outputs to the respective neighbors.

DISTANAv2 and DISTANAv3 strongly outperform the simpler DISTANA versions as well as TCN and ConvLSTM. Table 2 shows that DISTANAv2 reaches the lowest training error, while DISTANAv3 yields the best generalization performance. Fig. 2 shows that when closed loop predictions unfold after 15 steps of teacher forcing, DISTANAv2 and DISTANAv3 approximate the target value still similarly well while the other ANN architectures start to strongly deviate from the target values after only five to ten closed-loop prediction steps. Online video material[1] illustratively shows the further abilities of DISTANA, including its ability to generalize to larger grid sizes.

[1] `https://youtu.be/dH8qcBVuwFg`

ESANN 2020 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 2-4 October 2020, i6doc.com publ., ISBN 978-2-87587-074-2.
Available from http://www.i6doc.com/en/.

4 Discussion

Several ANN architectures were compared at approximating spatio-temporal processes. In the simple scenario, only ConvLSTM and our model, DISTANA, yield smaller test errors than the two baselines when closed loop performance over T prediction time steps is considered. This requires both intrinsic model stability and the maintenance of plausible ongoing dynamics. While the reported test error is in favor of ConvLSTM, DISTANA proved robust to few and variable training data, even with a network that contains only 108 parameters. These findings were corroborated by the evaluations in a second, more complex data set, in which waves were reflected at borders and thus heavily interacted with each other. All other considered architectures failed to generate lasting closed-loop predictions, except for two variants of DISTANA, which consider lateral information propagation explicitly (Figure 2).

Here we have considered regularly distributed grids. However, ongoing work shows that DISTANA can indeed handle irregularly distributed sensor meshes when introducing transition kernels. We thus expect to be able to scale to predict heterogeneously distributed spatiotemporal data, as, for example, necessary to generate highly accurate and further reaching short-range weather forecasts.

References

[1] Michael M Bronstein, Joan Bruna, Yann LeCun, Arthur Szlam, and Pierre Vandergheynst. Geometric deep learning: going beyond euclidean data. *IEEE Signal Processing Magazine*, 34(4):18–42, 2017.

[2] Nikola K Kasabov. Neucube: A spiking neural network architecture for mapping, learning and understanding of spatio-temporal brain data. *Neural Networks*, 52:62–76, 2014.

[3] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1725–1732, 2014.

[4] Zheng Zhao, Weihai Chen, Xingming Wu, Peter CY Chen, and Jingmeng Liu. Lstm network: a deep learning approach for short-term traffic forecast. *IET Intelligent Transport Systems*, 11(2):68–75, 2017.

[5] J. N. K. Liu, Y. Hu, Y. He, P. W. Chan, and L. Lai. *Information Granularity, Big Data, and Computational Intelligence*, volume 8 of *Studies in Big Data*, chapter Deep Neural Network Modeling for Big Data Weather Forecasting, pages 389–408. Springer, 2015.

[6] Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-kin Wong, and Wang-chun WOO. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 802–810. Curran Associates, Inc., 2015.

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

[8] Nal Kalchbrenner, Ivo Danihelka, and Alex Graves. Grid long short-term memory. *arXiv preprint arXiv:1507.01526*, 2015.

[9] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80, 2008.