Learning beyond simulated physics

Alexis Asseman **Tomasz Kornuta** **Ahmet S. Ozcan**
IBM Research AI, Almaden Research Center, San Jose, USA
alexis.asseman@ibm.com, {tkornut, asozcan}@us.ibm.com

Abstract

Most of recent advancements in spatio-temporal predictions are based on simulated physics. In this paper we introduce a new dataset based on videos of a double pendulum, captured with a high-speed camera, supplemented with positions of its datums and angles between its arms. Because the recorded trajectories depend on unknown initial conditions, the dataset constitutes a benchmark for chaotic behaviors that can be present in other real-world problems. As the description of the system state is extremely simple, the dataset enables careful testing and analysis of the behavior of the developed model. We provide results of stacked LSTM operating directly on angles between arms as a baseline for future research.

1 Introduction

Chaotic time series were always in the scope of interest of research on forecasting [5]. Extensively studied prediction tasks include the Mackey-Glass time series [14], chaotic laser data from the Santa Fe Institute contest [19] and Lorentz [13] and Rossler [16] attractors. All of these mentioned problems operate on numerical (tabular) data (please refer to attributes comparison of datasets from [20]).

The recent progress in artificial intelligence shifted the research direction towards long-term, spatio-temporal predictions [4], where the input images are processed by the model with the goal of predicting the future trajectories of objects. Those works mostly fall into the intuitive physics domain [9], where images reflecting object behavior are generated by simulators, typically being variations of *falling object*, *object collision* or *n-body* problems. For example, [1] used three such environments (n-body, bouncing balls and strings colliding with rigid objects), [18] operated on five physical domains (namely: spring, gravity, billiards, magnetic billiards and drift). Some authors combined simulation with real data, e.g. [22] used two domains (billiard and block towers), and first trained the system on synthetic billiard videos from [6] and used two-second *YouTube* billiard clips for validation. The block towers used sequences of real images of falling block towers from [10].

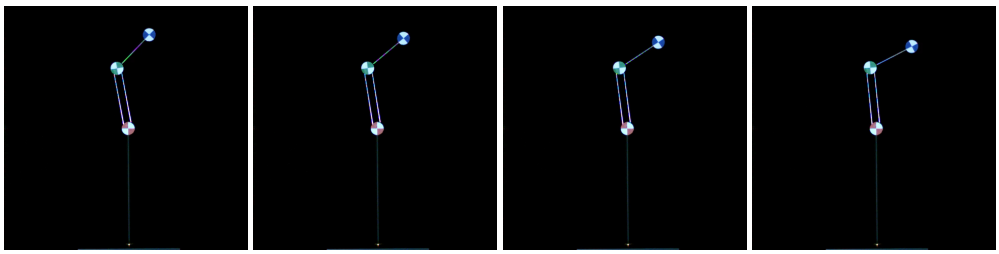


Figure 1: Four consecutive frames from one of the *Double Pendulum Chaotic Dataset* sequences

Those results indicate that the community lacks benchmarks based on real-world physical systems that will enable to test the robustness of predictive models along with the associated theory. In this paper we introduce a new *Double Pendulum Chaotic Dataset* (Fig. 1), enabling development of

Modeling and Decision-making in the Spatiotemporal Domain Workshop, NIPS 2018, Montréal, Canada.

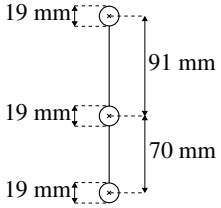
models that can deal with dynamics of a real, chaotic systems that goes beyond simple intuitive physics. We picked double pendulum as its behavior was well studied in the past [11, 3] and, despite extremely simple description of its current state, it exhibits complex motion patterns. The dataset, consisting of movies of double pendulum supplemented with ground truth positions of its datums and angles between arms, is the main contribution of the paper[1]. We also provide baseline results: a stacked LSTM operating on angles between arms in a challenging multi-step prediction setup, whereas the ultimate goal is to predict the motion straight from raw images in an end-to-end manner.

2 The Double Pendulum Chaotic Dataset

2.1 Mechanical device

A double pendulum is a pendulum with another pendulum attached to its end (Fig. 2a). Despite being a simple physical system, it exhibits a rich dynamic behavior with a strong sensitivity to initial conditions and noises in the environment (motion of the air in the room, sound vibrations, vibration of the table due to coupling with the pendulum etc.). Those influences at any given time will affect future trajectory in a way that is increasingly significant with time, making it a chaotic system [12].

(a)



(b)

Camera settings	
Focal length [mm]	50
Frame exposure time [µs]	90
Capture frame-rate [Hz]	400
Image resolution [px]	480x480
Image channels [-]	3

(c)

Figure 2: Hardware setup: (a) the used device, (b) its physical dimensions, (c) camera settings.

2.2 Data acquisition

Videos of the double pendulum were taken using a high-speed, *Phantom Miro EX2* camera, with settings presented in Tab. 2c. The camera's fast global shutter enabled us to take non-distorted frames, with a short exposure time to avoid any motion blur. To make the extraction of the arm positions easier, a matte black background was used, and the three datums were marked with red, green and blue fiducial markers. The markers were printed so that their diameter matches exactly that of the pendulum datums, which made their alignment easier. A powerful LED floodlight with a custom DC power supply (to avoid flicker) was used to illuminate the pendulum, to compensate for the short frame exposure time. The camera was placed at 2 meters from the pendulum, with the axis of the objective aligned with the first pendulum datum. The pendulum was launched by hand, and the camera was motion triggered. Our dataset was generated on the basis of 21 individual runs of the pendulum. Each of the recorded sequences lasted around 40s and consisted of around 17500 frames.

2.3 Marker position extraction

We implemented the program to extract the positions of the markers obtained from the video. The video frames were first upscaled 5 times to easily take advantage of subpixel positional resolution. The used *scikit-image* [17] to draw the fiducial markers templates. These templates were used with the OpenCV [2] cross-correlation algorithm to find the best matches on a per frame basis. The found matching markers were finally distinguished on the basis of their color.

2.4 Formulation of the challenge and description of the dataset

The proposed challenge is to predict the next 200 consecutive time-steps on the basis of the past 4 consecutive time-steps. For that purpose we have preprocessed the original 21 sequences in a way described below. Statistics of the resulting dataset are presented in Tab. 1.

[1] Available at `https://ibm.github.io/double-pendulum-chaotic-dataset/`.

Table 1: Dataset statistics

Data Representations		Set name	# of Seq.	Seq. Lengths
Image	480x480x3	Training	39	from 637 to 16850
Marker positions	(x_r, y_r), (x_g, y_g), (x_b, y_b)	Validation	24	204 = 4(i) + 200(t)
Arm angles	$\sin(\alpha)$, $\cos(\alpha)$, $\sin(\beta)$, $\cos(\beta)$	Test	60	204 = 4(i) + 200(t)

We extracted 5% of the data as "validation and test sets", in such a way that those sequences were homogeneously spread over the data and the runtime of the pendulum runs. In order to avoid strong correlations between the training and the validation/test sets, we discarded 200 time-steps before and after each of the extracted sequence. That resulted in 123 non-overlapping sequences: 39 training sequences of varying length (from 637 to 16850 time-steps) and 84 validation/test sequences of 204 time-steps each. In the latter case the first 4 steps represent the inputs **(i)**, and the next 200 steps correspond to the targets **(t)**. Finally, we randomized the order of all files.

We supplement the original images with two additional representations: **marker positions** and **arm angles**. Marker positions are three pairs (x,y) representing image coordinates of all three markers (each value is multiplied by 5, as explained in Sec. 2.3). Arm angles are sines and cosines of two angles α and β, where α is the angle between the horizontal image line pointing right and the the first arm, whereas β is the angle between the first and second arm.

It is worth noting that one might combine and use different representations for inputs and targets/predictions, what regulates the difficulty of the challenge. In particular, using raw images as both inputs and targets seems to be the most complex task, whereas utilization of arm angles as inputs and targets reduces the task into classic multiple-input multiple-output time-series prediction.

3 Time-series prediction baseline

In this section we provide results obtained with a recurrent neural network baseline. To make it as simple as possible, we used the angle representation (i.e. a vector of 4 values), both as inputs and targets of the model. The model consisted of a 4-layered stacked LSTM (Long-Short Term Memory) [7, 15], consuming a single input at a given single time-step, and followed by a fully-connected layer producing one output per time-step. Each LSTM cell had 32 hidden units.

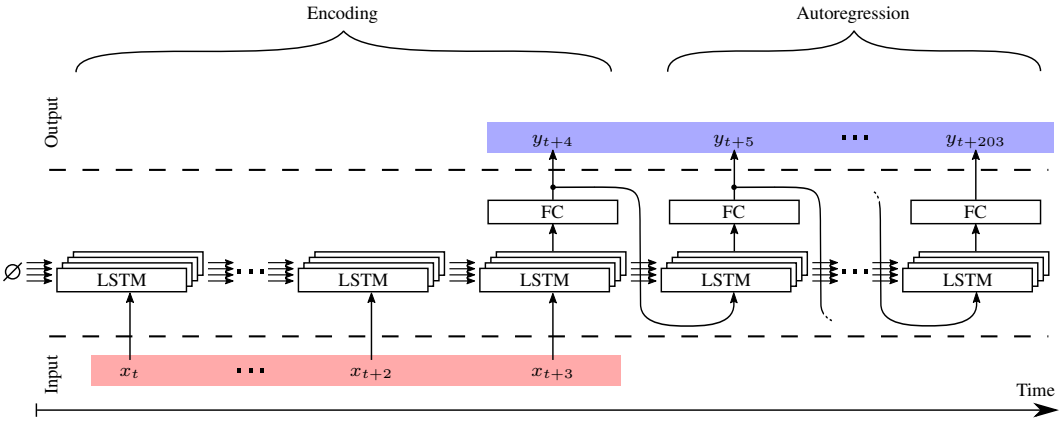
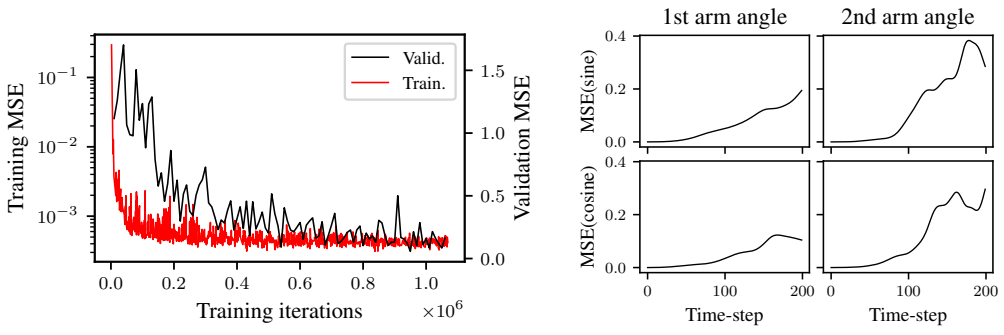


Figure 3: Baseline model in autoregressive, multi-step prediction during validation/testing.

The training was carried out on batches of 100 sequences, each picked at random positions from the training set. In training we have used *teacher forcing* [21], i.e. we started with a blank state and used sequences of 204 time-steps. Steps 0 to 202 were fed as input, and steps 1 to 203 were used as ground truths, resulting in learning one-step-ahead predictions. We used the Adam optimizer [8] with learning rate $1e^{-3}$ operating on the Mean Square Error (MSE) between predictions and targets.

During validation and testing the model was working in an *autoregressive mode*, as depicted in Fig. 3. We started with a blank LSTM state, input time-steps 0 to 3 sequentially, and kept only the latest



(a) (b)

Figure 4: (a) Training convergence. (b) MSE per time-step, averaged over all the test sequences.

prediction, that became input at time-step 4. From that moment we iteratively fed the predictions back to the model to get next predictions up to time-step 203. Please note that this heavily impacts loss; Fig. 4a presents convergence curves for both training and validation sets, where the latter is much higher due to autoregressive operation mode of the model. Training was stopped at 1.01 million iterations, with final test MSE loss equal to 0.09. Fig. 4a presents MSE per prediction time-step averaged over the whole test set; please notice that MSE systematically grows, which means that in autoreggresive mode our model handles well predictions only few dozens steps ahead. Fig. 5 presents some examples of relatively good (left) and bad (right) predictions. A movie with 200 time-steps ahead predictions made by our baseline is available at `https://youtu.be/AH6HNlgpBos`.

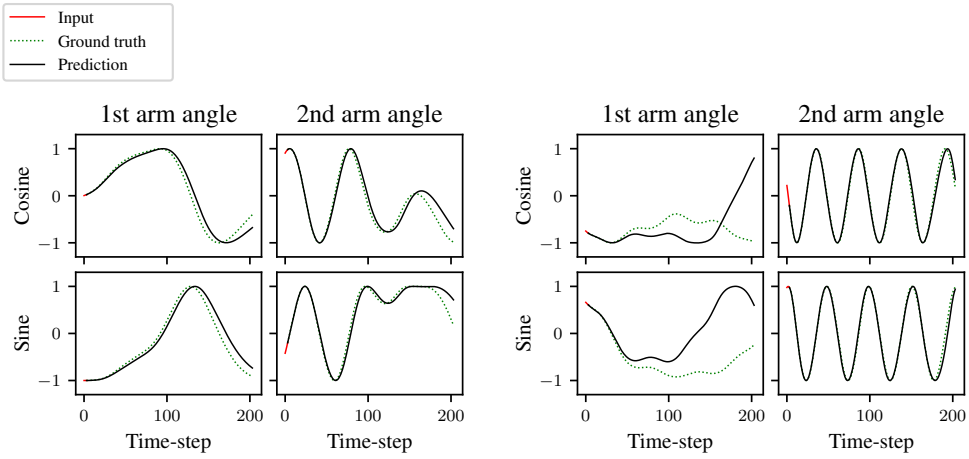


Figure 5: Exemplary predictions of our model for two test sequences

4 Summary

Despite the obvious advantages of simulation we believe that learning from real-world data is crucial for training models that have to cope with unobservable variables and randomness. Double pendulum, due to its chaotic behavior and simple state description, seems to be good object of interets. Thus we have recorded several videos of its motion using high-speed camera, processed them into a dataset and proposed the associated challenge. We also provide a simple baseline and show that is capable of predicting the only few dozens future positions correctly, leaving the challenge for end-to-end training straight from the images open. We hope the dataset will become a standard for benchmarking models inhibiting chaotic behaviors, and enable further research on spatio-temporal predictions.

References

[1] P. Battaglia, R. Pascanu, M. Lai, D. J. Rezende, et al. Interaction networks for learning about objects, relations and physics. In *Advances in Neural Information Processing Systems*, pages 4502–4510, 2016.

[2] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

[3] R. Cross. A double pendulum swing experiment: In search of a better bat. *American journal of physics*, 73(4):330–339, 2005.

[4] S. Ehrhardt, A. Monszpart, N. J. Mitra, and A. Vedaldi. Learning a physical long-term predictor. *arXiv preprint arXiv:1703.00247*, 2017.

[5] J. D. Farmer and J. J. Sidorowich. Predicting chaotic time series. *Physical review letters*, 59(8):845, 1987.

[6] K. Fragkiadaki, P. Agrawal, S. Levine, and J. Malik. Learning visual predictive models of physics for playing billiards. *arXiv preprint arXiv:1511.07404*, 2015.

[7] S. Hochreiter and J. Schmidhuber. Long short-term memory. 9:1735–80, 12 1997.

[8] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[9] J. R. Kubricht, K. J. Holyoak, and H. Lu. Intuitive physics: Current research and controversies. *Trends in cognitive sciences*, 21(10):749–759, 2017.

[10] A. Lerer, S. Gross, and R. Fergus. Learning physical intuition of block towers by example. *arXiv preprint arXiv:1603.01312*, 2016.

[11] R. Levien and S. Tan. Double pendulum: An experiment in chaos. *American Journal of Physics*, 61(11):1038–1044, 1993.

[12] R. B. Levien and S. M. Tan. Double pendulum: An experiment in chaos. *American Journal of Physics*, 61(11):1038–1044, 1993.

[13] E. N. Lorenz. Deterministic non-periodic flow. *Journal of the atmospheric sciences*, 20(2):130–141, 1963.

[14] M. C. Mackey, L. Glass, et al. Oscillation and chaos in physiological control systems. *Science*, 197(4300):287–289, 1977.

[15] R. Pascanu, Ç. Gülçehre, K. Cho, and Y. Bengio. How to construct deep recurrent neural networks. *CoRR*, abs/1312.6026, 2013.

[16] O. E. Rössler. An equation for continuous chaos. *Physics Letters A*, 57(5):397–398, 1976.

[17] S. van der Walt, J. L. Schönberger, J. Nunez-Iglesias, F. Boulogne, J. D. Warner, N. Yager, E. Gouillart, T. Yu, and the scikit-image contributors. scikit-image: image processing in Python. *PeerJ*, 2:e453, 6 2014.

[18] N. Watters, A. Tacchetti, T. Weber, R. Pascanu, P. Battaglia, and D. Zoran. Visual interaction networks. *arXiv preprint arXiv:1706.01433*, 2017.

[19] A. S. Weigend and N. A. Gershenfeld, editors. *Predicting the Future and Understanding the Past: a Comparison of Approaches. Proceedings of the NATO Advanced Research Worhhop on Time Series Analysis and Forecasting*. Santa Fe Institute Studies in the Sciences of Complexity, Addison-Wesley, 1993.

[20] A. S. Weigend and N. A. Gershenfeld. Results of the time series prediction competition at the santa fe institute. In *Neural Networks, 1993., IEEE International Conference on*, pages 1786–1793. IEEE, 1993.

[21] R. J. Williams and D. Zipser. A learning algorithm for continually running fully recurrent neural networks. *Neural computation*, 1(2):270–280, 1989.

[22] J. Wu, E. Lu, P. Kohli, B. Freeman, and J. Tenenbaum. Learning to see physics via visual de-animation. In *Advances in Neural Information Processing Systems*, pages 152–163, 2017.