Impact Factor: 4.14 (Calculated by SJIF-2015)
e- ISSN: 2348-4470
p- ISSN: 2348-6406

International Journal of Advance Engineering and Research Development

Volume 3, Issue 4, April -2016

Learning Non-linear Dynamical Systems From Raw Images

Amit Chahar[1], Anil Kumar[2],Rohit Rajput[3], Venkata Ramesh L[4], Prof. Sunil Dhore[5]

[1-5]Computer Engineering, Army Institute of Technology

Abstract —*We introduce a method for model learning and control of non-linear dynamical systems from raw pixel images. It consists of a deep generative model, belonging to the family of variational autoencoders, that learns to generate image trajectories from a latent space in which the dynamics is constrained to be locally linear. Our model is derived directly from an optimal control formulation in latent space, supports long-term prediction of image sequences and exhibits strong performance on a variety of complex control problems.For capturing the information of non-linear object's behavior, we need to use high-dimensional data. Processing the high-dimensional data is expensive and not feasible. So, in this model, first Auto-encoder is used for dimensionality reduction, and after prediction method (transition mapping) is used, and the imagereconstructed. We demonstrate that our model enables learning good predictive models of dynamical systems from pixel information only.*

Keywords*-machine learning, autoencoder, neural networks, latent space,non-linear systems, prediction, dynamical systems.*

I. INTRODUCTION

Dynamical systems are mathematical objects used to model physical phenomena whose state (or instantaneous description) changes over time. These models are used in financial and economic forecasting, environmental modeling, medical diagnosis, industrial equipment diagnosis, and a host of other applications. If we have two short movies of billiards balls rolling around on a table without friction, we could not tell which was recorded first. Hence this system is stationary. On the other hand, if there is friction, then we are in the non-stationary situation, because the balls will slow down as time progresses, and their speed gives us a way of deducing when the observation was made.

A key challenge is system identification, i.e. finding a mathematical model ofthe dynamical system based on the information provided by measurements from the underlying system. In the context of state-space models this includes finding two functional relationships between (a) the states at different time steps (prediction/transition model) and (b) states and corresponding measurements (observation/ measurement model)[1].

Control of non-linear dynamical systems with continuous state and action spaces is one of the key problems in robotics and, in a broader context, in reinforcement learning for autonomous agents. A prominent class of algorithms that aim to solve this problem are model-based locally optimal (stochastic) control algorithms. When combined with receding horizon control , and machine learning methods for learning approximate system models, such algorithms are powerful tools for solving complicated control problems [3, 4, 5]; however, they either rely on a known system model or require the design of relatively low-dimensional state representations. For real autonomous agents to succeed, we ultimately need algorithms that are capable of controlling complex dynamical systems from raw sensory input (e.g. images) only. In this paper we tackle this difficult problem.

II. NON-LINEAR DYNAMICAL SYSTEMS

A dynamical system will be defined to be a system in which the present state (the values of all of the variables and all of their derivatives) is somehow dependent on previous states of the system. A deterministic system will be taken to be a system in which the present state is entirely dependent on previous states of the system. A linear system is a system in which all of the dependence of the current state on previous states can be expressed in terms of a linear combination. A linear stochastic system is a system in which all of the dependence of the current state on previous states can be expressed in terms of a linear combination and the residual unpredictable portions can be expressed as additive, independent, identically distributed, random variables[2].

A nonlinear system is a system in which the dependence of the current state onprevious states cannot be expressed entirely as a linear combination; even if some of the dependence can be captured in a linear combination of the previous states, something extra is required to capture all of the dependence.

Simulations of nonlinear dynamical systems have shown that nonlinear time series can be entirely deterministic, that is generated without any random component, and yet exhibit behavior which appears to have an error variance when analyzed by linear statistical methods. This work will present a variety of techniques for the analysis of nonlinear time series which have the potential to be modeled as signal portions of time series that are often discarded as noise. Learning non-linear dynamical models from veryhigh-dimensional sensor data is even more challenging. First, finding (non-linear) functional relationships in very high dimensions is hard (un-identifiability, local optimal, over-fitting, etc.); second, the amount ofdata required to find a good function approximation is enormous. Fortunately, high-dimensional data often possesses an intrinsic lower dimensionality. Wewill exploit this property for system identification by finding a low-dimensional representation of high-dimensional data and learning predictive models in this low-dimensional space.

In this paper, we combine feature learning and dynamical systems modeling to obtain good predictive models for high-dimensional time series. We use auto-encoder neural networks for automatically finding a compact low-dimensional representation of an image. In this low-dimensional featurespace, we use a neural network for modeling the non-linear system dynamics.

An encoder g^{-1} maps an image Y_{t-1} at time step *t-1* to a low-dimensional feature z_{t-1}. In this feature space, a prediction model *l* maps the feature forward in time to z_t. Subsequently, the decoder *g* canbe used to generate a predicted image y_t at the next time step. This framework needs access to both the encoder g^{-1} and the decoder *g*, which motivates our use of the auto-encoder as dimensionality reduction technique.

III. MODEL

We consider a dynamical system where control inputsare denoted by u and observations are denotedby y. In the context of this paper, the observationsare pixel information from images. We assume thata low-dimensional latent variable z exists that compactlyrepresents the relevant properties of y. Sincewe consider dynamical systems, a low-dimensional representationz of a (static) image y is insufficient tocapture important dynamic information, such as velocities.Therefore, we introduce an additional latentvariable x, the state. In our case, the state x_t containsfeatures from multiple time steps (e.g., t-1 and t) tocapture velocity (or higher-order) information. Therefore,our transition model does not map features attime t -1 to 1 to time t, but thetransition function f maps states x_{t-1} (and controlsu_{t-1}) to states x_t at time t. The full dynamical systemis given as the state-space model

$$x_{t+1} = f(x_t, u_t; \theta) + w_t(\theta)$$
$$z_t = h(x_t; \theta) + v_t(\theta)$$
$$y_t = g(z_t; \theta) + e_t(\theta)$$

where each measurement yt can be described by alow-dimensional feature representation z_t . Thesefeatures are in turn modeled with a low-dimensionalstate-space model in, where the statex_t contains the full information about the state of thesystem at time instant t. Here $w_t(\theta)$,$v_t(\theta)$ and $e_t(\theta)$ are sequences of independent randomvariables and θ are the model parameters[1,6,7].

To identify parameters in dynamical systems, the prediction-error method has been applied extensively within the system identification community during the last five decades. It is based on minimizing the error between the sequence of measurements y_t and the predictions $\hat{y}_{t|t-1}(\theta)$, usually the one-step ahead prediction. To achieve this, we need a predictor model that relates the prediction $\hat{y}_{t|t-1}(\theta)$ to all previous measurements, control inputs and the system parameters θ.In general, it is difficult to derive a predictor model based on the nonlinear state-space model, and a closed form expression for the prediction is only available in a few special cases. However, by approximating the optimal solution, a predictor model for the features z_t can be stated in the form

$$\hat{z}_{t|t-1}(\theta_M) = l(Z_{t-1};\theta_M)$$

where $Z_{t-1} = (z_1 , u_1 ,\ldots, z_{t-1}, u_{t-1})$ includes all pastfeatures and control inputs, *l* is a nonlinear functionand θ_M is the corresponding model parameters.

IV. AUTOENCODERS

An autoencoder neural network is an unsupervised learning algorithm that applies backpropagation, setting the target values to be equal to the inputs. I.e., it uses y (i) = x (i). The autoencoder tries to learn a function $h_{W,b}(x) \approx x$. In other words, it is trying to learn an approximation to the identity function, so as to output *x'* that is similar to *x*. The identity function seems a particularly trivial function to be trying to learn; but by placing constraints on the network, such as by limiting the number of hidden units, we can discover interesting structure about the data.

Architecturally, the simplest form of an autoencoder is a feedforward, non-recurrent neural net which is very similar to the multilayer perceptron (MLP), with an input layer, an output layer and one or more hidden layers connecting them. The differences between autoencoders and MLPs, though, are that in an autoencoder, the output layer has the same number of nodes as the input layer, and that, instead of being trained to predict the target value *y* given inputs *x*, autoencoders are trained to reconstruct their own inputs *x*. Therefore, autoencoders are unsupervised learning models.

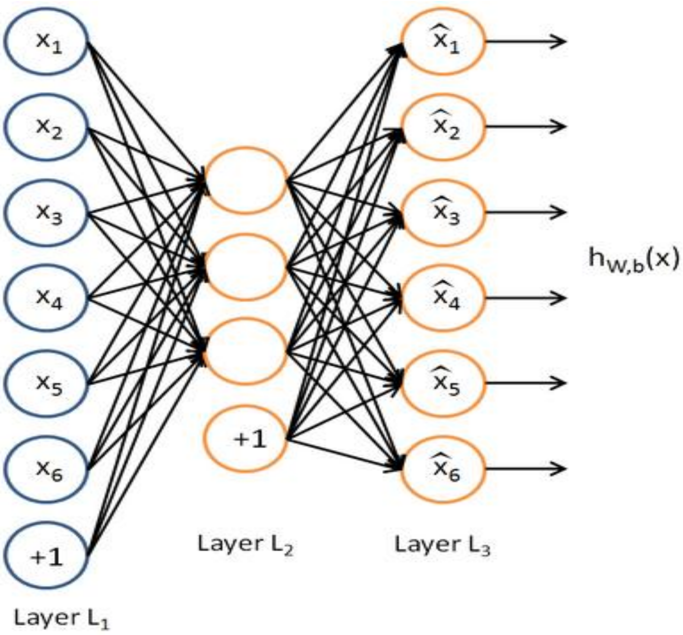

Fig 1. The general representation of autoencoder.

V. EXPERIMENTAL RESULTS

Model training. We consider two different network types for our model: Standard fully connected neural networks with up to three layers, which work well for moderately sized images, are used for the planar and swing-up experiments; A deep convolutional network for the encoder in combination with an up-convolutional network as the decoder which, in accordance with recent findings from the literature [8, 9], we found to be an adequate model for larger images. Training was performed using Adam [14] throughout all experiments. The training data set D for all tasks was generated by randomly sampling N state observations and actions with corresponding successor states. For the plane we used N = 2000 samples.

Baseline models. For a thorough comparison and to exhibit the complicated nature of the tasks, we also test a set of baseline models on the plane: a standard variational autoencoder (VAE) and a deep autoencoder (AE) are trained on the autoencoding subtask for visual problems. That is, given a data set D used for training our model, we remove all actions from the tuples in D and disregard temporal context between images. After autoencoder training we learn a dynamics model in latent space.

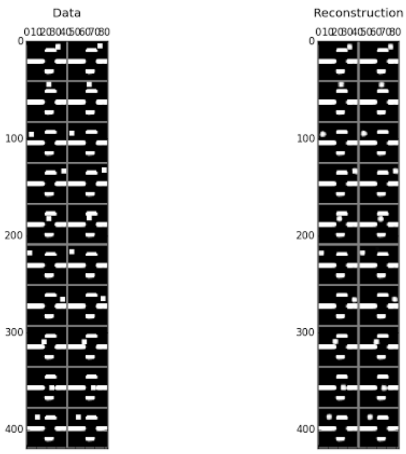

Fig 2. Left column are x_t, x_{t+1}, and right column are the reconstructed predictions.

Control in planar system. The agent in the planar system can be moved in a two dimensional plane limited by the choice of a continuousdisplacement in x and y-direction. High-dimensional representation of a state is a 40 × 40black and white image. Obstructed by fourrectangular obstacles, the task is to move to the right bottomimage, from a random x position at the top of the image. Encodings obstaclesare obtained before planning and an additional term of quadratic cost is penalizing proximity to them.[12,10]

A representation of the observations on which control is performed - along with their correspondingstatus values and fouling latent space - is shown in Figure 2. While trained separatelyautoencoders make aesthetically pleasing images, models failed to discover the underlyingstructure of the state space, which complicates the estimation of the dynamic and largely based invalidate costsat distances in the space. Including unrealized dynamic constraints on these models in end to endOn the other hand, it produces latent spaces approach optimal embedding planar.

VI. CONCLUSION AND FUTURE WORK

We presented a system for stochastic optimal control on high-dimensional image streams. Key to the approach is the extraction of a latent dynamics model which is constrained to be locally linear in its state transitions. An evaluation on four challenging benchmarks revealed that it can find embeddings on which control can be performed with ease, reaching performance close to that achievable by optimal control on the real system model. Future work includes predicting long sequence of images in an infinite time-series data.

REFERENCES

[1] Niklas Wahlstrom, Thomas B. Schon and Marc Peter Deisenroth, Learning deep dynamical models from image pixels, published on arXiv:Oct 2014.

[2] E. Todorov and W. Li. A generalized iterative LQG method for locally-optimal feedback control of constrained nonlinear stochastic systems. In ACC. IEEE, 2005.

[3] Y. Tassa, T. Erez, and W. D. Smart. Receding horizon differential dynamic programming. In Proc. Of NIPS, 2008.

[4] Y. Pan and E. Theodorou. Probabilistic differential dynamic programming. In Proc. of NIPS, 2014.

[5] S. Levine and V. Koltun. Variational policy search via trajectory optimization. In Proc. of NIPS, 2013.

[6] D. P. Kingma and M. Welling. Auto-encoding variational bayes. In Proc. of ICLR, 2014.

[7] D. J. Rezende, S. Mohamed, and D. Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In Proc. of ICML, 2014.

[8] M. D. Zeiler, D. Krishnan, G. W. Taylor, and R. Fergus. Deconvolutional networks. In CVPR, 2010.

[9] A. Dosovitskiy, J. T. Springenberg, and T. Brox. Learning to generate chairs with convolutional neural networks. In Proc. of CVPR, 2015.

[10] R. F. Stengel. Optimal Control and Estimation. Dover Publications, 1994.

[11] W. Li and E. Todorov. Iterative Linear Quadratic Regulator Design for Nonlinear Biological Movement Systems. In Proc. of ICINCO, 2004.

[12] M. Toussaint. Robot Trajectory Optimization using Approximate Inference. In Proc. of ICML, 2009.

[13] M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul. An introduction to variational methods for graphical models. In Machine Learning, 1999.

[14] D. Kingma and J. Ba. Adam: A method for stochastic optimization. In Proc. of ICLR, 2015.

[15]P. Vincent, H. Larochelle, Y. Bengio, and P.A. Manzagol. Extracting and composing robust features with denoising autoencoders. In ICML,2008.

[16] J. G. Schneider. Exploiting model uncertainty estimates for safe dynamic control learning. In Advances in Neural Information Processing Systems (NIPS). Morgan Kaufman Publishers, 1997.

[17] B. Scholkopf, A. J. Smola, and K.R. Muller. Nonlinear Component Analysis as a Kernel Eigenvalue Problem. Neural Computation, 10(5):1299-1319, 1998.

[18] T. B. Schon, A. Wills, and B. Ninness. System identication of nonlinear state-space models. Automatica, 47(1):39-49, 2011.

