Learning 2D Linear Dynamics in Image Space Using Neural Networks

Jeffrey Mahler*, Michael Laskey*, John Schulman, Sergey Levine, Jeff Donahue, Sachin Patil, Pieter Abbeel, and Ken Goldberg
* denotes equal contribution

***Abstract*—Grasping and manipulating a previously unknown object without assumptions about structure or relevant features in the environment is an especially challenging problem in robotics. Motivated by recent work on feature learning with deep neural networks, we approach this problem by learning a dynamical system that relates robotic control inputs (e.g., motor torques) to features learned from raw sensory data (e.g., images) using a deep neural network. We present an algorithm for inferring the most likely parameters of this system using approximate expectation maximization. Initial experiments show that the method is able to predict the next image of a pendulum in a 2D simulator to less than 1 pixel of mean squared error using only the current image and the control inputs to the system.**

I. Introduction

A fundamental challenge in robotics is to plan for grasping or manipulation tasks without a known model for the object of interest. For example, a robot in the home might have to manipulate objects with unknown mass distributions, coefficients of friction, or moments of inertia while dealing with occlusions. One approach to achieving this is to learn a dynamics model that relates the current robotic controls applied to the unknown object (e.g., motor torques) and current sensor data (e.g., images) to expected future sensor data. This predictive model could be used to build a controller directly around sensor data for a manipulation or grasping task.

For many robots the primary sensing modality is images, either acquired with color cameras or depth sensors like the Kinect. However, planning directly with images is challenging due to their high dimensionality and highly non-linear relationship with the physical state (e.g., position, orientation) of the system being imaged. This raises the problem of how to best extract a lower dimensional representation from images that is relevant to the planning task at hand. Visual servoing approaches address this problem by using hand-tuned features, such as SIFT or HOG, to identify key points on the object of interest and use these features to infer the pose of the object, along with a camera model an approximate geometry of the object; see [3] for an overview. However, due to the use of hand-tuned features, learning the dynamics of previously unmodeled objects can be difficult.

In this work, we discuss an algorithm for learning the dynamics of an object directly in image space from only applied controls and their corresponding image observations. Motivated by recent successes in using deep neural networks to learn feature representations from images [9], we model the problem as that of learning dynamics in the space of features generated by a neural network. We learn the maximum-likelihood parameters of both the dynamics and the neural network using expectation maximization. Our approach is similar to the works of Ghahramani et al. [6] or Briegel et al. [2], but using a deep neural network as an observation function.

The problem of learning dynamics models for robotic control has been well studied [12], but the majority of past work has not addressed how to jointly learn this model and extract relevant information from images. Siddiqi et al. addressed this by learning a linear dynamical model for the top 10 principal components of stereo image and laser range data on a Botrics O-bot [13]. Boots et. al used kernel-based Predictive State Represenations to learn a predictive model of depth images from motor commands and their corresponding observations [1]. Deep neural networks (DNNs) have been used successfully for learning relevant features from images in a variety of applications, including image classification, reinforcement learning, and modeling time series [9, 10, 7, 14]. DNNs have also been used to learn dynamical models of speech features over a discrete state space [4]. In contrast, our work deals with image data and continuous hidden state spaces.

II. Method

We consider the following dynamics model:

$$\begin{aligned} \mathbf{x}_{t+1} &= A\mathbf{x}_t + B\mathbf{u}_t + \mathbf{w}_t & \mathbf{w}_t &\sim N(0, \sigma_1^2 I) \\ \mathbf{y}_t &= h(\mathbf{x}_t) + \mathbf{v}_t & \mathbf{v}_t &\sim N(0, \sigma_2^2 I) \end{aligned}$$

where $\mathbf{x}_t \in \mathbb{R}^m$ is the hidden state of the system, $\mathbf{u}_t \in \mathbb{R}^p$ is the applied controls, such as forces and torques on the object, and $\mathbf{y}_t \in \mathbb{R}^n$ is the vectorized image observation. The matrices $A \in \mathbb{R}^{m \times m}$ and $B \in \mathbb{R}^{m \times p}$ correspond to the linear dynamical system, and $\mathbf{w}_t \in \mathbb{R}^m$ and $\mathbf{v}_t \in \mathbb{R}^n$ are the dynamics and observation noises, repectively. The nonlinear function $h : \mathbb{R}^m \to \mathbb{R}^n$, which we will refer to as a 'decoder', is a deep neural network acting as a non-linear observation function. The network is parameterized by the layerwise weights and biases $\theta_h = \{W_i^{(h)}, b_i^{(h)}\}$ [11]. The full parameters of this model are denoted $\theta = \{A, B, c, \sigma_1, \sigma_2, \theta_h\}$. We treat the observations $\mathbf{y}_t$ and controls $\mathbf{u}_t$ as inputs from a training sequence.

Subsituting the transition and observation distributions from our model into the standard LDS equations [5] and taking the logarithm we obtain the complete log-likelihood:

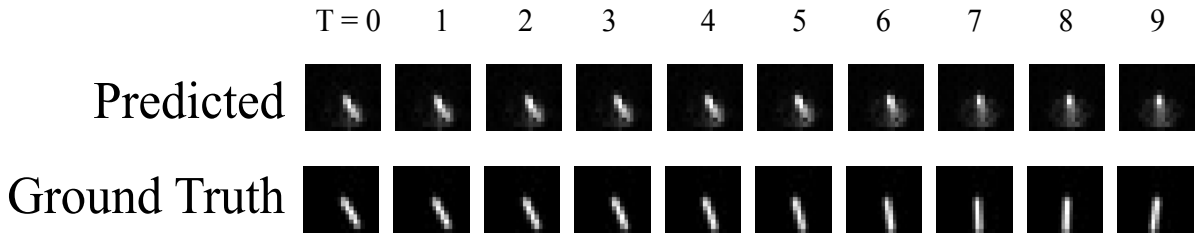

Fig. 1. Comparison of predicted observations using our trained model (top) versus ground truth observations (bottom) for a sequence of 10 test images of a pendulum. The model was trained for 1,000 iterations of EM. The predictions are close to the ground truth values for several timesteps, while predictions are blurred in later timesteps due to uncertainty

$$l_c(\theta) = \log(p(\mathbf{x}_0)) + \sum_{t=0}^{T-1} \log(p(\mathbf{x}_{t+1}|\mathbf{x}_t, \mathbf{u}_t)) + \sum_{t=1}^{T} \log(p(\mathbf{y}_t|\mathbf{x}_t))$$

which we then maximize using approximate expectation maximization, described below.

A. E-Step

A common method for performing the expectation step for a linear dynamical system is to perform Rauch-Tung-Striebel (RTS) smoothing over the hidden variables [5]. With a nonlinear observations function, the expectation of the hidden state $\mathbf{x}_t$ given an observation $\mathbf{y}_t$ cannot be computed exactly. Thus, we use linearize the observation function at each time step in an extended Kalman filter (EKF), computing the Jacobian at each state by running backpropagation through the decoding network for each dimension of the output $h(\mathbf{x})_i$. This results in a Gaussian distribution over hidden states $P(\mathbf{x}_{t+1}|\mathbf{y}_0, \mathbf{y}_1, ..., \mathbf{y}_T)$.

B. M-Step

In our maximization step, we find the approximate maximum-likelihood estimates of the parameters θ given the distribution on hidden states from the EKF. The LDS model decouples the maximization of the noise parameters, linear dynamics parameters, and the observation parameters. The linear dynamics parameters are estimated using weighted least squares with weights given by the smoothed variance estimates [5]. See [13] for how to derive the updated dynamics and observation noise. We cannot maximize the parameters of the decoding function $h(\mathbf{x})$ in closed form because the distribution of the neural network outputs is no longer Gaussian. Therefore we approximate the integral over hidden states in the log-likelihood using Monte-Carlo integration, generating samples from the state distributions given by Kalman smoothing. We then maximize the likelihood function with respect to the decoder weights over these samples using backpropagation, initializing the parameters θ_h with the parameters from the previous iteration.

III. Experiments

In our preliminary experiments, we simulated a pendulum moving about a fixed point under gravitational forces. The pendulum was actuated by forces applied to the free end of the pendulum, there were gravitational forces but no friction, and the center of mass was located in the center of the pendulum. Note that the dynamics of this system are non-linear with respect to the angle of the pendulum. We randomly applied forces and collected 9,000 grayscale training images and 1,000 grayscale test images along with the direction and magnitude of the applied forces and velocities at each timestep.

Our encoding and decoding neural networks each consisted of 2 hidden layers of sizes 100 and 10, with a sigmoidal non-linearity at each hidden layer. The neural network was sparsely initialized with random weights chosen from a zero-mean Gaussian distribution, and we optimized its parameters for each M-step using the Caffe library [8].

After 1,000 iterations of EM, our model was able to predict the next image of the pendulum given the current image and controls with a mean squared error of 1.22 pixels on the training set and 0.86 pixels on the test set. Training took approximately 14 seconds per E-step and 10 seconds per M-step. We compare a test 10-image sequence of the mean images predicted using our model given an initial image and planned controls with the ground truth images in Fig. 1. We see that for the first few timesteps the images are predicted accurately, and later images are blurred around the true location of the pendulum, reflecting the gaussian uncertainty in the hidden state estimate.

IV. Conclusion

We presented an EM-inspired algorithm for learning dynamics directly from raw images using a deep neural network observation function. Initial experiments demonstrate that our method is able to predict images of a pendulum in a 2D simulator for several time steps, suggesting that it could be used to build a controller around a plan specified directly in image space.

This is a first step towards predicting observation data using deep neural networks, but questions remain with respect to using this method for planning and manipulation in an actual robot workspace. First, it remains to be seen whether this method would extend to natural images or to depth data.

Given past success in reconstructing natural images using deep autoencoders, we believe that this would be possible given enough data and a well-chosen network architecture. Second, future work will need to validate that this methodology is useful for planning and manipulation. We will run experiments to control systems from images using model-based reinforcement learning techniques and to plan controllers around trajectories that were demonstrated to the robot in image space. Third, for a specific planning or manipulation task involving non-linear dynamics, a model of the dynamics needs to only be accurate in the task-relevant part of state space to achieve success. We will research combining our method with guided exploration to choose training trajectories that are relevant to a specific task at hand. Fourth, it is not clear how well our method works on systems with discontinuous dynamics, as is common in systems with collisions. We will run experiments with discontinuities such as collisions with walls or other objects, and we will also compare this with using mixtures of linear functions or non-linear functions in the dynamics.

REFERENCES

[1] Byron Boots, Arunkumar Byravan, and Dieter Fox. Learning predictive models of a depth camera & manipulator from raw execution traces. 2014.

[2] Thomas Briegel and Volker Tresp. Fisher scoring and a mixture of modes approach for approximate inference and learning in nonlinear state space models. *Advances in Neural Information Processing Systems*, pages 403–409, 1999.

[3] François Chaumette and Seth Hutchinson. Visual servo control. i. basic approaches. *Robotics & Automation Magazine, IEEE*, 13(4):82–90, 2006.

[4] Li Deng and Roberto Togneri. Deep dynamic models for learning hidden representations of speech features. 2014.

[5] Zoubin Ghahramani. Parameter estimation for linear dynamical systems. Technical report, 1996.

[6] Zoubin Ghahramani and Sam T Roweis. Learning nonlinear dynamical systems using an em algorithm. *Advances in neural information processing systems*, pages 431–437, 1999.

[7] Geoffrey E Hinton, Alex Krizhevsky, and Sida D Wang. Transforming auto-encoders. In *Artificial Neural Networks and Machine Learning–ICANN 2011*, pages 44–51. Springer, 2011.

[8] Yangqing Jia. Caffe: An open source convolutional architecture for fast feature embedding. http://caffe.berkeleyvision.org/, 2013.

[9] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, volume 1, page 4, 2012.

[10] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

[11] Andrew Ng. Sparse autoencoder. http://web.stanford.edu/class/cs294a/sparseAutoencoder_2011new.pdf, 2011.

[12] Duy Nguyen-Tuong and Jan Peters. Model learning for robot control: a survey. *Cognitive processing*, 12(4):319–340, 2011.

[13] Sajid M Siddiqi, Byron Boots, and Geoffrey J Gordon. A constraint generation approach to learning stable linear dynamical systems. Technical report, DTIC Document, 2008.

[14] Ilya Sutskever, Geoffrey E Hinton, and Graham W Taylor. The recurrent temporal restricted boltzmann machine. In *Advances in Neural Information Processing Systems*, pages 1601–1608, 2009.