Neurocomputing 141 (2014) 3–14

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Neural learning of vector fields for encoding stable dynamical systems

A. Lemme, K. Neumann, R.F. Reinhart, J.J. Steil

Research Institute for Cognition and Robotics (CoR-Lab), Bielefeld University, Germany

ARTICLE INFO

Article history:
Received 8 July 2013
Received in revised form
3 February 2014
Accepted 7 February 2014
Available online 8 April 2014

Keywords:
Extreme learning machine
Neural network
Vector fields
Dynamical systems learning
Motion generation
Movement generation

ABSTRACT

The data-driven approximation of vector fields that encode dynamical systems is a persistently hard task in machine learning. If data is sparse and given in the form of velocities derived from few trajectories only, state-space regions exist, where no information on the vector field and its induced dynamics is available. Generalization towards such regions is meaningful only if strong biases are introduced, for instance assumptions on global stability properties of the to-be-learned dynamics. We address this issue in a novel learning scheme that represents vector fields by means of neural networks, where asymptotic stability of the induced dynamics is explicitly enforced through utilizing knowledge from Lyapunov's stability theory, in a predefined workspace. The learning of vector fields is constrained through point-wise conditions, derived from a suitable Lyapunov function candidate, which is first adjusted towards the training data. We point out the significance of optimized Lyapunov function candidates and analyze the approach in a scenario where trajectories are learned and generalized from human handwriting motions. In addition, we demonstrate that learning from robotic data obtained by kinesthetic teaching of the humanoid robot iCub leads to robust motion generation.

1. Introduction

The approximation of vector fields from sparse data that represent dynamical systems, e.g. to encode quantitative flow visualization [1], optical flow in computer vision [2] or force fields in motor control [3,4], is an important but also persistently hard task for learning algorithms. In recent work, vector fields were applied to learn and generate complex motions for robots [5,6]. In such scenarios, training data typically consist of only a few trajectories and thus leave many regions in the state space with no information of the desired vector field. Generalization towards regions subject to sparse sampling is challenging, because small errors in the approximation of the vector field can get amplified during integration and can lead to diverging behavior of the dynamical system.

Thus, a strong model bias is needed for generalization which has to be derived from prior knowledge about the underlying dynamics. In [7], a superposition of irrotational basis fields is used to approximate a variety of vector patterns, where it is assumed that the data originate from the gradient of a potential function. Kuroe and Kawakami introduced a combination of neural networks to reconstruct vector fields where prior knowledge of inherent vector field properties is used to enhance the accuracy [8,9].

Wave propagation [10], which was introduced to dynamic path planning in [11], can be used to create potential fields, which in turn can be adapted to demonstrations [12]. Following the respective gradients then leads to an inherently stable dynamical system.

Motion generation methods, developed in computational imitation learning and programming by demonstration, appear to be promising in order to generalize to unseen areas in the workspace by providing stable solutions [13–15]. The stability of the motion is ensured by a linear spring damper system, which generates a straight line with a biologically plausible velocity profile. The shape is then induced by adding a perturbation force term. These time-dependent perturbations are learned by means of a mixture of Gaussian functions. The force term is suppressed at the end of the motion ensuring stability, because only the linear components drive the dynamical system. An alternative approach to the standard DMP approach is the task-parameterized Gaussian mixture model (TpGMM [16]), where the parameterization of the motion is variable. It models a second order dynamical system, which uses a probabilistic representation of the demonstrations. This representation can be parameterized time-dependent, task-dependent or in combination.

For motion generation from vector fields, one prominent approach is the stable estimator of dynamical systems (SEDS [17]). This learning approach represents vector fields by a Gaussian mixture of linear dynamical systems. Learning is achieved by

E-mail addresses: alemme@cor-lab.uni-bielefeld.de (A. Lemme), kneumann@cor-lab.uni-bielefeld.de (K. Neumann), freinhar@cor-lab.uni-bielefeld.de (R.F. Reinhart), jsteil@cor-lab.uni-bielefeld.de (J.J. Steil).

http://dx.doi.org/10.1016/j.neucom.2014.02.012
0925-2312/

solving a nonlinear constrained optimization problem formulated as a quadratic program. The learned dynamical system then complies to a specific quadratic Lyapunov function. The main advantage of this method is that the learned dynamics are provably globally asymptotically stable. On the downside, the stability constraints may be too restrictive with respect to the motion that shall be learned. If the training data and the stability constraints contradict, accurate learning of the desired motion is prevented.

An extension of SEDS called SEDS-II was very recently published in [18] and implements less conservative stability conditions as compared to SEDS. This extension relies on a stabilization approach called control Lyapunov function derived from Artstein and Sontag's stability theory [19]. Such functions are used to stabilize nonlinear dynamical systems through online corrections at runtime and interfere with the learned dynamical system. This methodology can be applied in combination with any learning approach to represent the training data and leaves the stability issue to the online correction mechanism. However, the learning of dynamics that satisfy desired Lyapunov functions and guarantees stability without interfering with the data or requiring online corrections is so far only solved for special cases and remains difficult in the case of using a dynamical systems represented by vector fields.

These issues are partly addressed in [20]. Here a neural network approach is used to learn from demonstrations and to generate motions for the humanoid robot iCub. The accuracy performance and the stability are addressed by two separately trained but superimposed neural networks. The first network approximates the data while the second network addresses stability by learning a velocity field, which implements a contraction towards the desired movement trajectory. However, the superposition of two networks seems complex for representing only one motion. Additionally, no guarantee for stable motion generation is given.

The contribution of this paper is the introduction of Lyapunov theory in neural networks learning for stable motion generation. Therefore, we extend the ideas recently published in [21] and propose a novel learning approach. This approach is based on the idea to represent time-independent vector fields in one neural network that lead to asymptotically stable dynamics in a predefined workspace. The learning is separated into three steps: First, construct a suitable Lyapunov candidate through parameter optimization towards the data. Second, use the constructed Lyapunov candidate to obtain inequalities constraints for learning. Third, add inequality constraints which ensure that the dynamics cannot leave a predefined region. The inequality constraints are implemented by a quadratic program, which minimizes the error between the training data and the output of the network. To keep the amount of used constraints to a minimum, a sampling algorithm identifies problematic regions and adds constraints until the dynamical system is stabilized. Thus, the resulting vector field induces stable dynamics by construction. This approach is schematically illustrated in Fig. 1.

The reminder of this paper is organized as follows. In Section 2 we explain the theoretical basis of training neural networks with stability constraints. In Section 3 we show that the accuracy of the estimates is highly dependent on the applied Lyapunov candidate and show two different candidates in comparison. A rigorous analysis conducted here evaluates the relation between the regularization of the weights, the obtained errors, and the number of sampled constraints needed to implement stability. Additionally, it is demonstrated that the approach generates smooth and accurate motions in several experiments including also a kinesthetic teaching scenario with the humanoid robot iCub. Before we conclude this work in Section 5, we discuss the main features of this approach in Section 4.

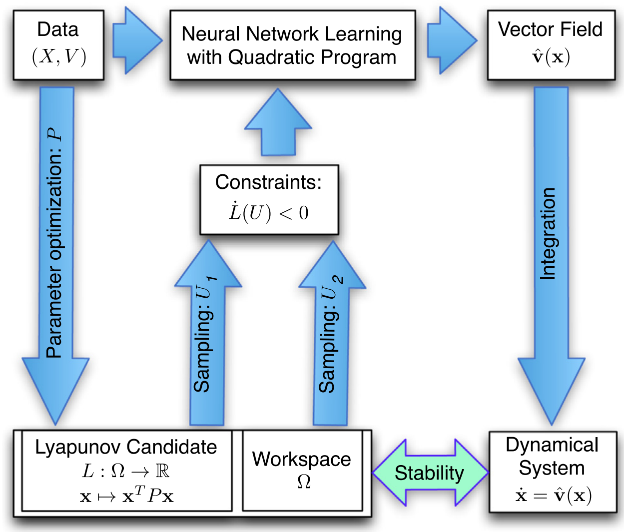

Fig. 1. Schematic view of the proposed approach to learn vector fields. The learning is separated into two main steps: (i) predefine a proper Lyapunov candidate through parameter optimization and (ii) use this function to sample inequality constraints that are implemented by a quadratic program learning the data and (iii) add constraints to restrict the motion to stay in the defined workspace. The resulting dynamical system approximates the data and is asymptotically stable in the defined workspace after learning.

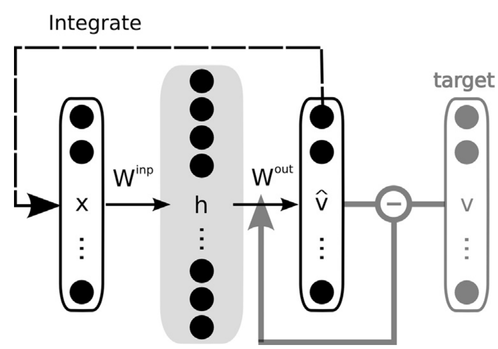

Fig. 2. ELM with its three layer structure used in an integration loop. Only the read-out weights are trained.

2. Extreme learning machine for estimation of vector fields

We consider trajectory data that are driven by time independent vector fields:

$$\dot{\mathbf{x}} = \mathbf{v}(\mathbf{x}), \quad \mathbf{x} \in \Omega, \tag{1}$$

where a state variable $\mathbf{x}(t) \in \Omega \subseteq \mathbb{R}^d$ at time $t \in \mathbb{R}$ with dimensionality d defines a state trajectory in the workspace Ω. It is assumed that the vector field $\mathbf{v}(\mathbf{x})$ is nonlinear and continuous with a single asymptotically stable point attractor $\mathbf{x}^*$ with $\mathbf{v}(\mathbf{x}^*) = 0$ in Ω. The limit of each trajectory in Ω thus satisfies:

$$\lim_{t \to \infty} \mathbf{x}(t) = \mathbf{x}^* : \forall \mathbf{x}(0) \in \Omega. \tag{2}$$

The key question of this paper is how to learn $\mathbf{v}$ as a function of $\mathbf{x}$ by using demonstrations for training and ensure its asymptotic stability at target $\mathbf{x}^*$ in Ω. The estimate is denoted by $\hat{\mathbf{v}}$ in the following. The evolution of motion can then be computed by numerical integration of $\dot{\mathbf{x}} = \hat{\mathbf{v}}(\mathbf{x})$, where $\mathbf{x}(0) \in \Omega$ denotes the starting point of the motion.

Consider the neural architecture depicted in Fig. 2 for estimation of $\mathbf{v}$. The figure shows a single hidden layer feed-forward neural network: $\mathbf{x} \in \mathbb{R}^d$ denotes the input, $\mathbf{h} \in \mathbb{R}^R$ the hidden, and