Neurocomputing 169 (2015) 343–349

Contents lists available at ScienceDirect

Neurocomputing

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

Exploiting similarity in system identification tasks with recurrent neural networks

Sigurd Spieckermann [a,b,*], Siegmund Düll [a,c], Steffen Udluft [a], Alexander Hentschel [a], Thomas Runkler [a,b]

[a] Siemens Corporate Technology – Learning Systems, Otto-Hahn-Ring 6, 81739 Munich, Germany
[b] Technical University of Munich – Department of Informatics, Boltzmannstr. 3, 85748 Garching, Germany
[c] Berlin University of Technology, Machine Learning, Franklinstr. 28-29, 10587 Berlin, Germany

ARTICLE INFO

Article history:
Received 30 June 2014
Received in revised form
28 October 2014
Accepted 11 November 2014
Available online 4 April 2015

Keywords:
Multi-task learning
Recurrent neural network
Factored tensor recurrent neural network
System identification
Dynamical system

ABSTRACT

A novel dual-task learning approach based on recurrent neural networks with factored tensor components for system identification tasks is presented. The goal is to identify the dynamics of a system given few observations which are augmented by auxiliary data from a similar system. The problem is motivated by real-world use cases and a mathematical problem description is given. Further, our proposed model—the factored tensor recurrent neural network (FTRNN)—and two alternative models are introduced which are benchmarked on the cart-pole and mountain car simulations. We show that the FTRNN consistently and significantly outperformed the competing models in accuracy and data-efficiency.

1. Introduction

The dynamics of complex technical systems such as gas or wind turbines can be approximated by data-driven models, e.g. recurrent neural networks [1]. Such methods have proven to be powerful alternatives to analytical models which are not always available or inaccurate [2]. Fitting the parameters to data generally requires large amounts of operational data. However, data is a scarce resource in many applications, hence, data efficient procedures utilizing all available data are preferred. Being able to obtain a model of a dynamical system is useful for a variety of purposes, e.g. condition monitoring and model-based control. The following real-world scenario is one among many that motivated the research presented in this paper.

Consider an industrial plant that is subject to modifications over time. During normal operation, the system behavior is observed and a simulation model is trained from the collected data. As a consequence of the modifications, the plant's dynamical properties change thereby invalidating the available model. However, an accurate model is needed as soon as possible after recommissioning the plant. Given that the overall plant remains largely the same, no fundamental changes of the general structure and complexity of the dynamics are expected. Therefore, information collected prior to the plant modifications can be exploited to learn a new model with significantly fewer data from the modified plant, compared to learning a new model from scratch.

The question of how to share or transfer knowledge among multiple learning tasks dates back at least almost two decades [3]. An early work presents evidence that multi-task learning may be superior to learning multiple individual tasks separately [4]. Significant research has been conducted since to explore methods allowing to transfer knowledge in various ways across domains, typically to alleviate the lack of labeled data in a target domain by exploiting prior knowledge from a relevant source domain. In [5] this field of research was surveyed by identifying and formalizing different kinds of approaches and naming related applications. In particular, the authors distinguish multi-task and transfer learning as follows: "[…] transfer learning aims to extract the knowledge from one or more source tasks and applies the knowledge to a target task. In contrast to multitask learning, rather than learning all of the source and target tasks simultaneously, transfer learning cares most about the target task. The roles of the source and target tasks are no longer symmetric in transfer learning." Prominent applications of transfer learning, that have attracted much attention in recent years, are vision, acoustics or natural language which require learning good representations of the typically high-dimensional data for further processing. In supervised learning tasks only relatively few labels of a target task are available which motivated researchers to investigate into transfer learning

* Corresponding author at: Siemens Corporate Technology – Learning Systems, Otto-Hahn-Ring 6, 81739 Munich, Germany. Tel.: +49 89 636 55734.
E-mail address: sigurd.spieckermann.ext@siemens.com (S. Spieckermann).

http://dx.doi.org/10.1016/j.neucom.2014.11.074
0925-2312/

approaches to share representation across multiple tasks using unsupervised methods, e.g. [6,7]. To the best of our knowledge, we are the first to explore multi-task learning approaches for system identification tasks using recurrent neural networks (RNN) in the context of the above introduced problem setting. Our data is typically relatively low-dimensional and structured, e.g. we observe a system through sensors each measuring a particular aspect of the system. Hence, we do not face the kind of representation or feature learning task as it is common with natural data.

The contributions of this paper comprise the introduction, motivation, and formal definition of the problem class, the proposition and discussion of adequate neural network architectures—predominantly the factored tensor recurrent neural network (FTRNN), which is a novel architecture in the context of multi-task and transfer learning—as well as their experimental evaluation using two well-known benchmarks. In particular, we compare the model errors for the two benchmarks across a range of data ratios between the source and target system, and investigate whether the utility of upsampling the target system data in order to equalize their weight in the combined error. As a result, the FTRNN outperforms the other models in all experiments and upsampling is consistently superior to keeping the original data ratios.

The remainder of the paper is structured as follows. Section 2 contains the formalized problem definition. In Section 3 our FTRNN approach and alternative methods are presented, that allow us to share parameters, and thus structure, among the two systems. The methods are benchmarked with respect to model accuracy and data-efficiency using the frictionless cart-pole [8,9] and mountain car simulations [8] in Section 4. Section 5 concludes the results and outlines future work.

This paper is an extended and revised version of [10] published upon invitation in this journal.

2. Problem definition

Let $I:=\{1,2\}$ denote a set of identifiers for fully observable deterministic systems, which are observed in fixed time intervals τ and show similar dynamics. A system is defined by the tuple (S,A,f_i) with a state space S, an action space A, and an unknown state transition function $f_i: S\times A\to S$ describing the temporal evolution of the state.

Let D_i be a set of state transition observations $(i,s,a,s')\in D_i$ of the i-th system where each observation describes a single state transition from state $s\in S$ to state $s'\in S$ caused by action $a\in A$. Further, let $D=\bigcup_{i\in I}D_i$ denote a data set of size $|D|$ drawn from a probability distribution $\mathcal{D}$.

Let $H\subseteq\{h\,|\,h: S\times A\to S\}$ denote a hypothesis space, i.e. a set of functions that are assumed to approximate the state transition function f_i. Further, let $\mathcal{L}: S\times S\to\mathbb{R}_{\geq 0}$ denote an error measure between a predicted successor state $\hat{s}'$ and the true successor state s'. The optimal hypothesis h_i^*, i.e. the best approximation of f_i within the considered space of hypotheses, minimizes the conditional expected error $\varepsilon_i(h):=\mathbb{E}_{(j,s,a,s')\sim\mathcal{D}}[\mathcal{L}(h(s,a),s')\,|\,j=i]$ where $\mathbb{E}$ denotes the expectation operator, hence, $h_i^*=\arg\min_{h\in H}\varepsilon_i(h)$. Since $\mathcal{D}$ is generally unknown, an approximately optimal hypothesis is determined by minimizing the empirical error $\hat{\varepsilon}_D(h):=1/|D|\sum_{(\cdot,s,a,s')\in D}\mathcal{L}(h(s,a),s')$ induced by a hypothesis hon a data set D, hence, $\hat{h}_i=\arg\min_{h\in H}\hat{\varepsilon}_{D_i}(h)$.

Given sufficient data D_1, it is expected that $|\varepsilon_1(\hat{h}_1)-\varepsilon_1(h_1^*)|\leq\epsilon$ for some small positive ϵ. In contrast, assuming the amount of data D_2 is insufficient, $|\varepsilon_2(\hat{h}_2)-\varepsilon_2(h_2^*)|\gg\epsilon$ and $\hat{h}_2$ may be useless. The problem addressed in this paper is to develop and assess methods that yield a better hypothesis of the insufficiently observed system through dual-task learning in order to utilize auxiliary information from D_1 as prior knowledge of the transition function f_2.

Therefore, we redefine the hypothesis space and the empirical error as follows. Let $H'\subseteq\{h\,|\,h: I\times S\times A\to S\}$ denote an extended hypothesis space, which includes the system identifier into the product space of arguments and thus approximates both transition functions by a single function. Further, let $\hat{\varepsilon}'_D(h):=1/|D|\sum_{(i,s,a,s')\in D}w_i\mathcal{L}(h(i,s,a),s')$ denote the empirical error of a hypothesis $h\in H'$ with the system specific error weight w_i. If the empirically optimal hypothesis $\hat{h}'=\arg\min_{h\in H'}\hat{\varepsilon}'_D(h)$ $(D=\bigcup_{i\in I}D_i)$ yields a smaller expected error than $\hat{h}_2$, i.e. $\varepsilon_2(\hat{h}_2)<\varepsilon_2(\hat{h}_2)$ with $\hat{h}'_2(s,a):=\hat{h}'(2,s,a)$, information from the well observed system was successfully utilized in the hypothesis search to find a better hypothesis of f_2 despite few data.

3. System identification with RNNs

In general, the state transition function of a fully observable deterministic dynamical system is described by some function $s_{t+1}=f(s_t,a_t)$. However, in practice the learning process of this function often benefits from predicting the sequence of successor states $s_{t+1},\ldots,s_{t+T}$ given a trajectory of T actions $a_t,\ldots,a_{t+T-1}$ for $T\in\mathbb{N}$, $T\geq 2$, time steps instead of predicting only a single step. For instance, if a system is observed at a high frequency, two subsequently observed states are typically very similar and a single step model would achieve low error by simply learning the identity function of the input state. In contrast, predicting a T-step trajectory will yield a large error for such a degenerate model thus forcing the learning process to find a better solution.

Let n_u and n_v denote the dimensionality of layer u and v in a neural network. Further, let $W_{vu}\in\mathbb{R}^{n_v\times n_u}$ be the weight matrix from layer u to layer v, $b_v\in\mathbb{R}^{n_v}$ be the bias vector of layer v and $\phi(\cdot)$ be an elementwise nonlinear function, e.g. $\tanh(\cdot)$.

Recurrent neural networks (RNNs) are powerful models for sequence modeling tasks. In contrast to feedforward neural networks, RNNs process their input vectors, $x_1,\ldots,x_T$, $x_t\in\mathbb{R}^{n_x}$, sequentially thereby taking its sequential structure directly into account. The input sequence is mapped to a hidden state sequence $h_1,\ldots,h_T$, $h_t\in\mathbb{R}^{n_h}$, from which the output sequence $\hat{y}_1,\ldots,\hat{y}_T$, $y_t\in\mathbb{R}^{n_y}$ is computed. Notation is slightly abused by overloading the variable h to describe the hidden state of an RNN as well as a hypothesis. A simple RNN is defined recursively for $t=1,\ldots,T$ by the following equations:

$$h_0=h_{\text{init}} \tag{1a}$$

$$h_t=\phi_h(W_{hx}x_t+W_{hh}h_{t-1}+b_h) \tag{1b}$$

$$\hat{y}_t=\phi_y(W_{yh}h_t+b_y) \tag{1c}$$

with some initial state $h_{\text{init}}\in\mathbb{R}^{n_x}$.

3.1. RNN and naïve RNN

In the context of modeling the dynamics of an open, fully observable system, a recurrent neural network may be defined by the following equations:

$$h_1=\phi_h(W_{hs}s_1+b_1) \tag{2a}$$

$$h_{t+1}=\phi_h(W_{ha}a_t+W_{hh}h_t+b_h) \tag{2b}$$

$$\hat{s}_{t+1}=\phi_s(W_{sh}h_{t+1}+b_s). \tag{2c}$$

The initial state s_1 is mapped to the hidden state of the RNN by a linear transformation followed by the nonlinear function $\phi_h(\cdot)$. The state space of a dynamical system is often real-valued and unbounded, hence, $\phi_s(\cdot)$ becomes the identity function. Fig. 1b depicts a graphical representation of the RNN architecture.