ESANN 2012 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2012, i6doc.com publ., ISBN 978-2-87419-049-0.
Available from http://www.i6doc.com/en/livre/?GCOI=28001100967420.

Recurrent Neural State Estimation in Domains with Long-Term Dependencies

Siegmund Duell[1,2], Lina Weichbrodt[1,3], Alexander Hans[1,4], and Steffen Udluft[1] *

1- Siemens AG, Corporate Technology, Intelligent Systems & Control,
Otto-Hahn-Ring 6, 81739 Munich, Germany
{duell.siegmund.ext|steffen.udluft}@siemens.com

2- Berlin University of Technology, Machine Learning,
Franklinstr. 28-29, 10587 Berlin, Germany

3- Otto-von-Guericke-University Magdeburg,
P.O.Box 4120, 39016 Magdeburg, Germany

4- Ilmenau University of Technology, Neuroinformatics and Cognitive Robotics Lab,
P.O.Box 100565, 98684 Ilmenau, Germany

Abstract. This paper presents a state estimation approach for reinforcement learning (RL) of a partially observable Markov decision process. It is based on a special recurrent neural network architecture, the Markov decision process extraction network with shortcuts (MPEN-S). In contrast to previous work regarding this topic, we address the problem of long-term dependencies, which cause major problems in many real-world applications. The architecture is designed to model the reward-relevant dynamics of an environment and is capable to condense large sets of continuous observables to a compact Markovian state representation. The resulting estimate can be used as input for RL methods that assume the underlying system to be a Markov decision process. Although the approach was developed with RL in mind, it is also useful for general prediction tasks.

1 Introduction

Reinforcement learning (RL) [1] is the machine learning approach to the optimal control problem. Instead of designing the control strategy, RL learns it from actual observations of the system to be controlled. Combined with powerful function approximators like neural networks, impressive results could be achieved [2–4]. The system is usually assumed to be a Markov decision process (MDP) $M := (\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R})$, where $\mathcal{S}$ and $\mathcal{A}$ denote the state and action spaces, respectively, $\mathcal{P} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \mapsto [0, 1]$ the transition probabilities, i.e., the probability of entering a successor state s' by executing an action a in state s, and $\mathcal{R} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \mapsto \mathbb{R}$ the reward function assigning a transition its immediate utility. The aim of RL is to derive a policy $\pi : \mathcal{S} \mapsto \mathcal{A}$ mapping each state to an action that maximizes the return, i.e., the sum of (discounted) future rewards. A central characteristic of an MDP is the Markov property, which states that the probability of reaching a certain successor state s_{t+1} depends on the current state s_t and action a_t only.

*This work has been supported by the German Ministry for Education and Science (BMBF) under grand "ALICE" (Autonomous Learning in Complex Environments).

ESANN 2012 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2012, i6doc.com publ., ISBN 978-2-87419-049-0.
Available from http://www.i6doc.com/en/livre/?GCOI=28001100967420.

However, in many real-world control problems the current state is not fully observable. Instead, only an observation $z_t \in \mathcal{Z}$ can be used as source of information about the true current state s_t, rendering the MDP into a partially observable Markov decision process (POMDP) $M' := (\mathcal{S}, \mathcal{Z}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \mathcal{O})$. In addition to the components of an MDP, a POMDP contains an observation space $\mathcal{Z}$ and an (usually unknown) observation function $\mathcal{O} : \mathcal{S} \times \mathcal{A} \mapsto \mathcal{Z}$, describing the mapping from state-action pairs to observations.

The paper introduces the Markov decision process extraction network with shortcuts (MPEN-S), a special neural architecture that aims at constructing a high-quality estimate of the current Markovian state s_t. Like previous recurrent neural network approaches [5,6], it reconstructs a Markovian state representation in a designated hidden layer. This reconstruction can then be used as input to RL methods relying on the Markov property.

While previous work [5,6] always assumed that the estimation quality of neural networks is not affected by long-term dependencies, experiments have shown a significant loss in estimation quality. The MPEN-S overcomes these restrictions by introducing shortcut connections. Other approaches also addressing the problem of long-term dependencies, however focusing on prediction performance only, include the NARX topology [7], the LSTM approach by Hochreiter [8], and the EBPTT by Boné [9].

2 State Estimation from a History of Observations

Most state-estimation approaches rely on Takens's theorem [10], which states that a sufficient number of past time slices contain all information necessary to estimate a Markovian state. This can be accomplished by simply accumulating a sufficient number of past time slices into a single state. However, in most cases the number of variables cause this naïve approach to be impractical. To overcome this problem, a neural bottleneck network (e.g., [11]) can be used to condense relevant information into a compact state representation with decorrelated state variables. To account for the time invariance of the dynamics, the bottleneck approach can further be improved by using a recurrent neural network, which avoids overfitting by reducing the number of free parameters. Recurrent state estimators show remarkable performance in various industrial applications [2].

Despite the advantages of recurrent neural networks, there have been concerns regarding their capability to model long-term dependencies [12]. In a system exhibiting long-term dependencies the system's output at time T is dependent on the input at time $t \ll T$ [7]. The problem was discovered by Mozer [13], who found recurrent networks to be unable to capture global effects in classical music. The main reason for this effect are vanishing gradients in gradient-based learning methods [7, 14]. Long-term dependencies occur in many time series, for instance from technical systems or financial data.

ESANN 2012 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2012, i6doc.com publ., ISBN 978-2-87419-049-0.
Available from http://www.i6doc.com/en/livre/?GCOI=28001100967420.

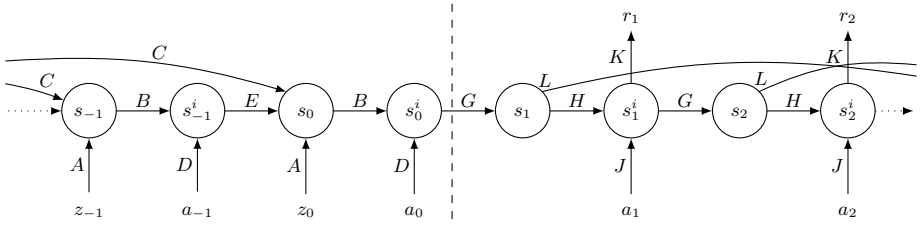

Fig. 1: The Markov decision process extraction network with shortcuts (MPEN-S) consists of a past (left) and a future (right) network part. The input variables are split into two groups: actions a_t are controllable by the RL agent, z_t denote observable variables from the environment. Only the future network part has outputs r_t (rewards). State transitions are modeled by s^i_t as well as s_t. Note that all weight matrices in the past $(A, \ldots, E)$ are different from future matrices (G, H, J, K, L).

3 Topology of the MPEN-S

The MPEN-S is a neural topology based on the MPEN [6]. The MPEN is an unfolded recurrent neural network, modified to suit the requirements of an RL problem, e.g., using rewards as targets to capture the reward-relevant dynamics and different weight matrices for the past and future part of the network. For the MPEN-S we introduce shortcuts, defining connections of length n of non-adjacent hidden states s_t (fig. 1). In contrast to the NARX approach, there are no connections of length $n-1$, $n-2$, ... As shown in fig. 1, the shortcut connections do not jump over the current state s_0, enforcing the encoding of all future-relevant information in s_0. The state estimate is represented by the activation values of s_0.

For selecting an adequate shortcut length n, a heuristic was developed. Since the severity of the vanishing-gradient problem is correlated with the number of steps that information has to be forwarded within the network, we suggest to choose the value n that minimizes the total number of steps information has to travel from any state in the past to the current state s_0, i.e., $\sum_{i=1}^{p} \text{steps}(s_0, s_{-i}, n) \rightarrow_n \min$, where $\text{steps}(s_0, s_{-i}, n)$ gives the minimum number of steps to travel from s_{-i} to s_0, including possible shortcuts. For example, if $n = 2$, $\text{steps}(s_0, s_{-1}) = 1$, $\text{steps}(s_0, s_{-2}) = 1$, $\text{steps}(s_0, s_{-3}) = 2$, $\text{steps}(s_0, s_{-4}) = 2$. The only information required for this heuristic is the maximum number of past time slices that are assumed to influence the current state. Fig. 2 shows results from experiments with different shortcut lengths indicating that the heuristic leads to reasonable results.

4 Experiments and Results

To demonstrate the capabilities of the MPEN-S two benchmarks are used, a sequence of random numbers as well as a gas turbine simulation. We compare

ESANN 2012 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2012, i6doc.com publ., ISBN 978-2-87419-049-0.
Available from http://www.i6doc.com/en/livre/?GCOI=28001100967420.

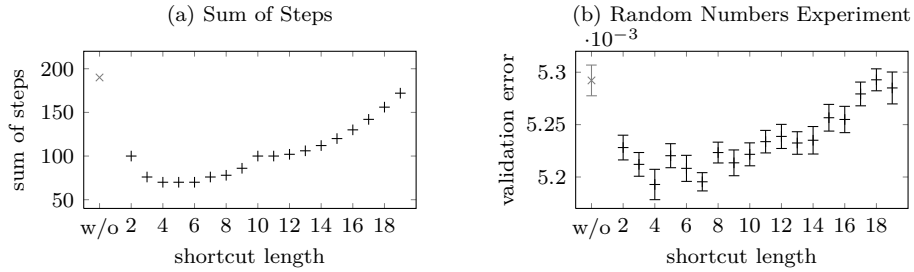

Fig. 2: Results from experiments with different shortcut lengths and a past of $p = 20$. (a) shows the sum $\sum_{i=1}^{p} \text{steps}(s_0, s_{-i}, n)$ of a network without shortcuts and networks with shortcuts of different lengths ($n \in \{4, 5, 6\}$ minimizes the sum). (b) shows the validation errors of these networks for the random numbers experiment (Sec. 4). The correlation between the sum of steps and the validation error is obvious.

the MPEN-S with shortcuts of length $n = 4$ to the MPEN, where $n = 4$ was chosen according to the heuristic. For each architecture and benchmark, we learn 10 networks using online backpropagation with a learning rate of 0.001 on 10,000 observations, of which 30% are used for validation. Evaluation is based on a further set of the same size but without noise. To judge the quality of the state estimate, represented by the activation values in s_0, the estimated states are used as inputs for a feedforward neural network (two hidden layers with 150 neurons each) whose targets are the true Markovian states of the benchmark applications. In the best case, the estimated states include all relevant information and consequently allow for a perfect mapping to the true Markovian states, i.e., the correlation between the target and output is 1.

4.1 Random Numbers Experiment

As a first experiment, sequences of equally distributed random numbers $x_i \in [0, 1]$ were used. The network with a past and a future horizon of i steps received a sequence $x_t, x_{t+1}, \ldots, x_{t+i}$ as inputs in the past part of the network. The sequence was then used as targets for the future part of the network, introducing a delay between input and corresponding output. This way, the network has to output an input given at time step t at time step $t + i$ to minimize its error. In addition, equally distributed noise $e \in [-0.05, 0.05]$ was added to the target values for training. The goal of the state estimation for the random numbers problem is to encode information about the past i random numbers.

4.2 Gas Turbine Simulation

To demonstrate the capabilities of the presented approach on a problem similar to the real-world application of our interest, we introduce a gas turbine simulation. The simulation provides a controllable variable *pilot* that affects the

ESANN 2012 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2012, i6doc.com publ., ISBN 978-2-87419-049-0.
Available from http://www.i6doc.com/en/livre/?GCOI=28001100967420.

architecture	experiment	delay	correlation	validation error
MPEN	random numbers	60	0.4	0.9
	gas turbine	30	0.988	0.0160
MPEN-S	random numbers	60	0.99	0.012
	gas turbine	30	0.995	0.0058

Table 1: Comparison of the MPEN and the MPEN-S in terms of validation error (MSE) as well as correlation between true and estimated Markovian state.

emissions and the resulting humming of the turbine. The goal of any strategy is to minimize both to meet emission requirements as well as avoid critical humming, which is reflected in the reward function. While humming reacts instantaneously, the emissions, like in real world, have a long delay and a defined blur over several steps in time. Each step, the simulator provides observations about its current state. The only additional information for the state estimation model is the maximal expected delay d of the simulation. The goal of the state estimator is to encode all information about the past d steps.

4.3 Results

Fig. 3 illustrates the correlation of the estimated and true Markovian states of the random numbers experiment ((a) and (b)) and the results of the gas turbine simulation experiment ((c) and (d)). Both benchmark results indicate that the MPEN approach is capable of estimating the Markovian states well for small delays. For longer dependencies however, the approximation quality drops significantly, while the MPEN-S can maintain its performance. Table 1 shows the average correlation and validation errors of all state variables. The numbers show that the MPEN-S improves on the MPEN both in reconstruction quality of the Markovian state (resulting in a better correlation) as well as raw forecast performance (lower validation error). The validation error is a good indicator for estimation quality, which is especially relevant for real-world applications.

5 Conclusion

A recurrent neural topology for state estimation in RL accounting for long-term effects, the MPEN-S, was introduced. Results on two benchmark applications indicate a significant improvement over previous approaches, especially for state variables with long-term delays. This is reflected in a superior validation error of the forecast as well as an improved estimation quality, especially for state variables dependent on highly delayed observables. Another important conclusion to draw from our experiments is the correlation between the validation error and estimation quality. This information is of high value, since in any real-world application one can only rely on the measure of the validation error.

ESANN 2012 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2012, i6doc.com publ., ISBN 978-2-87419-049-0.
Available from http://www.i6doc.com/en/livre/?GCOI=28001100967420.

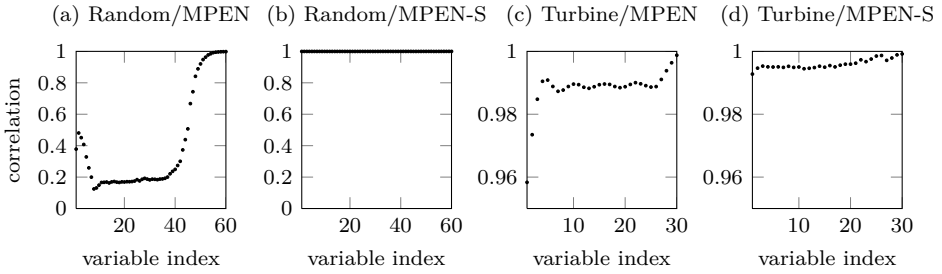

Fig. 3: Average correlations of true and estimated Markovian state. A higher variable index indicates a variable closer to the present. (a) and (b) show the estimation quality for the random numbers problem with a delay of 60 steps. (c) and (d) illustrate the estimation quality for the gas turbine simulation with a delay of 30 steps.

References

[1] R.S. Sutton and A.G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 1998.

[2] A.M. Schäfer, D. Schneegaß, V. Sterzing, and S. Udluft. A Neural Reinforcement Learning Approach to Gas Turbine Control. In *Proc. of the Int. Joint Conf. on Neural Networks*, 2007.

[3] T.C. Kietzmann and M. Riedmiller. The Neuro Slot Car Racer: Reinforcement Learning in a Real World Setting. In *Proc. of the Int. Conf. on Machine Learning and Applications*. IEEE, 2009.

[4] J. Peters and S. Schaal. Reinforcement learning of motor skills with policy gradients. *Neural Networks*, 21(4), 2008.

[5] A.M. Schäfer and S. Udluft. Solving Partially Observable Reinforcement Learning Problems with Recurrent Neural Networks. In *Workshop Proc. of the European Conf. on Machine Learning*, 2005.

[6] S. Duell, A. Hans, and S. Udluft. The Markov Decision Process Extraction Network. In *Proc. of the 18th European Symposium on Artificial Neural Networks*, 2010.

[7] T. Lin, B.G. Horne, P. Tino, and C.L. Giles. Learning long-term dependencies in NARX recurrent neural networks. *Neural Networks, IEEE Transactions on*, 7(6), 1996.

[8] S. Hochreiter, Y. Bengio, and P. Frasconi. Gradient flow in recurrent nets: the difficulty of learning long-term dependencies. In John F. Kolen and Stefan C. Kremer, editors, *A Field Guide to Dynamical Recurrent Networks*. IEEE Press, 2001.

[9] R. Boné, M. Crucianu, G. Verley, and J.-P. Asselin de Beauville. A Bounded Exploration Approach to Constructive Algorithms for Recurrent Neural Networks. *Proc. of the Int. Joint Conf. on Neural Networks*, 3, 2000.

[10] F. Takens. Detecting strange attractors in turbulence. *Dynamical Systems and Turbulence*, 898:366–381, 1981.

[11] G.E. Hinton and R.R. Salakhutdinov. Reducing the Dimensionality of Data with Neural Networks. *Science*, 313(5786):504–507, July 2006.

[12] Y. Bengio, P. Simard, and P. Frasconi. Learning long-term dependencies with gradient descent is difficult. *Neural Networks, IEEE Transactions on*, 5(2):157–166, 1994.

[13] M.C. Mozer. Induction of multiscale temporal structure. *Advances in neural information processing systems 4*, 4:275–282, 1992.

[14] P. Frasconi, M. Gori, and G. Soda. Local feedback multilayered networks. *Neural Computation*, 4(1):120–130, 1992.