Self-Motivated Development Through Rewards for Predictor Errors / Improvements

Jürgen Schmidhuber
TU Munich, Boltzmannstr. 3, 85748 Garching, München, Germany &
IDSIA, Galleria 2, 6928 Manno (Lugano), Switzerland
juergen@idsia.ch - http://www.idsia.ch/~juergen

Abstract

Even in absence of external reward, babies and scientists and others explore their world. Using some sort of adaptive predictive world model, they improve their ability to answer questions such as: what happens if I do this or that? They lose interest in both the predictable things and those predicted to remain unpredictable despite some effort. We can design curious, self-motivated robots that do the same. The author's old basic principles for doing so: a reinforcement learning (RL) controller is rewarded whenever its action sequences result in predictor errors (1990), or, more generally, predictor improvements (1991). We briefly review the history of these ideas.

Introduction

Consider a learning robotic agent with a single life which consists of discrete cycles or time steps $t = 1, 2, \ldots, T$. Its total lifetime T may or may not be known in advance. In what follows,the value of any time-varying variable Q at time t ($1 \leq t \leq T$) will be denoted by $Q(t)$, the ordered sequence of values $Q(1), \ldots, Q(t)$ by $Q(\leq t)$, and the (possibly empty) sequence $Q(1), \ldots, Q(t-1)$ by $Q(< t)$.

At any given t the robot receives a real-valued input vector $x(t)$ from the environment and executes a real-valued action $y(t)$ which may affect future inputs; at times $t < T$ its goal is to maximize future success or *utility*

$$u(t) = E_\mu \left[\sum_{\tau=t+1}^{E_\mu(T|h(\leq t))} r(\tau) \;\middle|\; h(\leq t) \right], \qquad (1)$$

where $r(t)$ is an additional real-valued reward input at time t, $h(t)$ the ordered triple $[x(t), y(t), r(t)]$ (hence $h(\leq t)$ is the known history up to t), and $E_\mu(\cdot \mid \cdot)$ denotes the conditional expectation operator with respect to some possibly unknown distribution μ from a set M of possible distributions. Here M reflects whatever is known about the possibly probabilistic reactions of the environment. For example, M may contain all computable distributions (Solomonoff 1964; 1978; Li & Vitányi 1997; Hutter 2004). Note that unlike in most previous work by others (Kaelbling, Littman, & Moore 1996; Sutton & Barto 1998), but like in much of the author's own previous work (Schmidhuber, Zhao, & Schraudolph 1997; Schmidhuber 2003), there is just one life, no need for predefined repeatable trials, no restriction to Markovian interfaces between sensors and environment (Schmidhuber 1991d), and the expected remaining lifespan $E_\mu(T \mid h(\leq t))$ appears in the utility function, that is, we take into account the possibility of extending it through appropriate actions (Schmidhuber 2003).

Intuitively, to achieve its goal the robot may profit from spending some time on building a predictive world model, by exploring its environment and learning about the consequences of its actions in particular contexts. Such activity is commonly referred to as *curiosity*. The obtained world model may later speed up or otherwise facilitate the computation of action sequences provoking external rewards.

Recent work has led to the first learning machines that are universal and optimal in various very general senses (Hutter 2004; Schmidhuber 2003). Such machines can in principle find out by themselves whether curiosity and world model construction are useful or useless in a given environment, and learn to behave accordingly.

The present paper, however, will assume *a priori* that world model building is good and should be done; here we shall not worry about the possibility that "curiosity may kill the cat." Towards this end, in the spirit of our previous work (Schmidhuber 1990; 1991c; 1991b; 1991a; Storck, Hochreiter, & Schmidhuber 1995; Schmidhuber 1997; 2002; 2002; 2004a), we split the reward signal $r(t)$ into two scalar real-valued components: $r(t) = g(r_{ext}(t), r_{int}(t))$, where g maps pairs of real values to real values, e.g., $g(a, b) = a + b$. Here $r_{ext}(t)$ denotes traditional *external* reward provided by the environment, such as pain (negative reward) in response to bumping against a wall, or pleasure (positive reward) in response to reaching some teacher-given goal state. In the context of the present paper, however, we are especially interested in $r_{int}(t)$, the internal reward, or intrinsic reward, or *curiosity* reward, which is provided whenever an internal predictive world model of the robot improves in some sense. In fact, we will often focus on the purely self-motivated case $r_{ext}(t) = 0$ for all valid t.

Our first publications on artificial curiosity (Schmidhuber 1990; 1991c) described a predictor based on a recurrent neural network (Werbos 1988; Williams & Zipser 1994; Robinson & Fallside 1987; Schmidhuber 1992; Pearlmutter 1995;

Schmidhuber 2004b) (in principle a rather powerful computational device, even by today's machine learning standards), predicting inputs $x(t)$ and $r(t)$ from the entire history of previous inputs and actions. The curiosity rewards were proportional to the predictor errors, following

Curiosity Principle 1 *(1990) Generate curiosity rewards for the reinforcement learning action selector (or controller) in response to prediction errors.*

That is, it was implicitly and optimistically assumed that the predictor will indeed improve whenever its error is high. Follow-up work (Schmidhuber 1991a; 1991b) pointed out that this approach sometimes may be inappropriate, especially in probabilistic environments: *We should not not focus on the errors of the predictor, but on its improvements.* Otherwise the system will concentrate its search on those parts of the environment where it can always get high prediction errors due to noise or randomness, or due to computational limitations of the predictor (Schmidhuber 1991a; 1991b; Storck, Hochreiter, & Schmidhuber 1995; Schmidhuber 1997; 2002; 2004a):

Curiosity Principle 2 *(1991) Generate curiosity rewards for the RL controller in response to predictor improvements.*

Our principles conceptually separate the goal (understanding the world) from the means of achieving the goal. Once the goal is formally specified in terms of an algorithm for computing curiosity rewards, based on a particular adaptive predictor, let the controller's RL mechanism figure out how to translate such rewards into world model-improving action sequences.

While the neural predictor of the first implementation of Principle 2, described in the follow-up work (Schmidhuber 1991a; 1991b), was indeed computationally less powerful than the previous one (Schmidhuber 1991c), there was a novelty, namely, an explicit (neural) adaptive model of the predictor's improvements. This meta-predictor essentially learned to predict the predictor's changes. For example, although noise details were unpredictable and led to wildly varying target signals for the predictor, in the long run these signals did not change the adaptive predictor parameters much, and the predictor of predictor changes was able to learn this. A standard RL algorithm (Watkins 1989; Kaelbling, Littman, & Moore 1996; Sutton & Barto 1998) was fed with curiosity reward signals proportional to the expected long-term predictor changes, and thus tried to maximize information gain (Fedorov 1972; Hwang *et al.* 1991; MacKay 1992; Plutowski, Cottrell, & White 1994; Cohn 1994) within the given limitations. Additional follow-up work (1995) also focused on non-deterministic worlds (Storck, Hochreiter, & Schmidhuber 1995).

More recent work (Schmidhuber 1997; 2002) addressed the problem of automatically creating predictable internal abstractions of complex spatio-temporal events, by greatly increasing the computational power of controller and predictor, implementing both as symmetric, opposing modules consisting of self-modifying probabilistic programs (Schmidhuber, Zhao, & Schraudolph 1997; Schmidhuber, Zhao, & Wiering 1997) written in a universal programming language (Gödel 1931; Turing 1936). The internal storage for temporary computational results of the programs was viewed as part of the changing environment. Each module could suggest experiments in the form of probabilistic algorithms to be executed, and make confident predictions about their effects, by betting on their outcomes, where the *"betting money"* essentially played the role of the intrinsic reward. The opposing module could reject or accept the bet in a zero-sum game, by making a contrary prediction. In case of acceptance the winner was determined by executing the algorithmic experiment and checking its outcome; the money was eventually transferred from the surprised loser to the confirmed winner. Both modules tried to maximize their money using a rather general RL algorithm designed for complex stochastic policies (Schmidhuber, Zhao, & Schraudolph 1997; Schmidhuber, Zhao, & Wiering 1997).

Some experiments in the references above focused on the pure developmental approach, where $r_{ext}(t) = 0$ for all t, and showed that the predictive world model of a curious system can improve much faster than the one of a system based on naive (i. e., random) exploration.

Other experiments, however, also verified that the presence of curiosity reward $r_{int}(t)$ can speed up the collection of *external* reward.

Recently several researchers also implemented variants or approximations of the principles above. For example, Singh and Barto focused on an implementation within the option framework of RL (Barto, Singh, & Chentanez 2004; Singh, Barto, & Chentanez 2005), directly using prediction errors as curiosity rewards (Principle 1). Kaplan and Oudeyer also used prediction errors in conjunction with a simpler RL algorithm (Kaplan & Oudeyer 2004), and more recently have switched to Principle 2.

Machine learning techniques for both prediction and RL have substantially improved since the introduction of our curiosity principles for self-motivated development. This represents a promising opportunity for new implementations.

References

Barto, A. G.; Singh, S.; and Chentanez, N. 2004. Intrinsically motivated learning of hierarchical collections of skills. In *Proceedings of International Conference on Developmental Learning (ICDL)*. Cambridge, MA: MIT Press.

Cohn, D. A. 1994. Neural network exploration using optimal experiment design. In Cowan, J.; Tesauro, G.; and Alspector, J., eds., *Advances in Neural Information Processing Systems 6*, 679–686. Morgan Kaufmann.

Fedorov, V. V. 1972. *Theory of optimal experiments*. Academic Press.

Gödel, K. 1931. Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I. *Monatshefte für Mathematik und Physik* 38:173–198.

Hutter, M. 2004. *Universal Artificial Intelligence: Sequential Decisions based on Algorithmic Probability*. Berlin: Springer. (On J. Schmidhuber's SNF grant 20-61847).

Hwang, J.; Choi, J.; Oh, S.; and II, R. J. M. 1991. Query-based learning applied to partially trained multi-

layer perceptrons. *IEEE Transactions on Neural Networks* 2(1):131–136.

Kaelbling, L. P.; Littman, M. L.; and Moore, A. W. 1996. Reinforcement learning: a survey. *Journal of AI research* 4:237–285.

Kaplan, F., and Oudeyer, P.-Y. 2004. Maximizing learning progress: an internal reward system for development. In Iida, F.; Pfeifer, R.; Steels, L.; and Kuniyoshi, Y., eds., *Embodied Artificial Intelligence*. Springer. 259–270.

Li, M., and Vitányi, P. M. B. 1997. *An Introduction to Kolmogorov Complexity and its Applications (2nd edition)*. Springer.

MacKay, D. J. C. 1992. Information-based objective functions for active data selection. *Neural Computation* 4(2):550–604.

Pearlmutter, B. A. 1995. Gradient calculations for dynamic recurrent neural networks: A survey. *IEEE Transactions on Neural Networks* 6(5):1212–1228.

Plutowski, M.; Cottrell, G.; and White, H. 1994. Learning Mackey-Glass from 25 examples, plus or minus 2. In Cowan, J.; Tesauro, G.; and Alspector, J., eds., *Advances in Neural Information Processing Systems 6*, 1135–1142. Morgan Kaufmann.

Robinson, A. J., and Fallside, F. 1987. The utility driven dynamic error propagation network. Technical Report CUED/F-INFENG/TR.1, Cambridge University Engineering Department.

Schmidhuber, J.; Zhao, J.; and Schraudolph, N. 1997. Reinforcement learning with self-modifying policies. In Thrun, S., and Pratt, L., eds., *Learning to learn*. Kluwer. 293–309.

Schmidhuber, J.; Zhao, J.; and Wiering, M. 1997. Shifting inductive bias with success-story algorithm, adaptive Levin search, and incremental self-improvement. *Machine Learning* 28:105–130.

Schmidhuber, J. 1990. Making the world differentiable: On using fully recurrent self-supervised neural networks for dynamic reinforcement learning and planning in non-stationary environments. Technical Report FKI-126-90, Institut für Informatik, Technische Universität München.

Schmidhuber, J. 1991a. Adaptive curiosity and adaptive confidence. Technical Report FKI-149-91, Institut für Informatik, Technische Universität München. See also (Schmidhuber 1991b).

Schmidhuber, J. 1991b. Curious model-building control systems. In *Proceedings of the International Joint Conference on Neural Networks, Singapore*, volume 2, 1458–1463. IEEE press.

Schmidhuber, J. 1991c. A possibility for implementing curiosity and boredom in model-building neural controllers. In Meyer, J. A., and Wilson, S. W., eds., *Proc. of the International Conference on Simulation of Adaptive Behavior: From Animals to Animats*. MIT Press/Bradford Books. 222–227.

Schmidhuber, J. 1991d. Reinforcement learning in Markovian and non-Markovian environments. In Lippman, D. S.; Moody, J. E.; and Touretzky, D. S., eds., *Advances in Neural Information Processing Systems 3 (NIPS 3)*, 500–506. Morgan Kaufmann.

Schmidhuber, J. 1992. A fixed size storage $O(n^3)$ time complexity learning algorithm for fully recurrent continually running networks. *Neural Computation* 4(2):243–248.

Schmidhuber, J. 1997. What's interesting? Technical Report IDSIA-35-97, IDSIA. ftp://ftp.idsia.ch/pub/juergen/interest.ps.gz; extended abstract in Proc. Snowbird'98, Utah, 1998; see also (Schmidhuber 2002).

Schmidhuber, J. 2002. Exploring the predictable. In Ghosh, A., and Tsuitsui, S., eds., *Advances in Evolutionary Computing*. Springer. 579–612.

Schmidhuber, J. 2003. Gödel machines: self-referential universal problem solvers making provably optimal self-improvements. Technical Report IDSIA-19-03, arXiv:cs.LO/0309048, IDSIA, Manno-Lugano, Switzerland.

Schmidhuber, J. 2004a. Overview of artificial curiosity and active exploration, with links to publications since 1990. http://www.idsia.ch/˜juergen/interest.html.

Schmidhuber, J. 2004b. RNN overview, with links to a dozen journal publications. http://www.idsia.ch/˜juergen/rnn.html.

Singh, S.; Barto, A. G.; and Chentanez, N. 2005. Intrinsically motivated reinforcement learning. In *Advances in Neural Information Processing Systems 17 (NIPS)*. Cambridge, MA: MIT Press.

Solomonoff, R. J. 1964. A formal theory of inductive inference. Part I. *Information and Control* 7:1–22.

Solomonoff, R. J. 1978. Complexity-based induction systems. *IEEE Transactions on Information Theory* IT-24(5):422–432.

Storck, J.; Hochreiter, S.; and Schmidhuber, J. 1995. Reinforcement driven information acquisition in non-deterministic environments. In *Proceedings of the International Conference on Artificial Neural Networks, Paris*, volume 2, 159–164. EC2 & Cie.

Sutton, R., and Barto, A. 1998. *Reinforcement learning: An introduction*. Cambridge, MA, MIT Press.

Turing, A. M. 1936. On computable numbers, with an application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society, Series 2* 41:230–267.

Watkins, C. J. C. H. 1989. *Learning from Delayed Rewards*. Ph.D. Dissertation, King's College, Oxford.

Werbos, P. J. 1988. Generalization of backpropagation with application to a recurrent gas market model. *Neural Networks* 1.

Williams, R. J., and Zipser, D. 1994. Gradient-based learning algorithms for recurrent networks and their computational complexity. In *Back-propagation: Theory, Architectures and Applications*. Hillsdale, NJ: Erlbaum.