Optimized behavior in a robot model of sequential action

Roy de Kleijn (kleijnrde@fsw.leidenuniv.nl)
Cognitive Psychology Unit, Leiden University
Leiden Institute for Brain and Cognition

George Kachergis (george.kachergis@gmail.com)
Department of Artificial Intelligence, Radboud University, Nijmegen
Donders Institute for Brain, Cognition and Behavior, Nijmegen

Bernhard Hommel (hommel@fsw.leidenuniv.nl)
Cognitive Psychology Unit, Leiden University
Leiden Institute for Brain and Cognition

Abstract

People learn and use complex sequential actions on a daily basis, despite living in a high-dimensional environment and body. Sequential action learning is sometimes studied in cognitive psychology using button-pressing tasks such as Nissen and Bullemer's (1987) serial respone time (SRT) task. However, the SRT task only measures the speed of button presses, neglecting the rich–and difficult to control–trajectory of the arm, which can show predictive movements and other contextual effects. In this study, we evolve neural networks to learn to control a robot arm to carry out a mouse-based SRT task under conditions of differing prediction uncertainty. We replicate behaviors found in a recent human experiment, and explore ramifications for human sequence learning.

Keywords: sequential action learning; neural networks; robot arm control; evolutionary optimization

Introduction

Sequential action is one of the cornerstones of everyday human action. Most of our everyday activities, such as coffee making or driving a car, can be regarded as complex sequential actions, governed by a structured hierarchy or grammar, and yet flexibly adapted under changing circumstances. How humans perform these sequential actions has been the subject of study for at least a century. Sequential action can be represented on a symbolic (what will my next action be?) level, as well as a subsymbolic, sensorimotor (what motor parameters should I use?) level (Yamashita & Tani, 2008). Interaction effects between the two levels of representation have been observed, and integration between the two is necessary to produce smooth sequential action. Due to their embeddedness (i.e. an implementation in a physical environment), both virtual and real robots–humanoid or not–are suitable subjects for developing and investigating models of behavior in which interaction with the environment is important (see Atkeson et al. (2000) for an extensive overview). Robot paradigms have been successfully used to investigate psychological phenomena that require such embeddedness like hand-eye coordination (Kuperstein, 1988), object handling (Ito, Noda, Hoshino, & Tani, 2006), and imitation learning (Schaal, 1999). Used in the proper way, they hold promise to investigate the relation between symbolic planning of actions and the subsymbolic execution of these actions. Modern multi-action robot planning algorithms such as DARRT require proposing and evaluating action plans across the entire state-space, from the first to the last action (Barry, Hsiao, Kaelbling, & Lozano-Pérez, 2013). Although these can run sufficiently quickly on modern hardware for action sequences of a few steps, the rapidly-exploring random tree algorithm typically used by these planners takes little cue from studies of human action planning and has a long-tailed distribution of computation time, with no guaranteed rate of convergence. The aim of this study is to investigate to what degree an optimized robot model's behavior matches human behavior in a sequential learning task, and to the extent that they match, propose new human-like strategies for multi-action robot planners.

Optimization of motor control

The choice of specific motor parameters used in the execution of motor commands is influenced by several factors and constraints. A good example is the *end-state comfort effect* (Cohen & Rosenbaum, 2004), in which the grasp location of an object is a function of the expected end state of the arm. People are observed to choose a grasp that optimizes the comfort of the arm's end state. For example, if one is reaching to right a cup that is upside-down, one will first invert one's hand (thumb-down) before grasping the cup. Other optimization is seen in the form of contextual lip rounding (Daniloff & Moll, 1968), where the lips are rounded in preparation for pronouncing the /u/ sound well in advance, and bending of mouse trajectories when sequentially reaching for stimuli with a mouse cursor by predicting its future location (Kachergis, Berends, de Kleijn, & Hommel, 2014; de Kleijn, Kachergis, & Hommel, in press).

Serial response time tasks

In such a trajectory serial response time task, four squares in the corners of a computer screen are visible, and participants are asked to move the mouse cursor as quickly as possible to the square that changes color. Participants are not told that the squares change color in a deterministic sequence of length 10. However, speed-up over time compared to a random sequence suggests that the sequence is learned, at least implicitly. Furthermore, during the course of the experiment differences in strategy seem to arise. Dale, Duran, and Morehead (2012) used a trajectory serial response time task similar to Kachergis et al. (2014), with different levels of sequence complexity[1]. They distinguished three types of movement during the inter-stimulus interval: (1) no movement, waiting for the next stimulus to appear; (2) actively moving toward the next predicted stimulus; and (3) moving the mouse cursor toward the center of the screen. As sequence complexity decreased, participants were found to make larger predictive movements (i.e. movements toward the next stimulus, as measured by distance-to-target at target onset) and be more likely to have explicit sequence knowledge.

Centering strategy

An interesting effect was visible in participants *not* making predictive movements or waiting for the next stimulus. These participants were observed to move their mouse cursor to the center of the screen, equidistant from all stimuli. The authors mention that "even participants with low pattern awareness engaged in this form of behavior" (p. 204). However, preliminary analyses of earlier collected data by the authors show that it is *specifically* this group without explicit sequence awareness that engages in this strategy. Duran and Dale (2009) agree with this finding, and report that this centering strategy is likely employed to compensate for (initial) lack of sequence knowledge, making it impossible to accurately predict the next target. In those circumstances, moving the mouse cursor to a position equidistant to all alternatives would be an effective strategy.

However, it remains unclear whether implicit or explicit sequence knowledge drives this behavior, and if the two forms of knowledge interact. On the one hand, participants may have acquired sequence knowledge but are unable to verbalize the sequence. On the other hand, participants may well *think* they have acquired knowledge of the sequence, but may simply be wrong. Analyses of earlier collected data shows that the latter group is quite small, making it difficult to investigate behaviorally.

The current study

In the current study, we directly manipulated prediction quality in a sequential reaching task with a virtual robot hand controlled by an artificial neural network. The task was similar in nature to the task described by Dale et al. (2012) and Kachergis, Berends, de Kleijn, and Hommel (2016): reaching for targets that appeared or changed color in a repeating sequence of locations.

In any modeling problem using artificial neural networks, the connection weights between the artificial neurons (or units) have to be optimized. In other words, the goal is to find those connection weights that cause the artificial agent to produce the behavior that most closely approaches the required behavior as measured by a fitness or cost function determined by the researcher. One of the most popular methods for determining suitable connection weights is *backpropagation* (Rumelhart, Hinton, & Williams, 1986). The backpropagation update algorithm operates when the network is presented with an input vector, after which the output produced by the network is compared to the desired output, and network weights are then updated according to their error value, starting with the output units and working back through the network.

Evolutionary algorithms such as *neuroevolution* (e.g. Angeline, Saunders, and Pollack (1994)) can find suitable network weights not by directly calculating an error measure for each input-output pair presented to the network, but by quantifying the performance of agents controlled by the network. In its most simple form, the method of neuroevolution generates a large number of agents with randomly initialized networks and quantifies how well they perform on the required task during a fixed period of time. Next, the best performing agents are allowed to "reproduce", and are copied to the following generation in a slightly modified way (e.g. by adding random noise to the connection weights). In subsequent generations, this procedure is repeated until some predefined fitness criterion is reached. Neuroevolution is considered an efficient approach to solving reinforcement learning problems. Past studies have shown neuroevolution to be faster and more efficient than reinforcement learning methods such as Q-learning on several tasks, including robot arm control (Moriarty & Miikkulainen, 1996; Moriarty, 1997; Stanley & Miikkulainen, 2002). Evolutionary algorithms have been used to simulate a wide range of psychological phenomena, ranging from reciprocity (André & Nolfi, 2016) to selective attention (Petrosino, Parisi, & Nolfi, 2013) and category learning (Morlino, Giannelli, Borghi, & Nolfi, 2012).

Method

Task design

The task used for the virtual robots was analogous to the task described by Kachergis et al. (2016), although the sequence itself was simplified (see Figure 1. It was designed as an environment of size 50 × 50 represented as floating-point values. Over the course of one run of 500 discrete time steps, target stimuli appeared sequentially in one of the four corners of the environment (distance 10 from the environment border), following a simple repeating 1-2-3-4 sequence. In one condition, networks were provided with accurate information about the next stimulus. In a second condition, the information was not predictive of the next stimulus. In a third condition, no information about the next stimulus was provided to the network. The exact implementation is described below under *Network design*.

A virtual robot arm was to touch the target (come within a square of size 6×6 centered on the target) as quickly as possible. After touching a target, no targets were visible for 20 time steps as an inter-stimulus interval (ISI), after which the next target would appear. Every run (one network-controlled

[1] More specifically, a measure of grammatical regularity was used inverse to the first-order entropy of the sequence, as used in (Jamieson & Mewhort, 2009).

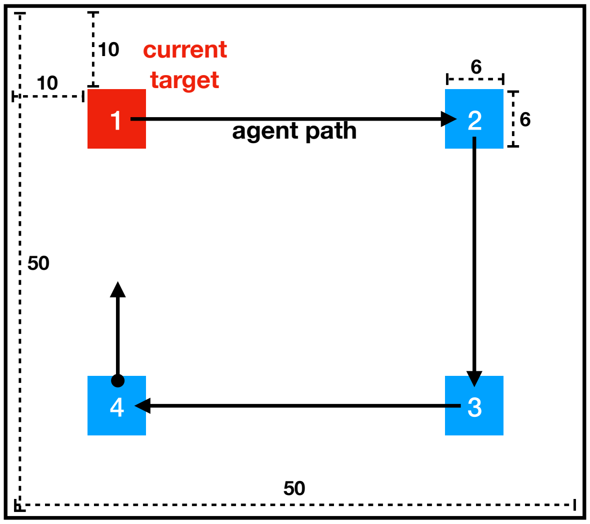


Figure 1: The simulated environment in which the arm was tasked with sequentially reaching locations 1-4, with a schematic perfectly-performing agent.

virtual robot arm performing the task for 500 time steps), the starting location was initialized to the center of the screen. During each run, the amount of targets touched and the total distance moved was calculated. Also, to encourage fast movement, a reward with decaying value was associated with each target. Rewards were initialized to value 100, decreasing by 1 with each time step. After completion of the run, network fitness was defined as

$$\textit{fitness} = \textit{touched stimuli} + \textit{total reward} - (.0001 \times \textit{dist. moved})$$

An agent with perfect prediction capability (i.e. immediately touching the stimulus that just appeared by already being in its location) would therefore be able to reach a theoretical maximum fitness score of 2525.

Network design

The virtual robot arm was controlled by a two-layer feedforward neural network with four sensory neurons, two prediction neurons, eight internal (hidden) neurons, and two motor neurons (see Figure 2). All sensory and prediction neurons were normalized in the range $[0.0, 1.0]$, with Gaussian noise sampled from $N(0, .05)$ added to the input. The two motor neurons were truncated to the range $[-2.0, 2.0]$, and allowed for movement in the two-dimensional plane. For simplicity we did not model the kinematics of an articulated effector.

The input to the two prediction neurons was constant (i.e. also present during the ISI) and represented either (1) the correct location of the next stimulus, (2) the location of one of the four stimuli, randomly chosen, or (3) a constant input of $[0.0, 0.0]$. So although in the second condition the prediction neurons were provided with the location of a stimulus, this location was not informative of the actual location of the next stimulus. These conditions will be referred to as accurate prediction, random prediction, and no prediction, respectively.

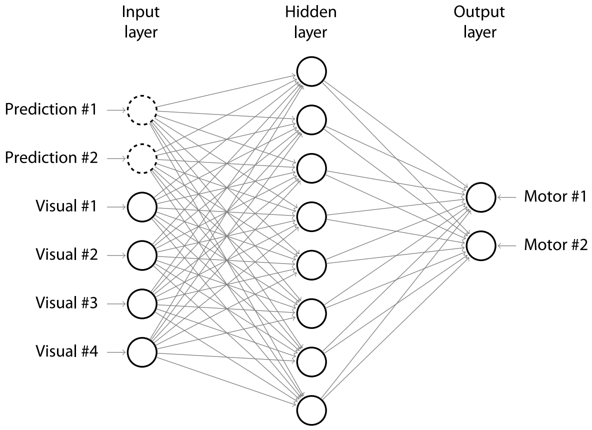


Figure 2: Two-layer feedforward network architecture used. Six input units (two prediction units and four sensory units), eight hidden units, and two output units controlled the virtual robot arm.

The output O_j of a hidden or motor neuron j was determined by the sigmoid activation function

$$O_j = \frac{1}{1 + \exp(-\sum_{i=1}^{N} w_{ij} O_i - b_j)} \qquad (1)$$

in which N represents the number of input neurons i, O_i their output, w_{ij} the connection weight from i to j, and b_j the bias. Of the four sensory neurons, two were used for sensing the target, and two for sensing the location of the agent.

Evolution of the network

Network weights were optimized using a neuroevolution algorithm using a direct encoding scheme (i.e. there was a one-to-one mapping of genotype to phenotype) similar to Nolfi, Parisi, and Elman (1994). Although direct encoding schemes have been criticized for being biologically implausible (Nolfi & Parisi, 2002), and having difficulties with scalability[2], direct encoding provided a good trade-off between simplicity and performance for the relatively simple networks used in this study. The initial population consisted of 100 networks with weights uniformly random $\in [-2.0, 2.0]$. For each subsequent generation, the twenty networks with the highest fitness value were allowed to reproduce by generating four copies each, with Gaussian noise sampled from $N(0, .3)$ added to the network weights. In addition, each of the twenty best networks was kept unmodified and added to the next generation, keeping the population size a constant 100. All simulations were run 30 times per condition, so a total of 90 simulations were run.

[2] The search space in direct encoding schemes increases exponentially with network size.

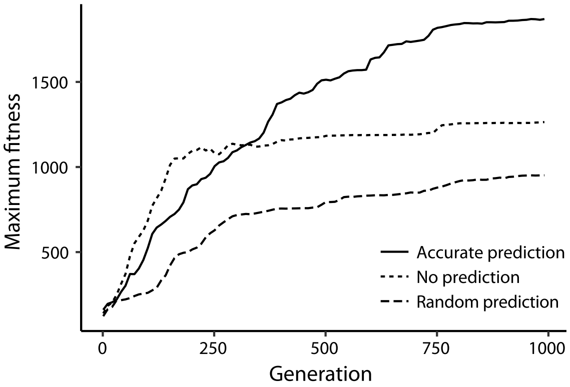


Figure 3: Networks with accurate prediction attained higher maximum fitness than networks with no prediction or random prediction. These networks evolved to make efficient use of the information from the prediction neurons. Displayed are means over 30 simulations per condition.

Results

Maximum fitness of the networks differed between conditions, $F(2, 87) = 9.29$, $p < .001$, $\eta^2_G = .176$. Post-hoc pairwise *t*-tests showed that networks with accurate predictions fed into the prediction neurons developed a higher maximum fitness (M = 1868) than networks with no prediction (M = 1262), $t(58) = 2.76$, $p = .008$, $d = .72$, and than networks with random prediction (M = 947), $t(58) = 4.12$, $p < .001$, d = 1.08. These differences remained significant after Holm-Bonferroni correction.

Figure 3 shows the evolution of fitness over time. Although the networks with no prediction evolved somewhat faster than networks with accurate prediction, maximum fitness leveled off after ∼250 generations. For the networks with accurate prediction the network weights evolved slower, but surpassed the fitness of the non-predicting networks after 320 generations and continued to increase. Networks with random prediction evolved slower overall, and attained lowest maximum fitness.

Centering behavior differed between conditions, $F(2, 86) = 8.09$, $p < .001$, $\eta^2_G = .158$. Post-hoc pairwise *t*-tests showed that networks with accurate prediction spent a smaller proportion of ITI time in the center 10×10 units (M = .195) than both networks with no prediction (M = .415), $t(57) = 4.64$, $p < .001$, $d = 1.23$, and networks with random prediction (M = .340), $t(58) = 2.96$, $p = .004$, $d = .778$. These differences remained significant after Holm-Bonferroni correction. The networks with no prediction and random prediction did not differ significantly, $p = .277$. Results are shown in Figure 4.

Movement across the environment is displayed in Figure 5. The networks with random prediction (Figure 5b) learned that the information provided was not informative, and reached their maximum fitness by returning to the center of the environment after touching each stimulus, whereas networks with accurate prediction (Figure 5a) moved toward the next target, waiting for it to appear.

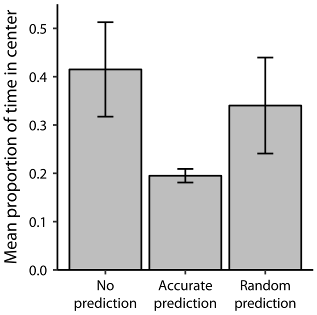
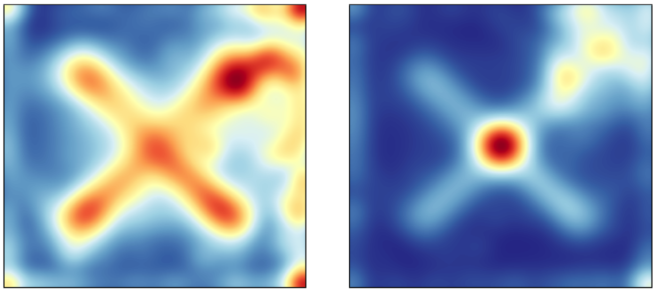


Figure 4: The mean proportion of ITI time spent in the center of the screen for all three conditions. Networks with accurate prediction spent less time in the center. Error bars indicate 95% CI.



(a) In the condition with accurate prediction, position density is clustered around the stimuli, indicating active movement toward stimuli.

(b) With random prediction, the networks evolve to produce centering behavior. Most time is spent in a position equidistant to all targets.

Figure 5: Density heat map showing the relative amount of time spent across locations in the accurate prediction and random prediction conditions, ranging from blue (little time spent) to red (most time spent).

Discussion

We investigated the behavior found in earlier work by Duran and Dale (2009), Dale et al. (2012), and de Kleijn et al. (in press). These studies describe a centering behavior in which participants moved their mouse to the center of the screen under some circumstances in sequential action learning tasks. Preliminary analysis of earlier collected data shows that this behavior seems to be related to the quality of the action plan, or the capability to predict the next stimulus. This also makes sense on a theoretical level, as a central position that is equidistant to all possible stimuli is optimal under maximum uncertainty.

In the current study we evolved artificial neural networks that controlled a robotic arm, with a task analogous to the trajectory serial response time task given to humans in de Kleijn et al. (in press) and Kachergis et al. (2014). In one condition, an accurate prediction of the next stimulus was provided to the network as part of the input. In the second condition, the input given was randomly determined, and unrelated to the next stimulus. In a third condition, input to the prediction neurons was kept constant at zero. Under the last two conditions, centering behavior developed, with the networks that were provided random input and networks that were given no input developing the same centering strategy as human participants in de Kleijn et al. (in press) that had not developed explicit sequence knowledge. In summary, we showed that centering behavior evolved in a robotic arm controlled by an artificial neural network in proportion to the unpredictability of the next stimulus. This result mirrors the centering strategy adopted by people under conditions of uncertainty, as described in de Kleijn et al. (in press).

Future research could shed light on the differences between the random prediction condition and the no prediction condition. From our results, it seems that performance was worse under the random prediction condition (although not significantly so), and developed more slowly. Apparently, the networks had trouble ignoring the dynamic, but uninformative input. In comparative studies with human participants, it would be interesting to distinguish between participants who *know* that they are unaware of the sequence (no prediction), and participants who are actively, but unsuccessfully, trying to predict the sequence (random, or at least partly incorrect prediction). We suggest that participants choose to deploy strategies on the basis of their interpretation of the task: e.g., whether it is deterministic or probabilistic, and to what extent they believe they are able to learn any regularities. It may be fruitful in future studies of human sequential action learning to inform the participants of the statistical nature of the sequences in order to discover how they choose strategies. Such knowledge would inform the creation of human-like multi-action planners, enabling our future robotic partners to interact with us more predictably.

Acknowledgments

The preparation of this work was supported by the European Commission (EU Cognitive Systems project ROBOHOW.COG; FP7-ICT-2011).

References

André, J. B., & Nolfi, S. (2016). Evolutionary robotics simulations help explain why reciprocity is rare in nature. *Scientific Reports*, *6:32785*.

Angeline, P. J., Saunders, G. M., & Pollack, J. B. (1994). An evolutionary algorithm that constructs recurrent neural networks. *IEEE Transactions on Neural Networks*, *5*, 54-65.

Atkeson, C. G., Hale, J. G., Pollick, F., Riley, M., Kotosaka, S., Schaul, S., ... Kawato, M. (2000). Using humanoid robots to study human behavior. *IEEE Intelligent Systems and their Applications*, *15*(4), 46-56.

Barry, J., Hsiao, K., Kaelbling, L. P., & Lozano-Pérez, T. (2013). Manipulation with multiple action types. In J. P. Desai, G. Dudek, O. Khatib, & V. Kumar (Eds.), *The 13th International Symposium on Experimental Robotics* (pp. 531–545). Heidelberg: Springer International.

Cohen, R. G., & Rosenbaum, D. A. (2004). Where grasps are made reveals how grasps are planned: Generation and recall of motor plans. *Experimental Brain Research*, *157*, 486–495.

Dale, R., Duran, N. D., & Morehead, J. R. (2012). Prediction during statistical learning, and implications for the implicit/explicit divide. *Advances in Cognitive Psychology*, *8*, 196–209.

Daniloff, R., & Moll, K. (1968). Coarticulation of lip rounding. *Journal of Speech and Hearing Research*, *11*, 707–721.

de Kleijn, R., Kachergis, G., & Hommel, B. (in press). Predictive movements and human reinforcement learning of sequential action. *Cognitive Science*.

Duran, N. D., & Dale, R. (2009). Predictive arm placement in the statistical learning of position sequences. In *Proceedings of the 31st Annual Meeting of the Cognitive Science Society* (pp. 893–898). Amsterdam.

Ito, M., Noda, K., Hoshino, Y., & Tani, J. (2006). Dynamic and interactive generation of object handling behaviors by a small humanoid robot using a dynamic neural network model. *Neural Networks*, *19*, 323-337.

Jamieson, R. K., & Mewhort, D. J. K. (2009). Applying an exemplar model to the serial reaction-time task: Anticipating from experience. *Quarterly Journal of Experimental Psychology*, *62*, 1757-1783.

Kachergis, G., Berends, F., de Kleijn, R., & Hommel, B. (2014). Trajectory effects in a novel serial reaction time task. In *Proceedings of the 36th annual conference of the cognitive science society* (pp. 713–718). Québec, QC.

Kachergis, G., Berends, F., de Kleijn, R., & Hommel, B. (2016). Human reinforcement learning of sequential action. In *Proceedings of the 38th annual conference of the cognitive science society* (pp. 193–198). Philadelphia, PA.

Kuperstein, M. (1988). Neural model of adaptive hand–eye coordination for single postures. *Science*, *239*, 1308–1311.

Moriarty, D. E. (1997). *Symbiotic evolution of neural networks in sequential decision tasks*. Unpublished doctoral dissertation, University of Texas at Austin.

Moriarty, D. E., & Miikkulainen, R. (1996). Efficient reinforcement learning through symbiotic evolution. *Machine Learning*, *22*, 11–32.

Morlino, G., Giannelli, C., Borghi, A., & Nolfi, S. (2012). Category learning through action: A study with human and artificial agents. *Cognitive Processing*, *13*, 47-48.

Nissen, M. J., & Bullemer, P. (1987). Attentional requirements of learning: evidence from performance measures. *Cognitive Psychology*, *19*, 1–32.

Nolfi, S., & Parisi, D. (2002). Evolution of artificial neural networks. In M. A. Arbib (Ed.), *Handbook of brain theory and neural networks* (pp. 418–421). Cambridge, MA: MIT Press.

Nolfi, S., Parisi, D., & Elman, J. L. (1994). Learning and evolution in neural networks. *Adaptive Behavior*, *3*, 5-28.

Petrosino, G., Parisi, D., & Nolfi, S. (2013). Selective attention enables action selection: evidence from evolutionary robotics experiments. *Adaptive Behavior*, *21*, 356-370.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, *323*, 533–536.

Schaal, S. (1999). Is imitation learning the route to humanoid robots? *Trends in Cognitive Sciences*, *3*, 233-242.

Stanley, K. O., & Miikkulainen, R. (2002). Evolving neural networks through augmenting topologies. *Evolutionary Computation*, *10*, 99–127.

Yamashita, Y., & Tani, J. (2008). Emergence of functional hierarchy in a multiple timescale neural network model: A humanoid robot experiment. *PLOS Computational Biology*, *4(11): e1000220*.