Sparse Coding of Learned State Representations in Reinforcement Learning

Jacob Rafati & David C. Noelle
(jrafatiheravi@ucmerced.edu, dnoelle@ucmerced.edu)
Computational Cognitive Neuroscience Laboratory
University of California, Merced
5200 North Lake Road; Merced, CA 95343 USA

Abstract

Temporal Difference (TD) Learning is a leading account of the role of the dopamine system in reinforcement learning. TD Learning has been shown to fail to learn some fairly simple control tasks, however, challenging this explanation of reward-based learning. We conjecture that such failures do not arise in the brain because of the ubiquitous presence of lateral inhibition in the cortex, producing sparse distributed internal representations that support the learning of expected future reward. We provide support for this position by demonstrating the benefits of learned sparse representations for two problematic control tasks: mountain car and acrobat.

Keywords: reinforcement learning; sparse coding; dopamine; lateral inhibition; value function approximation

Background

A class of reinforcement learning algorithms called *Temporal Difference (TD) Learning* succeeds at identifying good *policy functions*, mappings from the state of the agent to selected actions, by simultaneously learning a *value function*, mapping states to *values*, each value being an estimate of the expected future reward (Sutton & Barto, 1998). Interestingly, there are some fairly simple reinforcement learning problems for which TD Learning has been demonstrated to fail (Boyan & Moore, 1995). Failures arise when the state space is large and complex, so the value function must be learned by a nonlinear function approximator, such as a multi-layer artificial neural network. One issue is that the feedback used to learn the value function offers a *moving target*, making it difficult for learning to converge. Another issue is that value function approximators are typically biased toward continuity, with similar states tending to have similar values. In some cases, this bias can make an improvement in the value estimate of one state cause an "unlearning" of information about similar states. Recent reinforcement learning systems, using deep learning methods to approximate the value function, typically address these problems by integrating feedback over many trials before updating the value function, introducing additional memory needs and potentially slowing learning (Silver et al., 2016).

An alternative approach is to encode states using *sparse distributed representations* — high dimensional vectors in which most of the elements are zero. When used as input to a value function approximator, such representations have been shown to support learning convergence (Sutton, 1996). Sparse coding limits the impact of feedback for a state to only very similar states, reducing problems of "unlearning" as the value function chases a moving target. A major drawback of this approach, however, is the need to engineer a sparse state representation for the specific learning problem at hand.

TD Learning has enjoyed substantial success in accounting for learning in the brain, focusing on the role of dopamine in synaptic plasticity (Montague, Dayan, & Sejnowski, 1996). This raises the question of how the brain addresses value function approximation problems. We propose that an important part of the answer involves the ubiquity of lateral inhibition in cortex, supporting the learning of internal sparse distributed representations of agent states. Lateral inhibition in cortex has been shown to approximate a k-Winners-Take-All (kWTA) dynamic, in which activity is suppressed in all neurons in an area except those that are most active (O'Reilly, 2001). We have explored this conjecture by investigating the TD Learning of problematic control problems, using a multi-layer perceptron as a value function approximator, with a kWTA constraint on learned hidden layer representations. We have previously reported performance on the *puddle world* task (Rafati & Noelle, 2015). Here, we report on *mountain car* and *acrobat*.

General Methods

In each learning problem, the agent state was captured by a small set of real variables. Each was encoded as a vector, with each dimension assigned a preferred variable value and the preferences spread uniformly across the variable's range. For a given state variable value, each vector element was set to a magnitude in $[0, 1]$ based on its preferred value, determined by a Gaussian centered at the state variable value.

The vectors encoding the state variables were input to an artificial neural network value function approximator. The network had one output for each possible action, and each linear output unit was trained to estimate the expected future reward for being in the given state and taking the action associated with that output (i.e., $\hat{Q}(s, a)$). The network was trained using the SARSA TD Learning algorithm (Sutton & Barto, 1998), with error only applied to the output for the action taken. The most active output unit determined the action choice, but an ε-greedy strategy was used to ensure exploration. The agent received a reward of -1 on each time step until the goal was achieved, producing a reward of 0 and ending the trial.

The hidden units used a logistic sigmoid activation function. Importantly, they all had their activations forced to zero except the k units with the greatest net inputs. The k parameter was 10% of the number of hidden units. Only connection weights associated with active hidden units were updated during learning. Performance was compared to that of a network

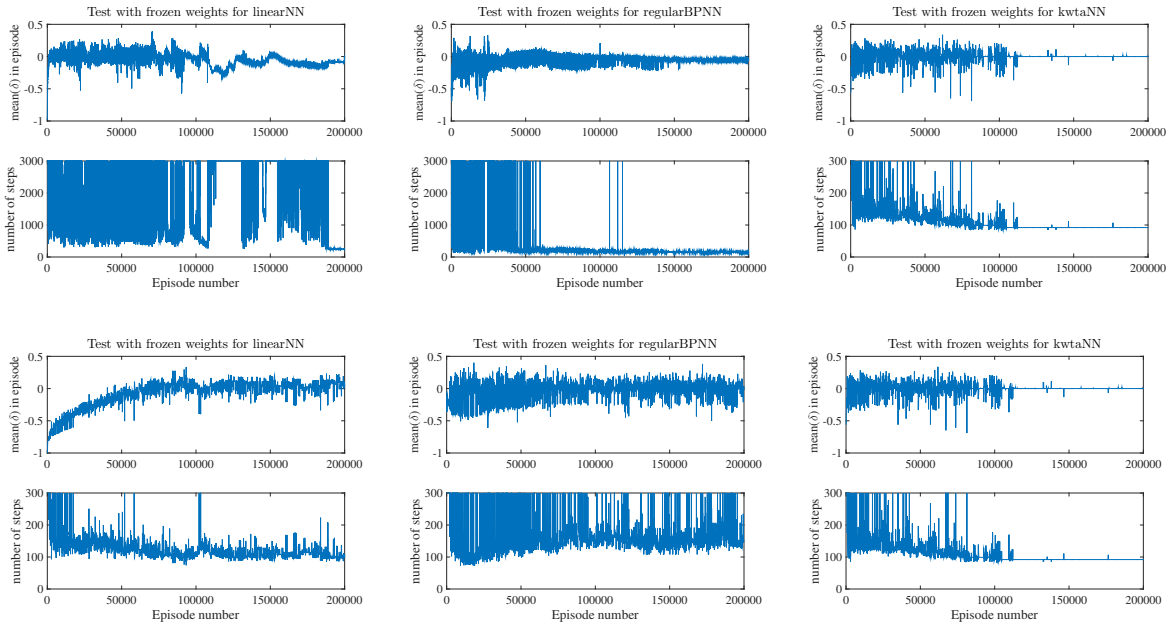

Figure 1: Mountain Car (top) and Acrobat (bottom) Learning Performance

without the kWTA constraint (i.e., a standard backpropagation network), as well as to a linear network with no hidden layer. Connection weights were initialized to uniformly sampled values from $[-0.05, 0.05]$, and the learning rate was 0.002.

Mountain Car

In the *mountain car* control problem, the agent was a car that could perform three actions: forward throttle, neutral, and backward throttle. The state of the agent involved two variables: the car location and the car velocity. The goal was to reach the top of a mountain, ahead, but the car could not generate enough thrust to drive straight there. It had to learn to first back away, up an adjacent hill, in order to produce a gravitational assist when subsequently moving forward.

The value function approximator encoded each state variable over a 61-dimensional vector, producing 122 inputs, and used 2604 hidden units ($k = 260$).

Figure 1 (top) shows representative performance over training trials. Note the stable learning of kWTA (TD error, δ, at 0) and the robust efficient solution that is found (steps).

Acrobat

In the *acrobat* control problem, the agent controlled the torque on the central joint of a double pendulum, chosing one of three actions: none, clockwise, and counter-clockwise. The state had three variables: the two joint angles and the corresponding rotational velocities. The goal was to bring the tip of the pendulum above a threshold, well above the base joint.

The value function approximator encoded each state variable over a 21-dimensional vector, producing 84 inputs, and used 8400 hidden units ($k = 840$).

Figure 1 (bottom) shows representative performance over training trials. Note the stable learning of kWTA (TD error, δ, at 0) and the robust efficient solution that is found (steps).

References

Boyan, J. A., & Moore, A. W. (1995). Generalization in reinforcement learning: Safely approximating the value function. In G. Tesauro, D. S. Touretzky, & T. K. Leen (Eds.), *Advances in neural information processing systems 7* (pp. 369–376). Cambridge, MA: MIT Press.

Montague, P. R., Dayan, P., & Sejnowski, T. J. (1996). A framework for mesencephalic dopamine systems based on predictive hebbian learning. *Journal of Neuroscience*, *16*, 1936-1947.

O'Reilly, R. C. (2001). Generaliztion in interactive networks: The benefits of inhibitory competition and Hebbian learning. *Neural Computation*, *13*, 1199–1242.

Rafati, J., & Noelle, D. C. (2015). Lateral inhibition overcomes limits of temporal difference learning. In D. C. Noelle et al. (Eds.), *Proceedings of the 37th annual meeting of the cognitive science society*. Pasadena, CA: Cognitive Science Society.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., van den Driessche, G., ... Hassabis, D. (2016). Mastering the game of Go with deep neural networks and tree search. *Nature*, *529*, 484–489.

Sutton, R. S. (1996). Generalization in reinforcement learning: Successful examples using sparse coarse coding. In D. S. Touretzky, M. C. Mozer, & M. E. Hasselmo (Eds.), *Advances in neural information processing systems 8* (pp. 1038–1044). Cambridge, MA: MIT Press.

Sutton, R. S., & Barto, A. G. (1998). *Reinforcement learning: An introduction*. Cambridge, MA: MIT Press.