Core RL Behavior Suite: `bsuite` report

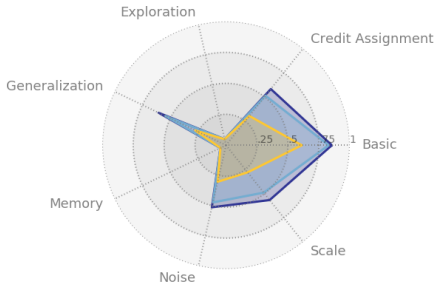
The *Core RL Behavior Suite*, or `bsuite` for short, is a collection of carefully-designed experiments that investigate core capabilities of a reinforcement learning (RL) agent. The aim of the `bsuite` project is to collect clear, informative and scalable problems that capture key issues in the design of efficient and general learning algorithms and study agent behaviour through their performance on these shared benchmarks. This report provides a snapshot of performance on `bsuite2019`, obtained by running the experiments from `github.com/deepmind/bsuite` [5].

1 Agent definition

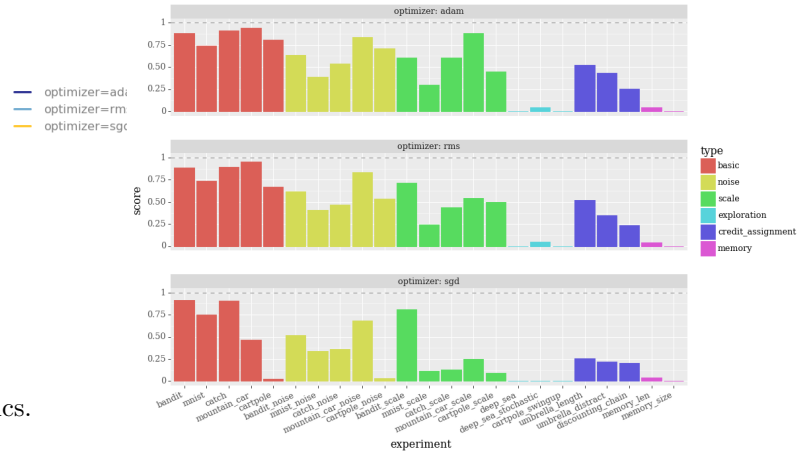
In this experiment, all agents correspond to different instantiations of the DQN agent [4], as implemented in `github.com/deepmind/bsuite/baselines/dqn`, and they were trained on `bsuite2019`. We compare three different optimizers: Adam [3], RMSProp [6], and vanilla SGD [2], as implemented in TensorFlow [1]. For each optimizer we tuned the learning rate for each of the bsuite categories (basic RL, scale, noise, memory, generalization, exploration and credit assignment), by picking the best value among {1e-1, 1e-2, 1e-3}. We used the default values from `bsuite2019`for the other hyper-parameters of DQN.

2 Summary scores

Each `bsuite` experiment outputs a summary score in [0,1]. We aggregate these scores by according to key experiment type, according to the standard analysis notebook. A detailed analysis of each of these experiments may be found in a notebook hosted on Colaboratory: `ADD-LINK-HERE`.



(a) Radar plot for snapshot of agent characteristics.



(b) Examining score by individual experiment.

Figure 1: Summary output from the `bsuite2019` experiments.

3 Results commentary

Both RMSProp and Adam perform better than SGD across the board. Adam slightly outperforms RMSProp in terms of its robustness to scale and on the credit assignment category.

References

[1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[2] Jeannette Kiefer and Jacob Wolfowitz. Stochastic estimation of the maximum of a regression function. 1952.

[3] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[4] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

[5] Ian Osband, Yotam Doron, Matteo Hessel, John Aslanides, Hado Van Hasselt, Eren Sezener, Andre Saraiva, Tor Lattimore, Csaba Szepezvari, Satinder Singh, Benjamin Van Roy, Richard Sutton, and David and Silver. Core RL behaviour suite. 2019.

[6] T. Tieleman and G. Hinton. Lecture 6.5—RmsProp: Divide the gradient by a running average of its recent magnitude. COURSERA: Neural Networks for Machine Learning, 2012.