MAP Propagation Algorithm: Faster Learning with a Team of Reinforcement Learning Agents

Stephen Chung

Nearly all state-of-the-art deep learning algorithms rely on error backpropagation, which is generally regarded as biologically implausible. An alternative way of training an artificial neural network is through treating each unit in the network as a reinforcement learning agent, and thus the network is considered as a team of agents. As such, all units can be trained by REINFORCE, a local learning rule modulated by a global signal that is more consistent with biologically observed forms of synaptic plasticity. Although this learning rule follows the gradient of return in expectation, it suffers from high variance and thus the low speed of learning, rendering it impractical to train deep networks. We therefore propose a novel algorithm called MAP propagation to reduce this variance significantly while retaining the local property of the learning rule. Experiments demonstrated that MAP propagation could solve common reinforcement learning tasks at a similar speed to backpropagation when applied to an actor-critic network. Our work thus allows for the broader application of teams of agents in deep reinforcement learning.

Background & Motivation

Training an ANN by REINFORCE without backprop

Though it is common to use backprop to train an artificial neural network (ANN), backprop is generally regarded as biologically implausible. Alternatively, we can **view an ANN as a team of reinforcement learning (RL) agents**, by injecting stochastic noise to each activation unit in the ANN and treating each unit as an RL agent that explores independently. For example, in an RL task, when learning the policy function parametrized by an ANN, we can view the ANN as the policies of multiple RL agents instead of a single agent:

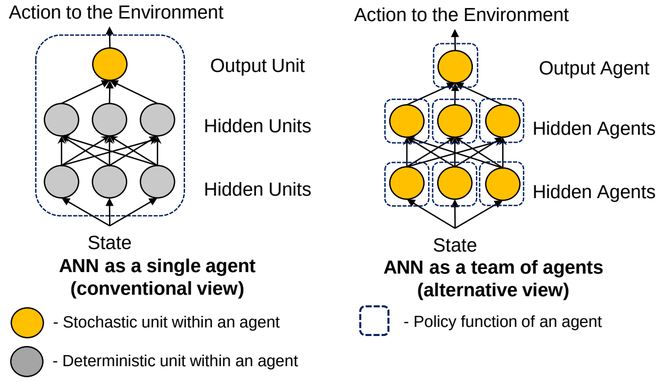


Fig 1 Illustration of viewing an ANN as a team of agents. In the alternative view, we treat each activation unit in an ANN as an agent. To each agent, all other agents are part of the black-box environment. For example, to the agents on the middle layer, the actions of agents on the first layer are considered as the state.

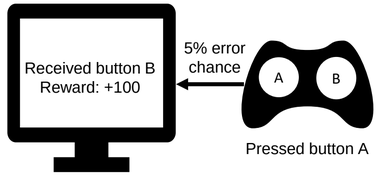
From this alternative view, we can pass the same reward from the environment to all agents and train them by REINFORCE [1]. This learning method, which we call global REINFORCE here, was proposed by Barto in 1985 [2]. Global REINFORCE gives an unbiased estimate of the gradient of reward [1,3], and is biologically plausible due to its similarity with reward-modulated STDP, a learning rule that is observed biologically. Despite these advantages, this learning method is seldom used in practice due to its large variance. Thus, the **goal of the paper is to reduce the variance of global REINFORCE** while retaining biological plausibility.

Special thanks to Andrew G. Barto, who inspired this research and provided valuable comments.

MAP Propagation

An algorithm to reduce the variance of global REINFORCE

We propose a novel algorithm called **maximum a posteriori (MAP) propagation** to reduce this variance effectively. Essentially, MAP propagation replaces the hidden agents' actions with their MAP estimates conditioned on the output agent's action, or equivalently, **minimizes the energy function of the network** while clamping the output agent's action, **before applying global REINFORCE.** The energy function is defined as -logPr(A,H|S), the negative log probability of actions of the output agent (A) and hidden agents (H) conditioned on the state (S).



How does MAP propagation reduce variance?

- Two buttons available (A & B); one with reward +100; another with reward -100;
- 5% chance that the opposite button is sent to the computer;
- You pressed button A, the computer showed it received button B and gave you a reward of +100;
- Should you press A or B more?

Button pressed = hidden agent's action
Button received by the computer = output agent's action

REINFORCE: Press A more since you pressed button A => learning in the wrong direction
What if 49% instead of 5% error chance? Wrong direction in 49% of trials!
MAP Propagation: Press B more since B is the button you most likely to have pressed

Properties of MAP propagation

- Can be derived from REINFORCE by using MAP estimate to approximate an expected term (approximation of Theorem 1);
- For normally distributed units, MAP propagation is equivalent to backprop with the reparameterization trick after minimizing the energy (Theorem 2);
- With a variant that can be applied to train critic networks (Theorem 3);
- **Can be used to train any networks with a computable energy function;**
- Biased due to approximation, but works well and converges in experiments;
- Higher computational cost due to the energy minimization phase.

Learning Method	Local learning rules	Parallel learning across layers	No symmetric feedback connections
Backprop	×	×	×
MAP propagation	✓	✓	×
Global REINFORCE	✓	✓	✓

Table 1. Comparison of the biologically plausible properties of different learning rules.

Experiment Results

- Applied MAP propagation to train an actor-critic network in four RL tasks;
- The actor-critic network is a two-hidden-layer ANN with 64 and 32 units on the first and the second hidden layer respectively;
- Significantly faster than the global REINFORCE baseline, and the learning speed is comparable to backprop;
- Also demonstrated more sophisticated exploration (e.g., no stuck in the task of mountain car).

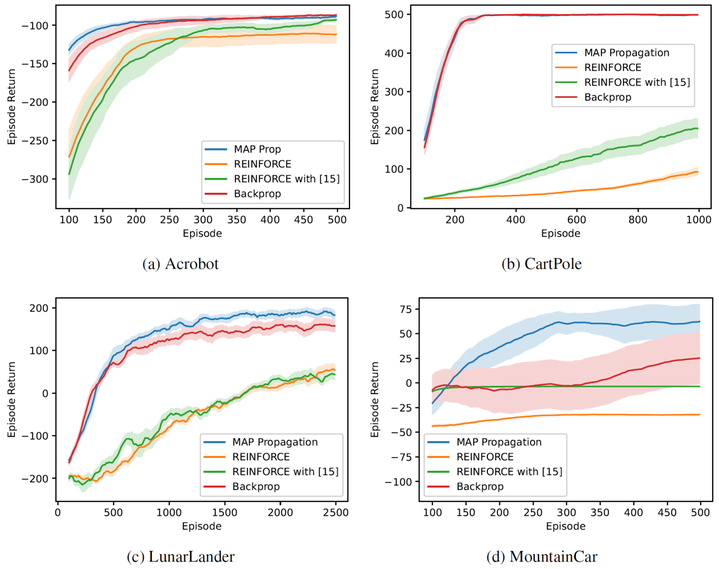


Fig 2 Running average returns over the last 100 episodes in four RL tasks.

Future Work

- Asynchronous MAP propagation that can be implemented efficiently with neuromorphic circuits;
- Different temporal resolution of agents such that the actions of agents can be extended temporally and become options;
- Investigate biological plausibility and possible neuroscience basis of MAP propagation.

References

[1] Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. Machine learning, 8(3-4):229–256, 1992.

[2] Andrew G Barto. Learning by statistical cooperation of self-interested neuron-like computing elements. Human Neurobiology, 4(4):229–256, 1985.

[3] Philip S Thomas. Policy gradient coagent networks. In Advances in Neural Information Processing Systems, pages 1944–1952, 2011.