Notes: Neural Episodic Control

esgl Hu

December 11, 2017

1 The problems of existing reinforcement learning algorithms

[1]

1. Stochastic gradient descent optimisation requires the use of small learning rates. Due to the global approximation nature of neural networks, high learning rates cause catastrophic interference. Low learning rates mean that experience can only be incorporated into a neural network slowly.

2. Environments with sparse reward signal can be difficult for a neural network to model as there may be very few instances where the reward is non-zero. This can be viewed as a form of class imbalance where low-reward samples outnumber high-reward samples by an unknown number. Consquently. the neural network disproportionaly underperforms at predicting larger rewards, making it difficult for an agent to take the most rewarding actions.

3. Reward signal propagation by value-boostrapping techniques, such as Q-learning, results in reward information being propagated one step at a time through the history of previous interactions with the environment. This can be fairly efficient if updates happen in reverse order in which the transitions occur. However, in order to train on randomly selected transitions, and, in order to further stabilise training, required the use of a slowlu updating *target network* further slowing down reward propagation.

2 Algorithms

DND: *lookup* and *write*. Performing a lookup on a DND maps a key k to an output value o:

$$o = \sum_i w_i v_i \tag{1}$$

where v_i is the ith element of the array V_a and

$$w_i = \frac{k(h, h_i)}{\sum_j (h, h_j)} \tag{2}$$

where h_i is the ith element of the array K_a and $k(x, y)$ is a kernel between vectors x and y, e.g., Gaussian or inverse kernels. for example,

$$k(h, h_i) = \frac{1}{\|h - h_i\|_2^2 + \delta} \tag{3}$$

The N-step Q-value estimate is then

$$Q^{(N)}(s_t, a_t) = \sum_{j=0}^{N-1} \gamma^j r_{t+j} + \gamma^N max_{a'} Q(s_{t+N}, a') \tag{4}$$

The classic tabular Q-learning algorithm:

$$Q_i \leftarrow Q_i + \alpha(Q^{(N)}(s, a) - Q_i) \tag{5}$$

Algorithm 1 Neural Episodic Control

$\mathcal{D}$: replay memory.
M_a: a DND for each action a.
N: horizon for N-step Q estimate.
for each episode **do**
  for $t = 1, 2, ..., T$ **do**
    Receive observation s_t from environment wirh embedding h.
    Estimate $Q(s_t, a)$ for each action a via (1) from M_a.
    $a_t \leftarrow \epsilon-$greedy policy based on $Q(s_t, a)$.
    Take action a_t, receive reward r_{t+1}.
    Append $(h, Q^{(N)}(s_t, a_t))$ to M_{a_t}.
    Append $(s_t, a_t, Q^{(N)}(s_t, a_t))$ to $\mathcal{D}$.
    Train on a random minibatch from $\mathcal{D}$.
  end for
end for

References

[1] Alexander Pritzel, Benigno Uria, Sriram Srinivasan, Adri Puigdomnech, Oriol Vinyals, Demis Hassabis, Daan Wierstra, and Charles Blundell. Neural episodic control. 2017.