Notes on: “Quantum Tensor Networks for Variational Reinforcement Learning”

1 Introduction

Reinforcement Learning (RL) is a novel field in machine learning. Some recent examples of its success are seen in DeepMind’s AlphaGo and AlphaZero, which beat the top players in both Go and Chess. This was achieved through the growth of computational resources and big data.
However, there are still many challenges for RL:

1. Traditional dynamic programming suffers from instability
2. Computation and storage complexities for Markov Decision Process/RL are combinatorially large
3. Reproducibility, reusability, and robustness is not clear, according to many researchers.

Using a variational optimization scheme, with the aid of tensor networks the paper will address these challenges.

2 Background and Preliminaries

We describe the background of MDP and tensor networks.

Notations: We denote tensors by calligraphic letters, e.g., $\mathcal{A}$. Let $\otimes$ stand for the tensor (Kronecker) product, $\times_k$ for mode-k product (a.k.a, tensor contraction, Einstein sum), $\circledast$ for the Hadamard (element-wise) product, and $\langle \cdot\ ,\ \cdot \rangle$ for inner product. We use order-N for N-way tensors, and rank[2] for the maximum number of linearly independent vectors in a tensor. Let $[n]$ denote the set $\{1, 2, \cdots, n\}$.

RL algorithms train the agent to interact with an environment, such that it chooses the sequence of actions that have the maximum expected rewards.

The equation for reward:

$$Q(\pi|s,a) = \mathbb{E}_\pi \left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \mid s_t = s, a_t = a \right]$$

R(t+k+1) denotes the direct reward for taking action a(t+k) at state s(t+k)

The Bellman equation gives the optimality condition for MDP problems

$$Q(\pi|s,a) = R^a_{s,s'} + \gamma \sum_{a'} \pi(s',a') Q(\pi|s',a'), \tag{2}$$

which expresses the expected reward at s as a summation of direct reward $R^{a'}_{s,s'}$ and discounted future reward at s'. The optimal policy is given by

$$\pi^* = \arg\max_{\pi} Q(\pi|s,a). \tag{3}$$

Tensors are algebraic objects that describe a multilinear relationship between sets of algebraic objects related to a vector space.

Tensor networks are useful for RL as they represent multilinear mapping using interconnected tensors, which allows for greater efficiency. This operation is called tensor contraction.

For tensors $\mathcal{C} = \mathcal{A} \times_n^m \mathcal{B}$, storing A and B instead of C is much more efficient for storage. Also, if two nodes are connected, that means the corresponding tensors are contracted, allowing for efficiency in that facet