Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 123 (2018) 347–353

Procedia Computer Science

www.elsevier.com/locate/procedia

8th Annual International Conference on Biologically Inspired Cognitive Architectures, BICA 2017

Grid Path Planning with Deep Reinforcement Learning: Preliminary Results

Aleksandr I. Panov[1,2], Konstantin S. Yakovlev[1,2], and Roman Suvorov[1]

[1] Federal Research Center "Computer Science and Control" of Russian Academy of Sciences, Moscow, Russia

[2] National Research University Higher School of Economics, Moscow, Russia

{pan,yakovlev,rsuvorov}@isa.ru

Abstract

Single-shot grid-based path finding is an important problem with the applications in robotics, video games etc. Typically in AI community heuristic search methods (based on A* and its variations) are used to solve it. In this work we present the results of preliminary studies on how neural networks can be utilized to path planning on square grids, e.g. how well they can cope with path finding tasks by themselves within the well-known reinforcement problem statement. Conducted experiments show that the agent using neural Q-learning algorithm robustly learns to achieve the goal on small maps and demonstrate promising results on the maps have ben never seen by him before.

Peer-review under responsibility of the scientific committee of the 8th Annual International Conference on Biologically Inspired Cognitive Architectures

Keywords: path planning, reinforcement learning, neural networks, Q-learning, convolution networks, Q-network

1 Introduction

Finding a single path on a given static grid is a well-known and well-studied problem in AI, planning and robotics communities [16] with a large variety of methods and algorithms proposed so far. Most of these algorithms rely on heuristic search in the state-space induced by the grid cells and are based on the well-known A* algorithm proposed in 1968 [4]. Among the most wide-spread algorithms that solve the task on-line, e.g. without any preprocessing, one can name IDA* [7], ARA* [8], JPS [3], Theta* [10] etc., all of which are apparently heuristic search algorithms.

At the same time recently we have witnessed a huge break-through in applying neural networks to all sorts of tasks within AI domain, including playing the game of Go [11], recognizing objects from images [13], generating images and representation learning [2], machine translation [17], speech recognition [5], etc. One common thing that can be noticed is that neural networks cope well with all sorts of tasks when sensory or image data (e.g. "raw pixels") is used as an input. Definitely square grids containing only two types of cells, i.e. traversable and

Peer-review under responsibility of the scientific committee of the 8th Annual International Conference on Biologically Inspired Cognitive Architectures
10.1016/j.procs.2018.01.054

untraversable, look like a perfect input to modern artificial neural network (NN) architectures, such as convolutional NN and residual learning NN. Path finding problem in that case can be viewed as a problem of learning which image pixels (grid cells) to identify as being part of the sought path.

In this paper preliminary results of applying reinforcement learning machinery to 2D grid-based path finding are presented. First, provided with the most commonly-used problem statement we define the appropriate machine learning problem (Section 2). Second, we suggest a variety of tools, e.g. various deep neural network architectures, to solve it (Section 3). And, finally, we evaluate their performance running a series of experiments in simulated environments (Section 4). The results of the experiments are twofold. On the one hand it is undoubtful that the proposed NN learns to plan paths, e.g. after the learning phase it is capable to solve previously unseen instances. On the other hand the NN sometimes fail to find a path. In case a path is found it is likely to be longer than the path found by A* algorithm, e.g. the shortest one. Obtained results provide an opportunity to claim that although the NN cannot be seen as an alternative to well-establishes path finding techniques so far, the application of various deep neural network architectures to path finding tasks is a perspective line of research.

2 Problem statement

Consider an agent operating on the 8-connected grid composed of blocked and unblocked cells. Blocked cells are considered to be ones (white pixels of the image) and unblocked zeroes (black pixels of the image). Given the start and goal locations, which are tied to the centers of distinct unblocked cells s and g, the task is to find a path π which is a sequence of adjacent traversable cells starting with s and ending with g, $\pi = \pi(s, g) = (s, succ(s), succ(succ(s)), \ldots, g)$. Here $succ(x)$ stands for the successor of cell x and we enforce successors to be neighboring (adjacent) cells. Cost of the move between x and $succ(x)$ equals 1 in case its a cardinal move and $\sqrt{2}$ in case its a diagonal move. The cost of the path is sum of all moves comprising π. In this work we are not limiting ourselves to finding least cost paths only but paths with smaller costs are preferable.

To transform the abovementioned path finding problem formalization, widely used in AI community, to reinforcement learning problem we utilize a framework of agent-centered search which interleaves planning and execution. Planning here means deciding which movement action (going left, right, up, down, etc.) $a_t = p_t \to p_{t+1}$ to perform next and execution stands to performing this action as well as receiving feedback from the environment, e.g. a reward that is a real number. We denote the position of the agent at time t as $p_t = (x_t, y_t)$, where x and y are grid cell coordinates. We consider only finite sequences of actions until the agent reaches the destination g or the maximal amount of steps T is achieved.

Agent has a limited field of view, denoted $s_t \in S, S = \mathbb{R}^{(2d+1)^2}$. In this work we presume that its a square window of the predefined size $(2d+1) \times (2d+1)$ (say 11×11, $d = 5$) with the agent placed in its center (so the agent is capable of seeing in all four directions). When the agent approaches a grid edge the portion of the s_t which is out of bounds is filled with ones (so the outer map is considered to be an obstacle).

The reward at time t, e.g. r_t, is calculated using the unknown to the agent function $G(s, g, t)$, where s and g are start and goal locations. This function as well as grid map, M, comprise the model of the environment $E = (M, G)$. The task of the agent operating in this environment is twofold. First agent has to learn. During the learning phase agent aims at maximizing its overall reward while acting in the environment, e.g. relocating on a grid map. Second agent has to navigate to goal location on any previously unseen grid map he is put into without having