Neural Net Based Multi-Agent Mobile Robots Control System: Practical Implementation on Different Platform

M L Patkin, G N Rogachev and N G Rogachev
Samara State Technical University, Samara, 443100, Russian Federation
E-mail: patkinm@gmail.com, gogachev@mail.ru and rogachev8@mail.ru

Abstract. Mobile robot control is a challenging task due to the need for lightning-fast decision making in a rapidly changing environment. We propose neural net with reinforcement learning based control system which can simultaneously manage mobile robots and communicate with other partners. The Actor Critic approach is used but we divide the Actor into the Communication Actor and Action Actor. Due to the fact that, unlike the classical approach, additional neural networks appeared, the question of optimization arose. We propose implementation of control system on three different platform: desktop, mobile and embedded platform with VPU acceleration. We demonstrate that we achieve similar performance and speed of low power embedded and mobile devices and full power desktop PC with modern GPU acceleration.

1. Introduction
The complication of technical systems and the stiffening of requirements for the accuracy of their handling involve to the need for continuous enhancement of control algorithms. In recent time, the approaches of decision making and intellectual control have enhance remarkable because of the use of artificial neural networks.

Reinforcement learning is a powerful approach for solving problems of control complicated systems, the main benefit of which is that it does not impose prior knowledge of the object model. The knowledge about the control object appears in the process of the agent's impact on the control object and receiving information about it. When solving a problem using the multi-agent approach, the complexity of the control algorithms is determined not only by the sophistication of the control objects but also by the degree of interaction of the agents. This fact suggests that practically unrestricted possibilities open up for improving the quality of facility management [1].

In this article, we propose an implementation of multi-agent cooperative interactions and a method for running this system on different platforms: PC, mobile device (iPhone) and embedded device (Raspberry Pi 3 with VPU acceleration Intel Movidius).

2. Approach
The classical method of the Actor-Critic assumes that political function of the Actor will be created on the basis of function of cost of the Critic [2, 3]. The Criticism function is developed according to an algorithm of TD [4, 5]. In cases with repeated agents (figure 1), especially in case of POMDP, cooperation between agents begins to have strong influence on quality of agents, starting with
information on observation which is more important for the robot, can be found. For such situations modification of an algorithm of the Actor-Critic has been executed [6].

The main idea of the changed Actor-Critic's method for a case of the interacting systems of the multiagent in division of the Actor in the Action Actor and the Communication Actor (figure 2). The actor of action, as in a classical method responsible for development of actions of control, and the Communication Actor makes the message for partners which brings into more effective management of the system of the multiagent.

Consider how the modified method described above works. Agents generate messages that join the observation vectors of the rest agents (Figure 3). Agents generate actions based on all information from observational vectors including information from other agents. During training and back propagation of errors from the end to the beginning, gradients propagate [7] from the Action Actor to the Communication Actor. Thus, Communication Actor is trained to generate only those messages that will reduce the number of errors in predicting the actions of Action Actor.

The figure 4 demonstrates red and green circles - agents, white lines stemming from them - sonars, yellow and blue circles – targets. Table 1 demonstrates architectures of neural networks of Actors and Critic where $N_a$ – number of actions and $N_m$ – length of message vector.

The robots are controlled by choosing one of three actions: "right", "forward", "left". The central idea of this experiment is that robot 1 knows the information about the position of target 2, but must catch target 1 and inverse. This formulation of the task allows the robots to be nested to accomplish the task, since the robots need to find a method or language of communication to accomplish it. Experiments were held for several environment settings: with a static target and with a target moving with a variable speed in an arbitrary direction.
Table 1. Architectures of Actors and Critic.

|   | Action Actor          | Communication Actor | Critic       |
|---|----------------------|---------------------|--------------|
| 1 | Fullyconnected: 1024 | Fullyconnected: 1024| Fullyconnected: 1024 |
| 2 | Dropout: 20%         | Dropout: 20%        | Dropout: 20% |
| 3 | RELU                 | RELU                | RELU         |
| 4 | Fullyconnected: 1024 | Fullyconnected: 1024| Fullyconnected: 1024 |
| 5 | Dropout: 20%         | Dropout: 20%        | Dropout: 20% |
| 6 | RELU                 | RELU                | RELU         |
| 7 | Fullyconnected: 1024 | Fullyconnected: 1024| Fullyconnected: 1024 |
| 8 | Dropout: 20%         | Dropout: 20%        | Dropout: 20% |
| 9 | RELU                 | RELU                | RELU         |
|10 | Fullyconnected: Na   | Fullyconnected: Nm  | Fullyconnected: 1  |

3. Implementation on different platforms

3.1. Desktop

For the desktop implementation we used Keras 1.0.7 framework for the training of the models. Training of the models and environment simulation took approximately 0.5 hour on Intel i7 2600k and Nvidia Titan Black.

3.2. Mobile device

For running the trained models, we converted it to CoreML format with coremltools 2.0 which can be used on Apples’ mobile devices iPhone SE and newer with operation system iOS 11.x and newer. We used iPhone 7 with A9 processor. Weights of a trained model were converted from fp32 number format (floating point format of real numbers taking 4 bytes of memory per number) to afp16 (floating point format of real numbers taking 2 bytes of memory per number). This causes neglectable loss in accuracy and sufficient gain in terms of final size of a model.

3.3. Embedded device: Raspberry Pi 3 and Intel Movidius stick

For many tasks, for example when controlling robots, an embedded device is required. One of the most popular and cheap devices is Raspberry Pi 3. However, the ARM processor contained in this device is not the optimal choice for launching neural networks. Therefore, we use the Intel Movidius device to running neural networks (Figure 5).
3.4. Experiments
As a result of experiments, we have shown (table 2) that neural network models work approximately the same time on both powerful PCs and mobile and embedded devices.

Table 2. Comparison of execution times of agents on different devices.

| Device name                                      | Time of execution (ms) |
|--------------------------------------------------|------------------------|
| Desktop(i7 2600k, Titan Black)                   | 0.7                    |
| iPhone 7                                         | 1                      |
| Raspberry Pi 3 with Intel Movidius               | 1.05                   |

4. Conclusion
The suggested multiagent control method requires an increased amount of computing resources relative to standard methods, however, if you use the techniques presented in this article, you can achieve similar performance on low-power devices without loss of quality.

Acknowledgments
The reported study was funded by RFBR according to the research project № 18-08-00506A and by RFBR with Samara Region according to the research project № 17- 48-630410p-a.

References
[1] Lowe R, Wu Y, Tamar A, Harb J, Abbeel O P and Mordatch I 2017 Advances in neural information processing systems pp 6379-6390
[2] Konda V R and Tsitsiklis J N 2000 Advances in neural information processing systems pp 1008-1014
[3] Bhatnagar S, Sutton R S, Ghavamzadeh M & Lee M 2009 Automatica 45(11) 2471-2482
[4] Tsitsiklis J N and Van Roy B 1997 Advances in neural information processing systems pp 1075-1081
[5] Sutton R S, McAllester D A, Singh S P and Mansour Y 2000 Advances in neural information processing systems pp 1057-1063
[6] Patkin M L and Rogachev G N 2018 Construction of multi-agent mobile robots control system in the problem of persecution with using a modified reinforcement learning method based on neural networks IOP Conference Series: Materials Science and Engineering 308 012018
[7] Gupta J K, Egorov M and Kochenderfer M (2017) Cooperative multi-agent control using deep reinforcement learning International Conference on Autonomous Agents and Multiagent Systems pp 66-83