UDC 629.114. 2

MEDVEDEV M. Yu., D. Sc. in Eng., As. Prof.,
Professor
E-mail: medvmihal@sfedu.ru

FARHOOD A. K.,
Ph.D. student
E-mail: eim@sfedu.ru

Southern Federal University, Taganrog, Russia

Received 14 May 2022

THE STUDY NUMERICAL TO DETECT COLLISIONS USING NEURAL NETWORK WITH D* ALGORITHM

A numerical study of the proposed iterative algorithm using the Matlab complex is carried out. During the study, the task of teaching a neural network to plan a route was solved in the same way as the "teacher" algorithm, for which D was chosen. The initial filtering option is selected in such a way that, when a collision occurs, a trajectory point that preceded the collision is fed into the filtered sample, i.e., in which the neural network incorrectly classified the situation. This method of filtering was not effective, because the wrong decision that led to the collision could have been made not immediately before the collision, but earlier. In this regard, the procedure for filtering examples for training has been modified so that when a collision occurs, maps with the robot's position at all points of its trajectory are added to the training sample. This allows you to significantly increase the success of achieving the goal.*

Key words: *the robot, path planning, the neural network, the Matlab complex.*

Introduction

It is known that a robot is an automatically operating machine that can perform some tasks that are performed by a human. The mobile robot is specifically designed for use in environments such as automated assembly halls, factories, or warehouses [1]. A distinctive feature of robots is the presence of a planning system, which today can also use neural networks with deep learning. Deep learning networks are widely used in text [2] and object recognition [3], navigation [4], scene understanding [5], and using the studied patterns in other domains [6]. Promising areas for the development of neural network systems are research related to deep learning and self-learning technologies with reinforcement and use of knowledge bases and programs of inferential learning, a posteriori learning. In automatic control systems, neural networks are used as adaptive controllers, identifiers, and trajectory planners. The stability and training of neural networks are exceptional and the most important thing in these systems. The article [7] presents an algorithm for training dynamic recurrent neural networks by Elman based on the optimization of particles in a swarm. He has developed a new control method in which a dynamic identifier allows you to identify the rotation frequency of an ultrasonic motor. The mobile system has the implemented ability to move forward or backward the desired distance, but the movement is inherently uncertain. Therefore, a neural network algorithm is used to plan the trajectory in uncertain 2D-environments. As a result of comparative analysis, the high efficiency of neural network in solving motion trajectory planning problems was shown [8]. In the article [9], a neural network is proposed for identifying the inverse dynamics of a discrete object. This allows it to provide predictive properties to the control system. In this study, we use convolution of 2D-layers of neural networks to tackle various complicated problems while the machine itself tries to explore the problem when it has enough data. The main challenge in this work was that the neural network (NN) controller had no memory of past actions and the state of the world in the past, so no collisions were detected. The architecture develops the solutions to this challenge.

Problem statement

We consider a mobile robot in a 2D-environment. The mathematical model of a mobile robot is described as follows [10, 11].

$$\dot{y}_i(t) = R(y_i) \cdot x_i\,;$$
$$M_i \cdot \dot{x}_i(t) = B_i \cdot u_i + F_{di}\,,$$

when $y_i = [y_{1i}\ y_{2i}\ y_{3i}]^T$ is the vector of the position (y_{1i}, y_{2i}) и and orientation (y_{3i}) of the mobile robot in a fixed coordinate system, $O_g Y_{g1} Y_{g2}$ (figure 1), $x_i = [x_{1i}\ x_{2i}\ x_{3i}]^T$ is the vector of linear (x_{1i}, x_{2i}) and angular (x_{3i}) velocities of the mobile robot in a moving coordinate system, OY_1Y_2, $R(y_i)$ is the kinematics matrix, M_i is the inertia matrix, F_{di} is the vector of dynamic forces, u_i is the vector of control actions, Bi is the input matrix.

The robot's navigation system measures vectors y_i and x_i

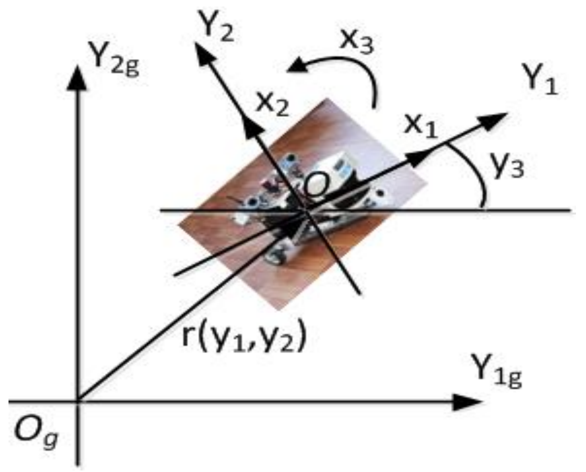

Figure 1 – The coordinate system of the mobile robot

The modelling environment we used in this experiment is a discrete map with 50×50 square cells, as we can see in figure 2.

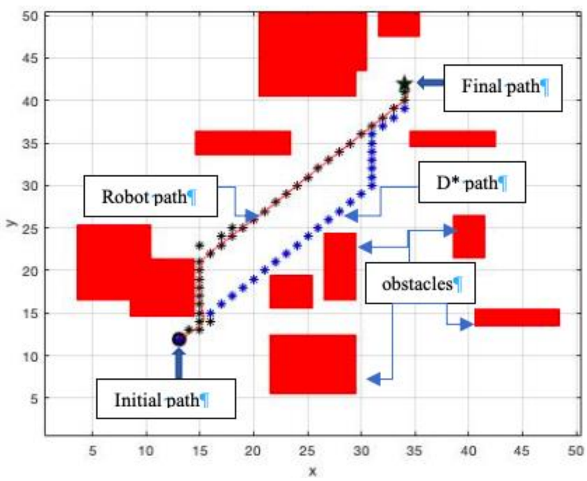

Figure 2 – The environment of learning robot

The environment contains up to 12 randomly placed obstacles. The initial position of the robot is indicated by a blue circle and the final position is indicated by a blue star. The path determined by the D* algorithm (the shortest path algorithm) is represented by a blue dotted line, and the obstacles are indicated by red squares.In this paper, the D* algorithm is used as a teacher that enables a mobile robot to learn independently of its own experience. The mathematical model of a mobile robot represented by equations (1) and (2) is considered. The results of modelling the robot (1) and (2) are used together with the trained network to update the neural network parameters for trajectory planning. The input of the neural block is a map of the environment and the trajectory developed by D* algorithms.

The situation is divided into nine classes by the D* algorithm. When planning with D*, the path passes through the cell numbered N = 1, 2, ..., 8 (figure 3), so the state class equals N. If there is no path from the current position to the robot's destination, then the state class is 9.

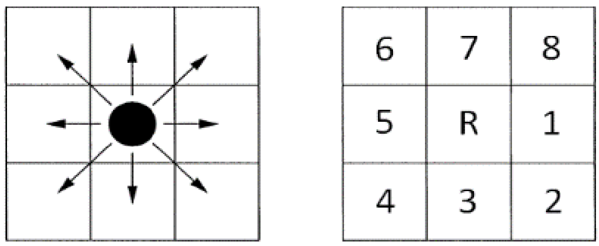

Figure 3 – Classification of cells adjacent to the robot

The task of a neural network is to lay a path and avoid collisions. The solution to this problem is divided into two stages. The first stage is basic training, which is performed by a D* supervisor. The second stage is the final training, which is performed by the reinforcement learning method.Development of the training procedure. The study of the algorithm for the development of the NN training. During the research, five iterations of training were carried out, the results of which are shown in Table and figure 4.

Table 1 – The iterative training of a neural network

Iteration	Sample size		Accuracy rating, %	Frequency of successful achievement, %
1	1000		67	
2	2000	Filtered	72	42
		unfiltered	79	51
3	3000	Filtered	73	51
		unfiltered	80	53
4	4000	Filtered	72	45
		unfiltered	81	56
5	5000	Filtered	68 %	59
		unfiltered	84 %	53

A 2D convolutional network is defined for an image of size h ×w ×3, where 3 is the number of channels for an RGB image. A convolutional neural network contains convolutional layers and auxiliary layers that terminate in fully connected layers. In the first iteration, a sample of 1000 images were created. This sample is used to train a network consisting of an input layer, five hidden convolutional layers, three hidden fully connected layers, and an output layer. The first convolutional layer contains 32 filters with a size of [3 × 3] and a step of 2. The second and third layers are also convolutional layers with 64 filters of size [3 × 3] and step 2. The fourth and fifth convolutional layers with 128 filters of size [3 × 3]. Three fully connected layers with 32, 16, and 8 neurons, respectively, were also used. All hidden layers use the ReLU activation function, which provides the fastest learning. The training options [12] include the sgdm solver used, the maximum number of epochs maxEpochs and the minimum size minibatchSize. The optimal settings were selected from the existing literature [13]. The specified network is trained with the first sample. Then the trained network is used to further increase the training sample. The motion of the robot was simulated using a trained neural network. If the robot allows a collision at any point, then such a map is added to the training sample. Otherwise, the situation is not included in the training sample. From the figure 4 shown the filtering procedure extends the training sample to 5000 images. At the same time, the training achieved an accuracy of 68 % for the sample with filtering and 84 % for the sample without filtering. However, the simulation showed that the success rate in reaching the target was 59 % for the filtered sample and 53 % for the normal sample. Filtering efficiency can be greatly improved by including the entire trajectory in the selection, rather than just the point at which the collision occurred. This can not only increase the efficiency in reaching the set target, but also improve the evaluation of accuracy during training. The results of neural network training are shown in Figures below.

Conclusion

We present the preliminary results of the work of a convolutional 2D neural network (N 1), trained without any pre-training So you can train a mobile robot to avoid obstacles. By using specially filtered images, you can correct any errors. The training was performed on a low-power laptop (MacBook Air) with a GPU.

Since the neural network only selects the direction and speed of the movement, it works faster than the D* algorithm, which builds the entire path. Moreover, the time for solving a problem is fixed by a neural network. IN the future, it is planned to develop the work by changing the filtering procedure (adding the entire trajectory to the training sample), increasing the size of the robot and the target point to improve the quality of the convolutional network, and planning a section of the trajectory by using an approximating neural network.

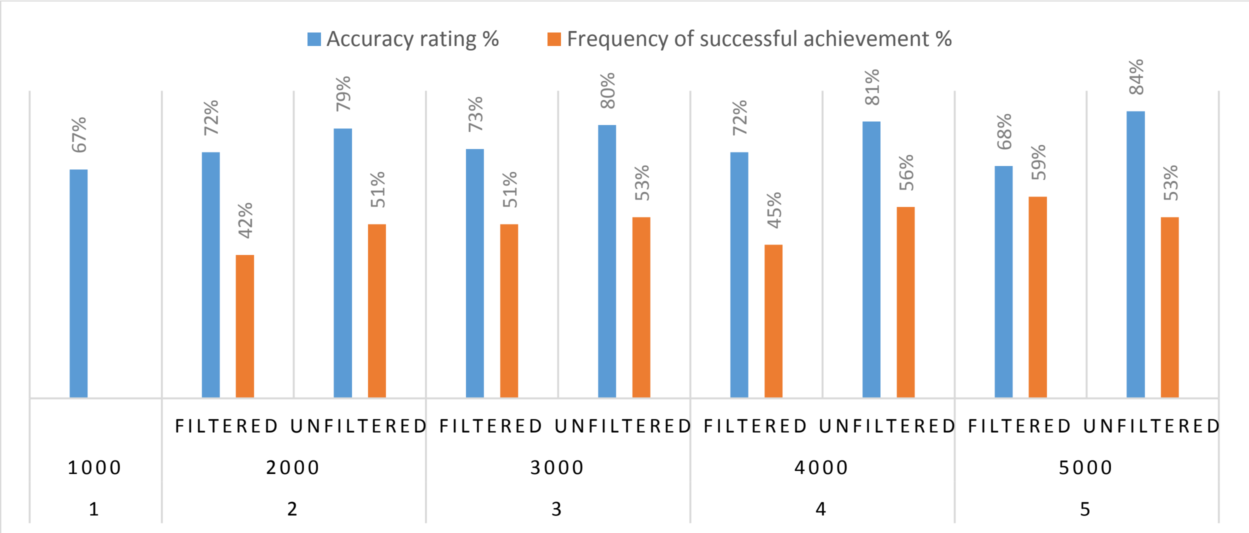

Figure 4 – A numerical study of the proposed iterative algorithm

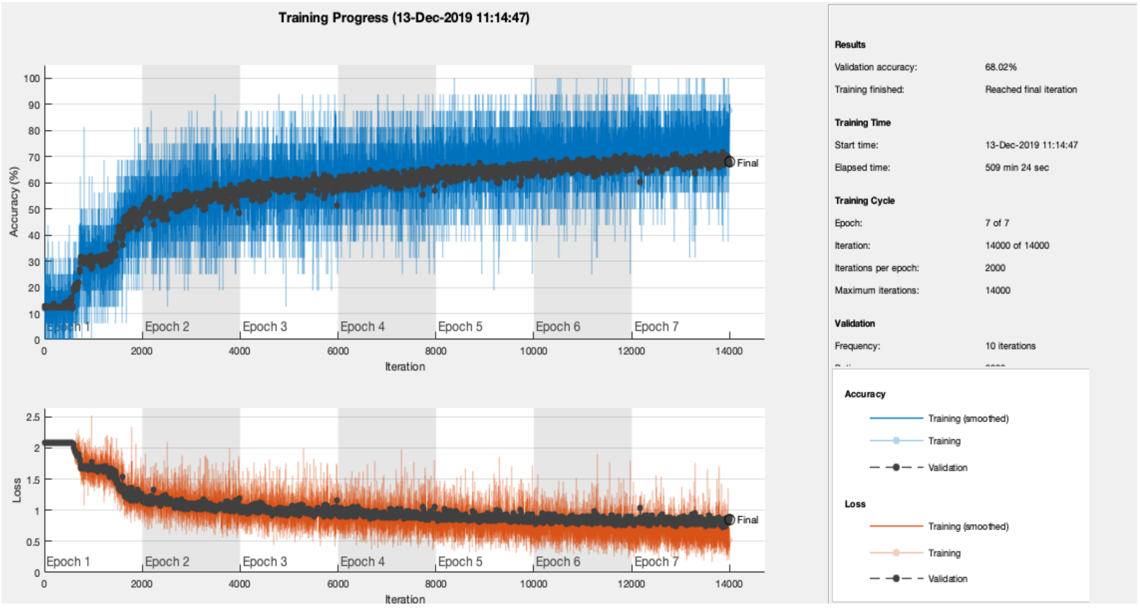

Figure 5 – Training NN with the filtering

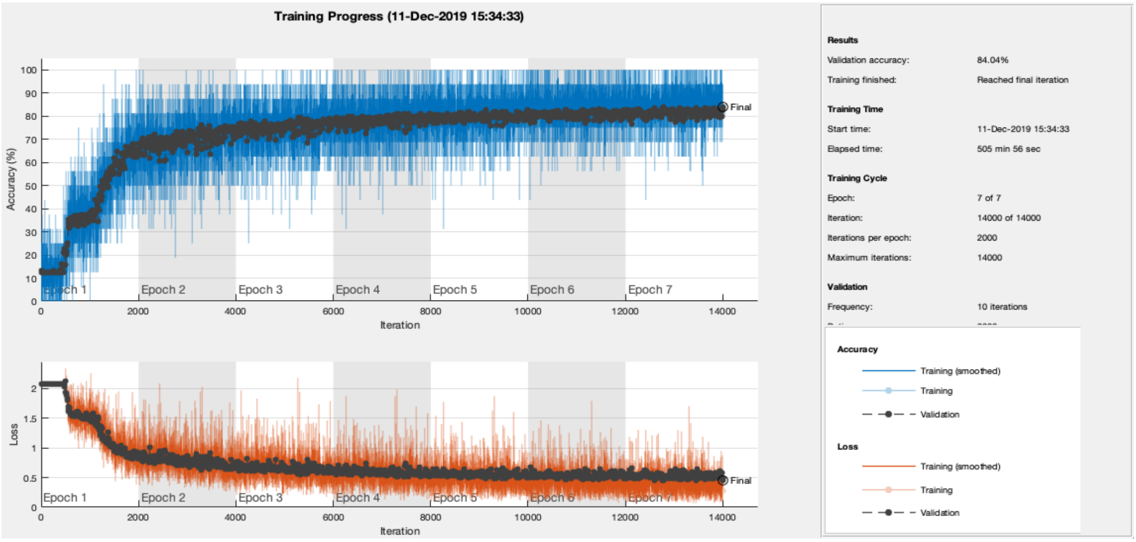

Figure 6 – Training NN without the filtering

References

1. Sandip, K. S. Data – Enhanced Convolution Network for Wall Following Navigation, 2019, IJEAT.

2. Kim, Y. Convolutional Neural Networks for Sentence Classification. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP 2014), (2014). P. 1746–1751.

3. Angelova, A., Krizhevsky A., Vanhoucke V. Pedestrian detection with a Large-Field-Of- View deep network. IEEE International Conference on Robotics and Automation. 2015.

4. Engel, J., Schöps T., Cremers D. LSD-SLAM: Large-Scale Direct Monocular SLAM. Lecture Notes in Computer Science, 2014. Vol. 8690.

5. Zhu, J., Park T., Phillip Isola, and Alexei A. Efros. «Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks», in IEEE International Conference on Computer Vision (ICCV), 2017.

6. Gholami, B., Rudovic O., Pavlovic V. Probabilistic Unsupervised Domain Adaptation for Knowledge Transfer Across Visual Categories. The IEEE International Conference on Computer Vision (ICCV), 2017, Pp. 3581-3590.

7. Hong-Wei G., Wen-Li D., Feng Q., Lu W. A Dissimilation Particle Swarm Optimization- Based Elman Network and Applications for Identifying and Controlling Ultrasonic Motors. Advances in Soft Computing. 2007. Vol. 41. – P. 1345–1360. Springer.

8. Guzik, V., Pereverzev V., Pyavchenko A., Saprykin R., Pshikhopov, V. NeuralNetworking Path Planning Based on Neural-Like Structures. Path Planning for Vehicles Operating in Uncertain 2D Environments, 2017. Elsevier, Butterworth Heinemann. – P. 25–96.

9. Notkin, B. Neuralnetworkidentificationofthe-inversedynamicsofanobjectfordesignof predictive control system. In Proc. of the 4th International conference SICPRO'05. 2005. Moscow.

10. Pshikhopov, V., Medvedev, M. Position-Path Control of a Vehicle // Path Planning for Vehicles Operating in Uncertain 2D Environments. Elsevier, Butterworth-Heinemann, 2017. P. 1–23.

11. Pshikhopov, V., Medvedev, M. Multi-Loop Adaptive Control of Mobile Objects in Solving Trajectory Tracking Tasks. Automation and Remote Control V. 81, no 11, P. 2078–2093 (2020).

12. https://ch.mathworks.com/help/deeplearning/ref/trainingoptions.html.

13. V. Nair, G.E. Hinton, «Rectified Linear Units Improve Restricted Boltzmann Machines». – Proc. of the ICML, 2010.

УДК 629.114. 2

МЕДВЕДЕВ Михаил Ю., д-р техн. наук, доц.,
профессор кафедра электротехники и мехатроники
E-mail: medvmihal@sfedu.ru

ФАРХУД А. К.,
аспирант
E-mail: eim@sfedu.ru

Южный федеральный университет, г. Таганрог, Россия

Поступила в редакцию 14.05.2022.

Проведено численное исследование предложенного итерационного алгоритма с использованием комплекса Matlab. В ходе исследования решалась задача обучения нейронной сети планировать маршрут таким же образом, что и алгоритм «учителя», в качестве которого выбран D. Начальный вариант фильтрации выбран таким образом, при возникновении коллизии в фильтруемую выборку подается точка траектории, которая предшествовала коллизии, т. е. в которой нейронная сеть неправильно классифицировала ситуацию. Такой способ фильтрации оказался не эффективным, т. к. неверное решение, которое привело к коллизии, могло быть принято не непосредственно перед коллизией, а ранее. В этой связи процедура фильтрации примеров для обучения была модифицирована таким образом, чтобы при возникновении коллизии в обучающую выборку добавлялись карты с положением робота во всех точках его траектории. Это позволяет существенно повысить успешность достижения цели.*

Ключевые слова: *робот, планирование траекторий, нейронная сеть, комплекса Matlab.*

Литература

1. Сандип, К. С. Сверточная нейронная сеть с улучшенными данными для навигации / К. С. Сандип, 2019, IJEAT.

2. Ким, Ю. Сверточные нейронные сети для классификации предложений / Ю. Ким // Материалы конференции 2014 года по эмпирическим методам обработки естественного языка (EMNLP 2014), (2014). С. 1746–1751.

3. Ангелова, А. Обнаружение пешеходов с помощью глубокой сети с большим полем обзора / А. Ангелова, А. Крижевский, В. Ванхоукке // Международная конференция IEEE по робототехнике и автоматизации. 2015.

4. Энгель, Дж. ЛСД-СЛЭМ: крупномасштабный прямой монокулярный СЛЭМ : конспекты лекций по информатике / Дж. Энгель, Т. Шепс, Д. Кремерс, 2014. – Т. 8690.

5. Чжу, Дж. Непарный перевод изображения в изображение с использованием циклически согласованных состязательных сетей / Дж. Чжу [и др.] // Международная конференция IEEE по компьютерному зрению (ICCV), 2017.

6. Голами, Б. Вероятностная адаптация предметной области без присмотра для передачи знаний по визуальным категориям / Б. Голами, О. Рудович, В. Павлович В. // Международная

конференция IEEE по компьютерной безопасности (ICCV), 2017. – С. 3581–3590.

7. Хун-Вэй, Г. Сеть Elman, основанная на оптимизации роя диссимиляционных частиц, и приложения для идентификации ультразвуковых двигателей и управления ими / Г. Хун-Вэй [и др.]. Достижения в области мягких вычислений. – 2007. –Т. 41. – С. 1345–1360.

8. Гузик, В. Планирование пути работы нейросети на основе нейроподобных структур. Планирование траектории движения транспортных средств, работающих в неопределенных 2D-условиях / В. Гузик [и др.]. Эльзевир, Баттерворт Хайнеманн, 2017. – С. 25–96.

9. Ноткин, Б. Нейросетевая идентификация нелинейного объекта для проектирования прогностической системы управления / Б. Ноткин // Материалы 4-й международной конференции SICPRO'05, Москва : 2005.

10. Пшихопов, В. Позиционно-траекторное управление транспортным средством / В. Пшихопов, М. Медведев // Планирование траектории для транспортных средств, работающих в неопределенных 2D-средах. Эльзевир, Баттерворт-Хайнеманн, 2017. – С. 1–23.

11. Пшихопов, В. Многоконтурное адаптивное управление подвижными объектами при решении задач траекторного слежения / В. Пшихопов, М. Медведев //. Автоматизация и дистанционное управление. – Т. 81. – № 11. – С. 2078–2093 (2020).

12. [Элетронный ресурс]. – Режим доступа: https://ch.mathworks.com/help/deeplearning/ref/trainingoptions.html. – Дата доступа: 20.03.2022.

13. Наир, В. Выпрямленные линейные блоки улучшают ограниченные машины Больцмана / В. Наир, Г. Э. Хинтон. Изд-во ICML, 2010.