Synthesis of neural networks for solving optimal control problems

A Voevoda and D Romannikov

1Novosibirsk State Technical University (NSTU), 630087, pr. Karla Marksa 20, Novosibirsk, Russia
E-mail: *voevoda@ucit.ru

Abstract. The task of the automated control system synthesis is an actual and complicated task especially for multichannel or non-linear objects, or optimal control systems. Neural networks can be used for these tasks in many approaches. One of the most popular its usage is for classical regulator coefficient adjusting. Another use is of a preliminary learned neural network instead of a classical regulator. The use of neural networks and reinforcement learning for the task of optimal control is considered in the paper-based example of two classical objects. Based on these examples, the method of synthesis neural network of the optimal control systems is proposed. The main steps of this method are to compose a closed-loop system with a considered object and relay control; form a reward function that is focused time of transition process minimization; perform a reinforcement learning with one of the existing methods. The proposed method is able to synthesize neural networks for optimal control of the system. Recommendations are given for choosing the structure of a neural network, error functions, and other practical recommendations to speed up learning. Examples of application of the proposed methodology on two classical problems of optimal control are given. The results obtained coincide with theoretical calculations for classical regulators.

1. Introduction
Nowadays, the use of neural networks for solving artificial intelligence tasks is very popular. Particularly, they are widely used for image recognition [1], voice and speech recognition, [2] and etc. Neural networks are also used in reinforcement learning tasks: convolutional neural networks were used for the implementation of systems that is able to play Atari games [3], Go [4], and chess. It needs to mention, that there are some modifications with using of long-short term memory (LSTM) or gain recurrent neuron (GRU) layers, and etc. In [5], variants of the synthesis of neural networks for logical-arithmetic problems are proposed, and in [6] for other static problems.

Neural networks also can be used for tasks of closed-loop system stabilization. There are many approaches of such use. The most popular are considered below. The first one is a use of a feed-forward neural network (or some analogs with additional layers) instead of classical regulators. The main idea of this approach is to learn a neural network with earlier received data (the date can be calculated with reverse time approach or it can be collected from the preliminarily calculated classical
closed-loop system) [7]. The second approach is to use a regulator for object stabilization and the use of a neural network for adjusting the regulator’s coefficients [8, 9].

Neural networks and reinforcement learning were applied for the task of stabilization of closed-loop systems. For example, in the [10] were proposed using of neural network that can stabilize the closed-loop dual-mass spring system with two inputs and two outputs. The first important aspect of the [10] paper is that a process of stabilization of the system is performed from the initially unstable state. The second aspect (and it’s a drawback of the [10]) is that approach can be applied for stabilization only with the preliminary deterministic set point. The approach of stabilization with preliminary deterministic setpoints range where proposed in the [11]. Particularly, it was shown on the same object and setpoints range \(0, 4\).

In the paper, authors apply reinforcement learning and neural networks for the tasks of optimal systems. In the classical theory of closed-loop system control, optimal systems were successfully investigated [12, 13] and optimal trajectories of the phase characteristics are well known for some number of objects. We consider using neural networks and reinforcement learning on examples of two classical objects that are described in the next section.

2. Methods
We consider the problem of the synthesis of neural networks that can stabilize objects optimally. This problem is considered on the two objects:

\[
W_{obj1} = \frac{1}{s} \tag{1}
\]

\[
W_{obj2} = \frac{1}{s^2 + 1} \tag{2}
\]

from the non-zero initial conditions of the object. Objects are converted into discrete representation for the purpose of modeling. For both objects we changed continuous integrator to its discrete representation and made sure that the nature of the transition process persisted and the difference between them is lesser than 5%. For the object Eq. (1) we used Matlab’s command \(c2d([1], [1 0]), 0.05)\) and for the object Eq. (2) – command \(c2d([1], [1 0]), 0.05, \text{ ‘foh’} \) .

Objects models are displayed in figures 1 and 2 respectively.

**Figure 1.** Model of the object \(W_{obj1} \).

**Figure 2.** Model of the object \(W_{obj2} \).
For both object relay control with amplitude {-1, 1} is used.

In terms of reinforcement learning, input information is a state of the object, and a model or a policy needs to stabilize. For both objects state that determines a next state is values of all 1/z elements of the objects (see figures 1 and 2). For some objects these values are not available, but they can be well estimated with neural network observers [10], and in this research we use values from the objects. The schema of the closed-loop system with using of the neural network is displayed on the figure 3.

Figure 3. Schema of the system with a neural network.

On figure 3 instead of a regulator we use a neural network. Input signals of the neural network are values of delay units inside of the object. An output signal of the neural network is control value for the object.

One of the problems why it’s difficult to apply reinforcement learning to the task of stabilization of closed-loop systems is that a range to setpoints must be supported [10]. For the task of optimal control, it’s not an issue due to setpoint is stable and it’s zero values on the object’s delay elements however initial conditions on the delay’s elements can be different.

Based on the above and [10] we formulate the method of the neural network learning with using reinforcement learning for objects stabilization by optimal or close to optimal control:

1. Compose a closed-loop system with an object and a neural network that has a state of an object on the input and probability to perform one of two actions {-A, A} on the output, where A is an amplitude of the control signal;
2. Perform learning of the neural network using the Deterministic Policy Gradient method [15] (instead of this method any other reinforcement learning method can be applied, but authors got the best results with use of this method) on the non-zero initial conditions of the system. Initial conditions should be randomly changed in a predefined range. It’s recommended to use buffers replay [3] for the more stable learning process. For providing a close to optimal transition process needs to modify reward function:

$$\text{Reward} = \begin{cases} 1 - y & \text{if } t \geq T \text{ and } e < \text{Error Limit} \\ -1, & \text{otherwise} \end{cases}$$

3. where $y$ is object output, $t$ is a number of the current step, $T$ is a number of steps for stabilization, $e$ is a current error and $\text{Error Limit}$ is a parameter of static error;
4. Every N (can be chosen depend on the task) epoch, perform testing of the model on predefined set of initial conditions of the object of the system:

$$\text{Loss} = \sum_{j=0}^{k} e_j^2$$
5. where Loss is a total loss of the testing that calculated for $k$ initial conditions and the error on each testing episode is a sum of squared errors on each $j$ step of the stabilization process. After calculating the Loss, the model with the lowest value must be selected.

3. Results
The method above we apply for the task of stabilization of objects from the problem section. For the Eq. (1) the best reward was 417 that was archived after 308000 episodes. It was done with a neural network with two inputs (state of the object), two hidden layers with 20 and 10 neurons respectively, and one output neuron. All neurons (except the output) has relay activation functions, the output neuron has hyperbolic tangent to provide bounds on probability values. The process of learning was done with Adam optimizer [14] with 0.0001 learning rate value. The length of one episode was 100 steps where each step last 0.05 seconds according to discretization step. For the all experiments testing was performed each 1000 iterations on the following initial conditions of the object’s delays: {2, 2}, {1, 1}, {1, 0}, {0, 1}. The transient graph can be seen in figure 4 and the corresponding phase characteristics are displayed in figure 5.

![Figure 4](image4.png)

**Figure 4.** Output of the object Eq. (1) with initial condition {1, 1}.

![Figure 5](image5.png)

**Figure 5.** Phase trajectory of the object Eq. (1).
The Eq. (2) is stabilized with the neural network with the same structure and the same optimizer. The best reward result (316758) for the second object was achieved after 77000 episodes. Transient graph can be seen in figure 6 and corresponding phase characteristics are displayed on figure 7.

Figure 6. Output of the object Eq. (2) with initial condition {1, 1}.

Figure 7. Phase trajectory of the object Eq. (2).

All obtained transition processes in figures 5, 7 are consistent (with a low error of several percent) with expected and mathematically sound results, so the trajectory of the neural network’s control signal is close to optimal.

4. Conclusion
The article presents a method for training neural networks, the structure of which was synthesized based on recommendations [5, 6], with the goal of obtaining a regulator for stabilization of an object close to optimal control. The proposed method was tested at two objects for which close to optimal trajectories of the transition process and phase characteristics were obtained. The proposed structure of the neural network is implicitly based on the structure of the corresponding controller: in the classical control theory, there is an assumption that the controller contains at least the same number of delays as in the object. And based on
this assumption, it is enough for a neural network to contain several outputs equal to the number of delays in the object, and another additional from signal control. The number of inputs should be equal to the number of delays in the object and one additional for the error signal.

References
[1] Krizhevsky A, Sutskever I, Hinton G E 2012 *ImageNet Classification with Deep Convolutional Neural Networks*, Proceedings of Neural Information Processing Systems (NIPS), pp 1097-1105
[2] Graves A, Mohamed A, Hinton G 2013 *Speech recognition with deep recurrent neural networks*, Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp 6645-49
[3] Mnih V, Kavukcuoglu K, Silver D, Graves A. 2013 *Playing Atari with Deep Reinforcement Learning*, Proceedings of Neural Information Processing Systems (NIPS), pp 1-9
[4] Silver D, Huang A, Maddison C, Guez A, Sifre L, Driessche G 2017 *Mastering the game of Go with deep neural networks and tree search*, Nature, pp 484-503
[5] Voevoda A A, Romannikov D O 2017 *The synthesis of a neural network for solving logical and arithmetic problems*, Proceedings of SPIIRAS, pp 205-223. https://doi.org/10.15622/sp.54.9
[6] Voevoda A A, Romannikov D O 2018 *Synthesis of a neural network for n-dimension surfaces*, Actual problems of electronic instrument engineering (APEIE–2018) DOI: 10 1109/APEIE 2018 8545704
[7] Sutton R, Barto A 2018 *Reinforcement Learning: An Introduction*, MIT Press, Cambridge, p 1328
[8] Alvarado R, Valdivinos L, Salgado-Jiménez T, Gómez-Espinosa A 2016 *Neural Network-Based Self-Tuning PID Control for Underwater Vehicles*, Sensors (Basel), pp 898-903
[9] Kumar R, Srivastava S 2016 Gupta,Artificial Neural Network based PID controller for online control of dynamical systems, 2016 IEEE 1st International Conference on Power Electronics, Intelligent Control and Energy Systems (ICPEICES), pp 1-6
[10] Voevoda A A, Romannikov D O 2019 *Synthesis of regulators for multichannel systems using neural networks*, Scientific Bulletin of Novosibirsk State Technical University, pp 7-16. http://dx.doi.org/10.17212/1814-1196-2019-4-7-16
[11] Preusse S 2013 *Technologies for Engineering Manufacturing Systems Control in Closed Loop*, Hallenser Schriften Zur Automatisierungstechnik, (Book 10) p 145
[12] Goodfellow I, Bengio Y, Courville A 2016 *Deep Learning*, MIT Press, p800
[13] Haykin S 2009 *Neural networks and learning machines*, Pearson Education, p 936
[14] Kingma D, J Ba Adam 2014 *A Method for Stochastic Optimization*, Proceedings International Conference on Learning Representations, pp 1-12
[15] Silver D, Lever G 2014 *Deterministic policy gradient algorithms*, In Proceedings of the 31st International Conference on International Conference on Machine Learning, pp 387-395