Reinforcement Learning Based on Backpropagation for Mobile Robot Navigation

Rudolf Jakša, Pavol Majerník, Peter Sinčák
Computational Intelligence Group
Department of Cybernetics and Artificial Intelligence
Technical University, Košice
E-mail: jaksa@neuron-ai.tuke.sk

Abstract. The paper deals with application of reinforcement learning based on backpropagation algorithm to control a mobile robot. The control was based on sensor information concerning position of vehicle in environment and radar information about obstacles and provide a steering signal and signals for acceleration/deceleration of vehicle. The control task is to reach desired position from any point of the environment. The neurocontroller consists from two neural networks with backpropagation learning algorithm accomplishing reinforcement learning approach.

1 Introduction

The neurocontrol is based on application of neural networks in control engineering. Key properties of neural networks used for control are:

- nonlinear dynamics,
- natural complexity (multiple inputs/outputs and complex internal structure),
- adaptability and learning ability.

Neural Networks have high application potential in control engineering [7], [5]. Theoretical background of this fact is presented in “brain as neurocontroller” idea [11]. The most common approach concerning artiﬁcial neural networks (ANN) is error backpropagation algorithm (see [4]), especially when reflecting applications. This is the motivation for the use of the backpropagation with all concerning know-how within the framework of reinforcement learning. From some point of view the approach which we used is an extension of the backpropagation algorithm with ability to solve more general (reinforcement learning) tasks. From this point of view, they are three main modes of ANN based control [2]:

1. copying an existing controller in supervised learning mode,
2. differentiating a model – backpropagating through a forward model of the plant to determine controller errors,
3. reinforcement learning.

Sophisticated analysis of reinforcement learning applied to control tasks can be found in recent works of Dr. Werbos [12] and good introduction to robot control and navigation in [8].
2 Integration of reinforcement learning and backpropagation

One of the most interesting areas in neurocontrol is a reinforcement learning (RL). There are many of different reinforcement learning algorithms [3], [10]. Well known are especially actor-critic architectures and Q-learning algorithm. Reinforcement learning problems are equivalent to optimal control problems. The control task is defined using evaluation function and the goal is to control plant the way minimizing this evaluation function.

One of characteristics of the area of reinforcement learning based control can be a lack of applications. Most preferred reinforcement learning approach here is probably Q-learning [1] followed by CMAC based adaptive critic architectures with simple neural networks [6]. However, it is possible to combine adaptive critic architectures with all supervised learning approaches, backpropagation including. This is the case used in our approach or for example in [9].

There was used an actor-critic reinforcement learning architecture in our experiments, where both, the action and the critic networks, were layered feedforward neural networks with a backpropagation learning algorithm. This is not usual type of reinforcement learning system configuration and it is next step after backpropagation supervised learning based control of mobile robot. The idea was: to do simplest backpropagation based reinforcement learning.

2.1 Control task

A mobile robot navigation is complex control task – suitable for neurocontrol. Experiments were done on mobile robot simulator which is similar to Khepera robot. The main goal of the robot was to start robot movement from any start position in the environment to reach a desired position in unknown environment.

2.2 Controller

Typical example of used neurocontroller based on RL is on Figure 1. and most simple architecture on Figure 2., where $x, y$ are position coordinates of vehicle, $cx, cy$ are coordinates of target position, $r0, r1, r2$ are radar signals used for obstacle detection, $v$ is acceleration of vehicle, $\alpha$ is course of vehicle and $ev$ is evaluation of last vehicle’s action.

Evaluation of controller’s actions can be computed in many ways, for vehicle used in experiments was in general used function: $ev = (cx - x)^2 + (cy - y)^2 + r0 + r1 + r2$. Evaluation is best in minimal values and worst in maximal values. The goal of whole system is to minimize the evaluation. The critic’s task is to compute approximate evaluation. It is pure backpropagation task with error function: $J = (ev - y)^2$, where $y$ is critic’s output. The learning of actor is made by modified backpropagation algorithm with error function $J = y$. The $y$ is critic’s output and the derivation of $J$ is propagate through critic, without any impact to the values of the weights. Backpropagation is, in general, recursive gradient descent algorithm:

$$\Delta w_{ij} = -\gamma \frac{\partial J}{\partial w_{ij}} = -\gamma \frac{\partial J}{\partial x_i} \frac{\partial x_i}{\partial in_i} \frac{\partial in_i}{\partial w_{ij}}.$$

The $x_i$ is activation of $i$-th neuron and for last layer is $x_i = y$, $i = 1$. The $in_i$ is input of $i$-th neuron $in_i = \sum_{j=1}^{N_j} w_{ij} x_j + \theta_i$, where $N_j$ is number of neurons to which is
From equation (1) is simple to derive backpropagation rule for error function $J = \sum_{i=1}^{N_i} (ev - y)^2$ (critic’s adaptation). For last layer it is:

$$\Delta w_{ij} = \gamma (ev_i - x_i) f'(in_i) x_j$$  \hspace{1cm} (2)

and for other layers:

$$\Delta w_{ij} = f'(in_i) x_j \sum_{h=1}^{N_h} \Delta w_{hi} w_{hi}$$  \hspace{1cm} (3)

Subscript $h$ is here used for neurons connected to $i$-th neuron. For actor’s adaptation is $J = x_i$, $\frac{\partial J}{\partial x_i} = 1$ and adaptation rule (2) is changed to:

$$\Delta w_{ij} = -\gamma f'(in_i) x_j$$  \hspace{1cm} (4)

The $\Delta w_{ij}$ values are applied to actor’s weights, but not to critic’s weights. In the critic case are $\Delta w_{ij}$ temporarily stored and used for error backpropagation only. Not used to change weights in critic network. Whole algorithm is:

1. initialization of neural networks, computing of control signals and applying them to the plant,
2. evaluation of actual behavior of vehicle,
3. adaptation of critic using evaluation from step 2,
4. adaptation of actor
   (a) error backpropagation through critic,
   (b) true actor adaptation,
5. computing of control signals, applying them to vehicle and continuing with step 2.

3 Experiments

In recent experiments we were concentrated to critic. Very simple structure of actor was used only. The experiments shows the necessity of high speed critics learning (relatively to the sampling rate). We found that the most simple way to do this, is using a very high values of learning rate (Figure 5, 6).

We often used learning rate $\gamma > 3$. This approach often lead to saturation of neurons, which caused that learning stops. We found that dynamic changing of learning rate
can avoid such problems in some cases. We presume, that use of more complex actor
network will reduce requirements to the learning algorithm. More complex network has
ability to store information about state-action mapping for bigger range of states. With
simple network it is necessary to re-learn network always, when small area of actually
covered states is leaved off.

4 Conclusion

Our recent research is presenting experience with a very simple reinforcement learning
algorithm for mobile robot navigation. We found that it’s possible to use classical
layered feedforward neural networks with backpropagation learning algorithm as critic
networks in reinforcement learning. In future experiments we would like to concentrate
on action network, and test action networks with more neurons, also test recurrent
neural networks. Our goal is to study characteristics of reinforcement learning algorithm
for complex system control with aim of possible improvements of used controllers.

References

ference on Artificial Neural Networks Proceedings. VUJE Trnava a.s., 1998. ISBN: 80-
967935-2-7.
[8] Ben Kröse and Joris van Dam. Neural vehicles. In Omid Omidvar and Patrick van der
the IEEE/ICNN International Joint Conference on Neural Networks (ICNN97), 1997.
http://www.acil.ttu.edu/publications.html.
1998.
Figure 5: Deviation between desired evaluation $ev_f$ and approximate critic’s output $ev_{nn}$ with learning rate $\gamma < 1$ (one of best results)

Figure 6: Deviation between desired evaluation $ev_f$ and approximate critic’s output $ev_{nn}$ with high learning rate $\gamma > 1$ (typical behavior before saturation)