Backpropagation in Supervised and Reinforcement Learning for Mobile Robot Control*

Rudolf Jakša, Peter Sinčák, Pavol Majerník
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 backpropagation algorithm in both, supervised and reinforcement learning approaches in task of mobile robot navigation. The experimental environment used in both cases is the same. The control is 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 in reinforcement learning case and to reach desired position using supervisor’s instructions in supervised learning case. The neurocontroller consists from two neural networks with backpropagation learning algorithm accomplishing reinforcement learning approach or from one simple neural network with backpropagation algorithm in supervised learning experiments.

1 Introduction

Most applications of neural networks in 90-ties are based on supervised learning approach and especially on feedforward networks with error backpropagation learning algorithm. The same situation is in applications of neural networks in control area, or in other words in neurocontrol. However in this field is very bold impact of reinforcement learning approach.

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]). 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 in reinforcement learning experiments 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]:

*The core of this paper was presented on the CIMCA’99 (Computational Intelligence for Modelling, Control and Automation, Vienna 1999) conference.
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.

Introduction to reinforcement learning can be found in recent book by Barto and Sutton [10]. 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 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 without supervisor’s intervention or with it, depending on reinforcement versus supervised learning approach.

Figure 1: Example of vehicle’s trajectory (reinforcement learning)

Experiments were done using computer simulation of vehicle in hill-less environment. There was used very simple model of vehicle with very simple radar (3 short radar rays in front of vehicle).

3 Controller

3.1 Supervised learning based controller

In our experiments with supervised learning based control were used two basic configurations of controller. The 1st one was single network neurocontroller. It was single neural network with all outputs from the vehicle as their inputs and producing all control signals for the vehicle. Classical layered feedforward network with backpropagation learning algorithm was used.

In the 2nd configuration was used semimodular neural network consisting from three networks cooperating in parallel on control of vehicle. All the networks had the same inputs, but every one was specialized for producing one control signal (for control of course of vehicle, control of accelerator and brakes).

In both cases (single network and semimodular) was used the same basic principle of supervised learning based control. It was copied an existing controller into neural network. In our experiments it means, that it was used manual control of vehicle (driving) to produce patterns for learning the network(s). After successful
learning the network(s) was(were) able to drive the vehicle.

3.2 Integration of reinforcement learning and backpropagation

One of the most interesting areas in neurocontrol is reinforcement learning (RL). There are many 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.

3.3 Reinforcement learning based controller

Typical example of used neurocontroller based on RL is on Figure 5. and most simple architecture on Figure 6., 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 com-
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 $J = \sum_{i=1}^{N_i} (ev - y)^2$ (critic’s adaptation). For last layer it is:

$$\Delta w_{ij} = \gamma(ev_i - x_i) f'(in_i) x_j ,$$

and for other layers:

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

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:

$$\Delta w_{ij} = -\gamma f'(in_i) x_j .$$

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.

3.4 Implementation

The whole system (vehicle simulation and controller) was implemented under a Unix-like operating system.

Figure 8: GUI of vehicle simulator

The simulation environment was designed in a highly modular system and any type of modification of the particular module is possible. Single logic parts of the simulation system were realized as independent tasks in the multitasking operating system. Shared memory (System V IPC standard) was used for communication between tasks. The modules were programmed in C language to utilize a full power of CPU. In supervised learning based experiments was used SNNS\textsuperscript{1} simulator software for learning neural networks and creating neurocontrollers, while in reinforcement learning experiments was used own implementation of learning algorithms, able to cooperate on-line with vehicle simulator.

![Diagram](image)

Figure 9: Modular supervised learning based control implementation

4 Experiments

Concerning supervised learning based experiments we can conclude that a semi-modular neural network complex controller is much more convenient for control tasks than a single neural network controller. In fact we was not able

\textsuperscript{1}Stuttgart Neural Network Simulator
to obtain usable behavior of vehicle with single network configuration of neurocontroller.

In recent experiments with reinforcement learning we were concentrated to critic. The situation in reinforcement learning experiments is slightly different from supervised learning control. The task is harder (not supervisors information is available) and basic configuration is modular (in actor-critic architecture), however there is no relation between modules from our supervised learning based experiments and those reinforcement learning based.

Up to now in our experiments 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 11, 12).

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 lefted off.

The results of our reinforcement learning based controllers are not so good as these of supervised learning controllers, but the reinforcement learning architecture we used is not so mature as commonly used supervised learning approach and, as was mentioned sooner, the task is harder in reinforcement learning case,
but more powerful and universal. We think, with improvements of algorithm we will be able to obtain in reinforcement learning case as good results as in supervised learning case.

5 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


