Learning of Sensorimotor Contingencies to move a Four-Legged Robot

CARSTEN BÖHME[1], PHILIPPE BÜTTNER[1], CORNELIUS WEFELSCHEID[1] & TILMAN WEKEL[1]

[1)] Technische Universität Berlin, Germany

Keywords: AIBO, four-legged autonomous robot, feed-forward neural network, learning, artificial intelligence

Abstract

In this paper, an approach for learning of crawling movements of a four-legged autonomous robot is presented. On the basis of a self-supervised strategy, the robot learns to choose its next movement by its sensorimotor states. This is based on feedback gained from the environment by actively tracking an object. Within a few hours of training, the robot shows interesting crawling behaviour patterns which occur in different stages, comparable to the learning of movement by infants.

1. Introduction

Locomotion is one of the most important skills of many animals including humans, and is essential for survival from an evolutionary perspective. Some theories even say that animals have evolved brains mainly for movement purposes, whereas plants did not need to evolve a brain for the same reasons (Wolpert 2002).

In young infants, walking behaviour develops through several skill levels. To explore first sensorimotor capabilities, young infants perform simple body movements, without locomotion. Then an effective, low-risk way of moving forward is discovered – crawling. A period of sensorimotor exploration is then followed by first attempts to walk upright. Certainly, four-legged movement is easier and more stable, so a direct comparison between the development of human walking and robotic four-legged walking is not possible (Lungarella et al. 2003).

Similar work in robotics has been performed to research walking strategies. In Golubovic & Hu (2003) for example, a fast walk for the AIBO has been discovered using optimisation through artificial evolution. Kohl & Stone (2004) also used the Sony AIBO for learning a robot walk. But reinforcement learning was used here for optimisation the parameters of a predefined walking gait. However, many prerequisites of walking have already been implemented. This approach focuses on learning from scratch. How much fundamental programming is needed as input for the robot for self-advised learning to move forward?

Figure 1: The AIBO used for the experiments during a learning phase. The robot can perform all physical possible movements in order to try to move forward, as long as they are symmetric and do not put too much force on the joints. The binary touch sensors on the paws are also fed into the neural network, together with the position of the joints and the intended movements of these joints.

In this paper, a simple way of having a four-legged robot learn to crawl is presented. In the experimental section, the experimental setup is introduced together with the neural network used for the experiments. Finally, the results are presented and the advantages and problems with this approach are also discussed.

2. Experiments

2.1 Experimental Setup

The experiment has been performed with an AIBO (Artificial Intelligence Robot), a robotic pet designed and manufactured by Sony (see figure 1). The project's goal was to identify ways in which the robot learns which actions cause which reactions and how to value coherences. Therefore as little knowledge as possible was given to the robot. Walking should be learnt by the robot autonomously.

The goal of the robot was to approach a pink ball. To make sure the robot always sees the ball, a ball tracking algorithm was implemented. At the same time, the camera image seen by the robot was also used to measure the distance between the ball and the robot, which was the main factor for the rating function for learning to move forward.

The experiment was performed in an office with constant lighting conditions. This was important for obtaining good pictures from the camera in order to centre the ball in the picture and finally to calculate the distance to the object. The AIBO is equipped with two distance sensors, two microphones and one camera in the head and one distance sensor in the chest. Altogether the AIBO has 20 degrees of freedom. For this experiment we only needed 15 degrees of freedom. The head uses 3 degrees for tracking the ball and 12 degrees for walking, or 3 degrees for each leg. The remaining degrees of freedom are for the tail, the ears and the mouth, which had been disregarded for this experiment.

2.2 Neural Network

An artificial neural network tries to simulate the real connections of biological neurons within the brain. The structure of artificial neural networks is usually for simple problems nearly the same. There is one input layer, one or more hidden layers and one output layer. Picking the right structure is the hardest part. The ideal structure for this problem would be a network which receives the actual position and returns the next movement. The training of such a neural network would be very complicated, because specific training data is needed that deliver for every position the best succession position. With that solution the robot never finds an own walk, but it would just copy an existing walk. That was not our goal, so this structure was rejected. Instead we used a structure where the joints and the next movement are the input of the network and the potential differential distance of the movement is the output.

Now we had the opportunity to run a list of possible movements through the network. The movement with the highest result is executed. After the movement is finished the synapses are trained with the output of the rating function. We also needed to find a way to rate neutral movements, in which the position of the AIBO does not change. A walk is a combination of many movements. When walking, about half of the movements will be needed to bring the joints back to their previous position. So the AIBO is able again to move forward.

The first neural network we tried has a 28 dimensional input vector with the following parameters: 12 joint position, 4 feet sensors, and 12 potential movements. The hidden layer consists of 14 neurons and the output layer is just one neuron. That makes an overall of 406 synapses that needed to be trained. The learning algorithm used in these experiments is back-propagation with a learning rate of 0.1 and a momentum term of 0.8. The amount of training data needed was between 1000 and 10000 samples. In order to reduce the number of synapses we created a second neural network and also restricted the mobility of the AIBO to symmetrical movements. This meant it lost the possibility to learn a perfect walk, but could still crawl. The second neural network has a 16 dimensional input vector (6 joints positions, 4 feet sensors, 6 potential movements). After this change we had a total of 238 synapses, which also reduced the training data required (see figure 2). Starting with a new training session means the neural network is initialised with random numbers. Usually the AIBO always performs the same movement until it reaches its physical limits. To avoid that behaviour the AIBO in the beginning carries out a lot of random movements and trains the neural network with the result (Russel 2004). With the number of fulfilled movements, the likelihood for making a random movement decreases.

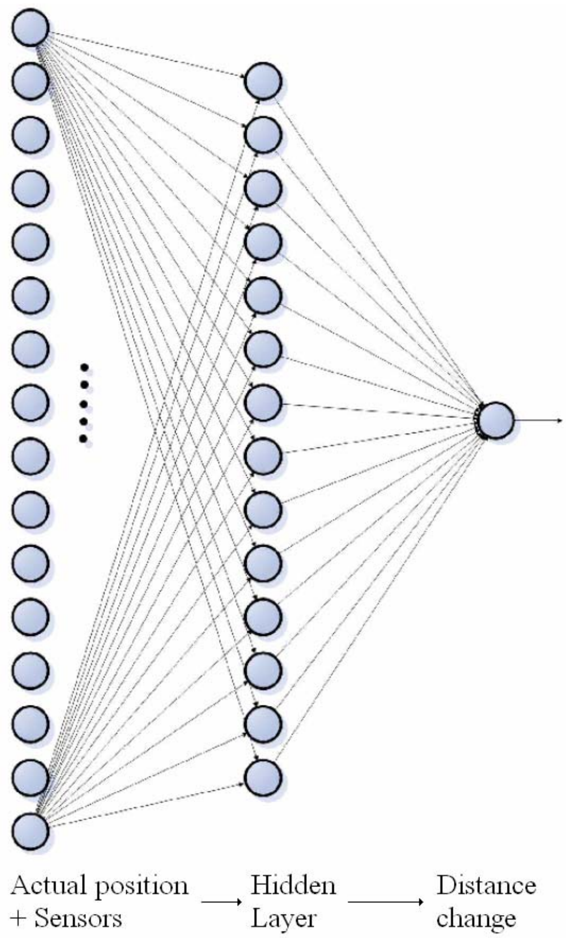

Figure 2: The feed-forward neural network structure used for learning the joint values. We used a multi-layer perceptron with input, hidden and output layer.

The neural network is trained by supplying a set value for the movement. The set value depends on the distance measurement and a weighting function. In the normal case, the weighting function *B* is made up of the distance and the previous three distance values as follows:

$$B = \Delta d + \frac{\Delta d_1 + \Delta d_2 + \Delta d_3}{3(100 \cdot \Delta d + 1)}$$

Normal case means there is no collision and the movement of the leg is within of the joint's movable area. The aim is to evaluate a movement that has no distance change as result. This neutral movement is necessary when the leg is pushed forward to make a new movement possible. Basically, we consider that neutral movements are convenient and essential if they are followed by a good movement. Hence, the previous three Δd values flow into the calculation.

To prevent the legs from colliding with the body, the distance change flows only with 10% into the weighting function, if there is a collision:

$$B = 0.1 \cdot \Delta d$$

Movements outside the joints' movable area will be trained to -0.1, so the AIBO learns to avoid such movements in the future:

$$B = -0.1$$

3. Results

In this section we want to describe the results in comparison with the non-symmetric attempt. The two curves in figure 3 show the relative error of a new network (red) and a trained network (blue). The improvements of the trained network are apparent. The parts between the peaks are usually movements that did not change the position of the robot. Peaks in the blue line indicate a distance change to the target. It is clear that it is easy for the network to learn if a movement has no consequences, but it is harder to tell the exact distance change.

The major problem of this attempt is that probably less than 1% of all possible movements are good ones, but the network needs to do them at least a few times until it has trained them. This problem gets bigger for the non-symmetric attempt, because the interplay of different movements is even more important. Finally we can say that the training data that would be needed to train a perfect walk are huge and the speed to get the training data from the AIBO by doing random movements is really slow.

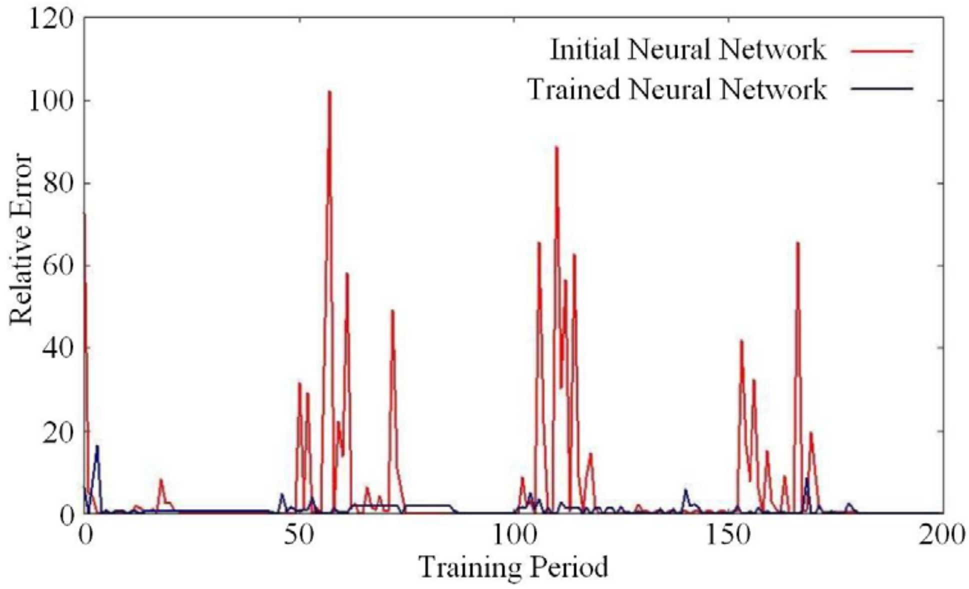

Figure 3: Curve with prediction errors for the distance change. The red line shows the results for a new neural network before training, the blue curve shows the results of the network after a few hours of training.

4. Conclusion

The prediction of the consequence of each movement is relatively precise, but for large movements the failure increases (see figure 3). In this respect, we can say that the neural network mainly learns movements that have no direct consequence. But also the movements that are not possible because of the mechanics or which cause collisions with other joints are learnt quite well. So the general attempt did not collapse. The AIBO is crawling in a forward direction, but it isn't really walking. Probably the attempt can be improved with further optimisation. Finally, we can say that it is possible to use neural networks as a prediction for robot movements.

5. Future Work

In this section, we discuss further possibilities and give a motivation to continue this study. The AIBO should have the possibility to handle the symmetric leg programming in an advanced stage. Therefore learning should be divided in different phases. By reaching an advanced phase the symmetric leg programming will be badly evaluated and avoided in further steps. This is unfortunately anticipated by the central neural network. But nature has already shown us that a central control of legs is of disadvantage. The grasshopper regulates the legs by a decentralised control. Each leg is controlled by its own coordination centre in which the information for the joints arrives. This works if local rules control the movement of the joints. By using this technique the grasshopper reaches incredible movements although it has only a few thousand neurons.

The absence of an intelligent walking gadget could be put into practice through a simulation, in which the elasticity of each joint is simulated via a weighting function. The leg moves back into the initial position by simulation, as soon there is no intentional movement. This again should lead to learning. But if the environment of the AIBO is simulated and if AIBO would be fed with an adequate amount of physical knowledge, movements could be simulated and evaluated by success before they are accomplished. By a successfully simulated movement the movement itself can be transferred into the real world. Finally, the trial and error runs could be reduced significantly.

References

Golubovic, D. & Hu, H. (2003). *Parameter optimization of an evolutionary algorithm for online gait generation of quadruped robots.* Proceedings of the IEEE International Conference on Industrial Technology - ICIT'03.

Lungarella, M. et al. (2003). *Developmental Robotics: A Survey.* Connection Science, vol. 15, no. 4, pp. 151-190.

Kohl, N. & Stone, P. (2004). *Machine learning for fast quadrupedal locomotion.* The Nineteenth National Conference on Artificial Intelligence, pp. 611-616.

Russell, J.S. (2004). *Künstliche Intelligenz (Artificial Intelligence),* 2^{nd} ed. Munich: Pearson Studium.

Wolpert, L. (2002). *Love is a many-moleculed thing.* London: The Observer.