NEAT in Increasingly Non-Linear Control Situations

Matthias J. Linhardt
Department of Psychology III
University of Würzburg
97070 Würzburg, Germany
MatthiasLinhardt@web.de

Martin V. Butz
Department of Psychology III
University of Würzburg
97070 Würzburg, Germany
mbutz@psychologie.uni-wuerzburg.de

ABSTRACT
Evolution of neural networks, as implemented in NEAT, has proven itself successful on a variety of low-level control problems such as pole balancing and vehicle control. Nonetheless, high-level control problems still seem to trouble neuroevolution approaches. This paper presents such a complex task and explores how different aspects of problem difficulty have varying strong influences on NEAT’s performance. Based on these findings, the question is discussed why certain problem domains are less beneficial for neuroevolution approaches’ performance, which may provide useful insights into how to design the next generation of neuroevolution algorithms.

Categories and Subject Descriptors
1.2.6 [Artificial Intelligence]: Learning—Connectionism and neural nets; 1.2.9 [Artificial Intelligence]: Robotics—Kinematics and dynamics

General Terms
Algorithms, Performance, Experimentation

Keywords
Neuroevolution, Adaptive control, Dynamic control, NEAT

1. INTRODUCTION
In the wide field of developing neural networks, the method of evolving neural networks using genetic algorithms gained a lot of attention lately. This recent approach is very promising, as it seems to outperform regular reinforcement learning approaches by far in efficacy [2]. A very successful and well-known representative of this genre is the Neuroevolution of Augmenting Topologies (NEAT) method, which has already proven itself excelling in a wide variety of low-level control domains (e.g. pole balancing) [4, 5, 6, 7, 8].

The main intention of our work is to investigate which mechanisms guide human development of reaching behavior. Due to its good incremental performance in control tasks, we chose to experiment with NEAT as a potential candidate system, which may mimic arm reaching development. NEAT’s task was to evolve neural networks capable of controlling a simulated 3-degrees-of-freedom arm in a 2-dimensional space with gravity. To our surprise, NEAT performed far less successful than expected. Based on this finding, we started a series of slightly modified arm goal reaching tasks in order to see how NEAT’s performance scales with the different task settings and to identify those aspects of the problem that make it harder to find a solution.

The findings of these experiments are presented in this paper with the intention of contributing a partial answer to the question, which task domains are the most and which are the least suitable for NEAT’s modus operandi. We hope the results are not only helpful for deciding which learning approach to use to develop neural network structures or to find suitable control routines in general, but also for providing useful information to design the next generation of neuroevolution algorithms.

The remainder of this paper first gives a short overview of the NEAT system. Next, the goal reaching task is introduced and the simulated robotic arm model is specified. In Section 4.3 we provide the experimental results. A final discussion concludes the paper.

2. NEAT
The Neuroevolution of Augmenting Topologies (NEAT) method [8, 7] is a neuroevolution approach that combines the usual search for the appropriate connection weights with a simultaneous evolution of the optimal network structure. Starting with a minimal structure of input and output neurons without any hidden units, NEAT does not only mutate the connection weights but also adds new neurons or links between already existing units. These mutations result in networks with varying structures, incrementally gaining size through the so-called process of complexification until further added nodes or connections no longer provide performance improvements. As these networks’ chance of survival or reproduction is regulated by their fitness evaluation, which in turn not only depends on the connection weights but also on the structure, eventually the network’s weights and structure both approach an optimum.

This process of evolving structure incrementally causes three technical challenges. First, there has to be a genetic representation that allows different topologies to be crossed over in a meaningful manner. NEAT addresses this problem by usage of innovation numbers. Each genome contains a full list of connection genes and node genes, each of which being assigned a unique innovation number which in turn is inherited during crossover. This procedure allows NEAT to crossover different structures without time consuming topo-
logical analysis. Second, to be successful, NEAT has to assure that topological innovations are protected long enough to have a chance of being optimized before being discarded. This is cleverly done by speciating the population so that each individual first has to compete within its own evolutionary niche instead of with the whole population at once. Additionally, NEAT uses explicit fitness sharing as its reproduction mechanism, which results in all organisms of one species sharing their fitness, thus avoiding the risk of one species taking over the whole population. Third, the complexity of the evolved structures should be kept minimal. Therefore, NEAT uses the already mentioned method of starting with minimal networks without any hidden nodes while adding new structure incrementally through structural mutation. This process is designed to find the optimal level of complexity by letting only those structures survive that are evaluated as the most useful.

These system properties certainly play a role in NEAT’s outstanding efficacy. NEAT virtually outperforms every other method in the double pole balancing task by both speed and minimum structure [7], but also proves itself successful in a variety of impressive tasks as robotic strategy learning [8], vehicle control, collision warning or even real-time evolutionary processes in video games’ character control tasks [4].

3. DOUBLE POLE BALANCING

In order to assure proper functionality of the used NEAT implementation (ANJI version 2.01) double pole balancing experiments with and without velocity information were evaluated first.

3.1 Experimental Approach

Two poles (lengths 1 and 0.1 units) were connected to a cart by a hinge. NEAT had to keep the poles balanced by applying a force to the cart and thus moving it back and forth on its finite track. Fitness was determined by the number of timesteps both poles were considered balanced, which was defined as angles less than 36° from vertical. Initial starting position was upright for the short pole and 1° from vertical for the long pole. The task was solved perfectly if both poles were kept balanced for 100,000 timesteps. Ten controllers were evolved over 100 generations in two experimental settings each—one with velocity information (Markovian), the other without (Non-Markovian).

3.2 Results

Figure 1 shows the results for these two settings, depicting the average number of generations in which the task was solved. As can be seen, the easier Markovian task setting can be solved in less than 20 generations, whereas the Non-Markovian condition’s generations of first solution are widely distributed, thus indicating increased task difficulty. Nonetheless, these results are in line with published double pole balancing data, hence proving the general functionality of the used NEAT implementation, which in turn allowed us to proceed with our own experiments.

4. GOAL REACHING ARM TASK

In our research on human development of arm reaching, the intention is to investigate similarities and differences between the learning processes of goal-directed reaches in infants and a simulated robotic arm controlled by a neural network.

4.1 General Information

As a start, we had NEAT evolve a motor-controller for a simulated dynamic 3-degrees-of-freedom arm in a 2-dimensional space with gravity. The arm was allowed to move in a range of 180° for the shoulder joint and 360° for the elbow and wrist joints. In each generation, multiple samples with different start and goal postures were presented for an appropriate number of time steps. In each sample, the arm started with zero velocity and was supposed to move to the goal posture as fast and accurate as possible. In order to convey the desired behavior, the fitness function was defined as one minus the normed average difference between current angles and goal angles, which was weighted by the natural logarithm of the current time step and averaged over all time steps of all samples. In this way, a fitness close to one suggests optimal performance.

For each joint, three input nodes (representing the current angular speed, the current angle and the desired movement direction, in other words the difference between goal and current angle) and one output node (representing a simple motor neuron defining the muscular force and its direction) were used. The information about the current joint angles in addition to the desired movement direction was necessary because the effects of gravity are dependent of the current posture and therefore have to be counterbalanced at each position differently.

An important additional difficulty in our goal reaching arm task is the fact that the behavior of the three limbs depends on the movement and orientation of the others at all times, as they directly influence each others movement speed and direction while swinging back and forth. Nonetheless, after having heard of all the different tasks NEAT excelled in, we were quite sure it could easily solve our arm goal reaching task. This seemed right at first test trials with only two degrees of freedom and without gravitational forces,
which were solved pretty fast and accurate. However, after having switched to three degrees of freedom and gravity, suddenly NEAT was no longer able to provide sufficiently good solutions to the problem. These observations are now investigated in further detail.

4.2 Experimental Approach

In order to find out how the different aspects of our goal reaching arm task influence NEAT’s performance, we decided to separately evolve controllers for gravity turned on and off and for an arm with two and with three limbs, ergo four experimental settings. In each setting, 5 runs were evaluated, whereas each run lasted for 300 generations with a population size of 150. The three limbs of the arm have the lengths $l_1 = 0.6m$, $l_2 = 0.5m$ and $l_3 = 0.4m$ and weights $w_1 = 12kg$, $w_2 = 10kg$, $w_3 = 8kg$. The shoulder joint is restricted to values within $0^\circ$ and $180^\circ$, whereas the elbow and wrist joints are allowed to rotate full $360^\circ$—however, without the possibility of circling, which would result in one limb moving through the same 2-dimensional space occupied by another limb.

In order to maximize comparability between runs, each run was presented the same four samples for the exact same amount of time (500 time steps), which were prearranged to the start and goal postures depicted in Figure 2. To guarantee comparability between arms with two and arms with three limbs regarding the fitness function, the start and goal angles of the third joint were identical. The four samples were built out of three postures with the following angle combinations, whereas the last angle of each triplet evidently remained unused in the 2-joint condition: $[135^\circ; 90^\circ; 180^\circ]$, $[45^\circ; 270^\circ; 180^\circ]$, $[90^\circ; 180^\circ; 180^\circ]$.

All runs started with a minimal network structure of three input nodes per joint plus a bias node and one output node per joint. Input nodes were fed with the current angle a, the current angular speed v and the goal angle g minus the current angle a, which can be interpreted as the desired movement direction. The output node worked as simple motor-neuron, dictating the direction and strength of force to be applied to the joint. The fitness of each genome was evaluated by calculating the average sum of the current joint angles differences to the desired goal angles over all time steps of all samples. In order to increase the importance of goal reaching with increasing time spent on the task, the sums of joint angle differences were weighted with the natural logarithm of the number of already spent time steps, thus making it not only attractive to achieve high accuracy but also to do so in the fastest possible way.

The used implementation of NEAT is the Java based ANJI (Another NEAT Java Implementation) in the latest version 2.01. All parameters except the named ones were used with default values provided from ANJI.

4.3 Results

Five runs in each of the four test settings (two limbs with and without gravity and three limbs with and without gravity) were evaluated. Figure 3 shows the average fitness values of the four conditions during learning. Overall, in the first 100 generations fitness increases quickly whereas afterwards only incremental changes take place. In case of the two joint no gravity setting, this seems to be a ceiling effect as the behavior and thus its fitness cannot be optimized much further. Nevertheless, in case of the three joint conditions this could be interpreted as early stagnation caused by over-complexification. Furthermore, it has to be noted that even small differences in fitness values indicate huge differences in the task solution. Whereas after 300 generations both two joint settings’ results are nearly perfect, controlling the arm with satisfying accuracy (some with less than $0.5^\circ$ difference to the goal angle in the last timestep), both three joint settings fail miserably (some of them with more than $50^\circ$ difference to the goal angle in the last timestep).

Figure 4 illustrates this, showing the mean difference between current joint angles and desired goal angles in the last timestep, averaged over all samples of the best evolved individuals in each experimental setting. Unlike abstract fitness values, these errors measured in degrees make the huge difference in task performance between settings visible. While a mean joint error of $1^\circ$ in the easiest setting means great accuracy, a mean joint error of nearly $20^\circ$ in the hardest setting results in disastrous behavior. Eventually, Figure 4 shows the influence of the different task settings on the maximum fitness values during learning. Depicting the final fitness values at the time of the completed runs. Obviously, the number of limbs dramatically increases the task difficulty, while the influence of gravity basically seems to make things worse in the two joint condition while hardly being of consequence in the three joint setting.

5. DISCUSSION

The results described in this paper show that although NEAT is a very powerful tool for developing neural networks in a variety of tasks that on the first glimpse look much more impressive, there are some aspects of seemingly much simpler problems that render NEAT underpowered. To be correct, retrospectively our goal reaching arm task probably is a very difficult task in general, not only for NEAT. There are two major difficulties in our goal reaching arm task. First, the neural network has to compensate gravity. As gravitational forces are dependent on the current pose.
Figure 3: The evolution of the average fitness values over time shows that two degrees of freedom arms are not only easier to control but the necessary control routines can also be learned faster.

Figure 4: The best fitness values produced after 300 generations show that three degrees of freedom are significantly harder to control than two degrees of freedom.

ture angles, even standing still without any movement can be challenging, because for each posture this means different gravitational forces which must be withstood to stay in balance.

Being a tough task for the first limb, this even becomes harder with additional limbs, as their position in space and therefore gravity’s influence on them cannot be estimated by the current joint angle itself, but by combining the information of the current joint angle of this joint and of the higher order joint or joints. This thought introduces the second major problem in our task, the inter-dependency of limbs. To decide the ideal amount of force to apply to a joint, not only the three inputs for that specific node are required, but also information of the other joints’ input nodes. Moreover, the other joints’ influence is dependent on the order of the joints. For example, the second limb is much more affected by the first limb (which influences the second joint’s position in space and its movement speed) than by the third (which only affects the second joint’s movement speed). These non-linear dependencies make the seemingly easy task of moving the arm from one posture to another rather complicated. These considerations also suggest that it is not astonishing that the number of limbs has a greater effect on task difficulty than gravity—more joints yield more non-linear interactions and thus require more sophisticated networks.

More complex networks on the other hand require much more computation time and have a higher risk of getting stuck while evolving. NEAT’s incremental complexification approach seems to be much better suited for finding solutions that require small network structures than finding complex ones, since incremental changes in large networks may not be enough to overcome local optima for the higher purpose of approaching the global optimum—especially if structural improvements in the long run result in lower fitness ratings in the short run. Nonetheless, it has to be noted that our observations are restricted to NEAT itself and therefore cannot be regarded as representative for all neuroevolution algorithms. Although only measured by benchmarking results of pole balancing tasks, recent studies also showed great potential in other neuroevolution procedures, which partially even outperform NEAT [2, 1]. Future research has to provide further insights into the question if these alternative algorithms still perform better when used for tasks more difficult than pole balancing, such as the complex network structures necessary in the investigated goal reaching arm task.

Miikkulainen and Kohl proposed the idea that NEAT and other neuroevolution algorithms have difficulties with problem domains that possess a fractured decision space, which is loosely defined as a space where adjacent states require radically different actions [3]. In contrast to traditional benchmarking tasks such as double pole balancing, in which the correct action for one state is similar to the correct action for neighboring states, the optimal action in a fractured domain may change abruptly and discontinuously. Although our goal reaching arm task seems to be a non-fractured domain at first glance, the above stated two major difficulties perfectly match the idea of fractured domains: the best action for one limb is dependent not only on gravitational forces of its current position but also of the other limbs’ movements and positions, thus requiring different actions despite similar states. For optimal action choice, the network has to take a much broader view of the state and action space, which is also a characteristic of fractured domains. Miikkulainen concludes that it is very hard for neuroevolution algorithms to solve such problems because much more sophisticated structures with multiple layers are required. These are unlikely to be produced by the process of incremental complexification, since such structural changes may result in worse fitness evaluations for a long period of time before ultimately the structure is completed by adding further nodes in the right spaces.
6. CONCLUSION

NEAT and neuroevolution algorithms in general have proved themselves successful in a variety of tasks. However, there are tasks that emerge as surprisingly difficult for these approaches. One example is the goal reaching arm task presented in this paper, which was evaluated and reviewed in detail in the intention of finding possible sources of its difficulty. Based on these experimental results, assumptions about the causality between certain aspects of the problem domain and NEAT’s unexpected insufficient performance were expressed. Eventually, these findings may be helpful for a better understanding of NEAT’s limitations regarding some types of problem domains, thus providing information about which problem domains the next generation of neuroevolution has to address in order to succeed.

Acknowledgments

The authors acknowledge funding from the Emmy Noether program of the German research foundation (grant BU 1335 / 3-1) and like to thank their colleagues at the department of psychology and the COBOSLAB team.

7. REFERENCES