Mirosław Łącki
Akademia Morska w Gdyni

REAL-TIME TRAINING ALGORITHMS IN NEUROEVOLUTIONARY NAVIGATIONAL DECISION SUPPORT SYSTEM

ABSTRACT

The paper presents the idea of using advanced machine learning algorithms to aid decision making in ship manoeuvring in real time. Evolutionary neural networks are used in this purpose. In the simulated model of manoeuvring ship a helmsman is treated as an individual in population of competitive helmsmen, which through environmental sensing and evolution processes learn how to navigate safely through restricted waters.

Keywords:
machine learning, artificial intelligence, evolutionary neural networks, marine navigation, routing and manoeuvring, safe ship control, computer simulation.

INTRODUCTION

Research and development of navigational decision support systems is intensively growing these days. Such advanced systems add many improvements to complex decision making processes: they speed up the process of decision making; decrease the amount of human errors during data analysing; speed up learning and improve effectiveness of training courses and help automate some aspects of complex decision making processes that occur during manoeuvring a vessel on restricted waters. Increasing computational efficiency of personal computers allow to implement complex artificial intelligence methods and algorithms into that systems.

One of the main tasks in Artificial Intelligence is to create the advanced systems that can effectively find correct answers for given problems and improve it over time. Intelligent autonomous units used in these systems can quickly adjust their activity to current situation, i.e. change their behaviour based on interactions with the environment (fig. 1), become more efficient over time, and adapt to new situations as they occur.

Evolutionary artificial neural networks, which is evolving neural networks with genetic algorithms, has been highly effective in advanced tasks, particularly those with continuous hidden states [6] and in the real-time learning systems [5]. Neuroevolution gives an advantage from evolving neural network topologies along with weights which can effectively store action values, related to state vector, in machine learning tasks.

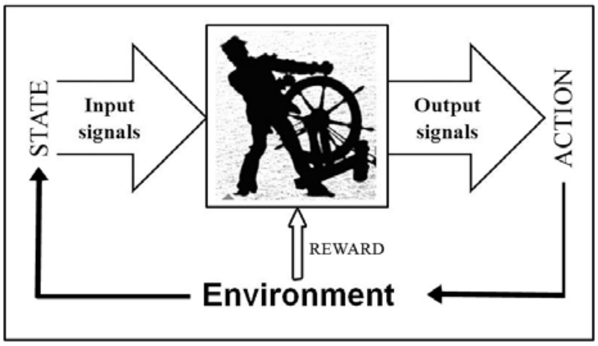

Fig. 1. Interaction of helmsman with an environment

Source: own study.

The main idea of using evolutionary neural networks in ship handling is based on evolving population of helmsmen. Learning process for simpler tasks also can be performed using classic approach, like Temporal Difference Reinforcement Learning [4, 16, 18], RL with Eligibility Traces [14], sparse coarse coding [17] or neural network with fixed structures (fig. 2).

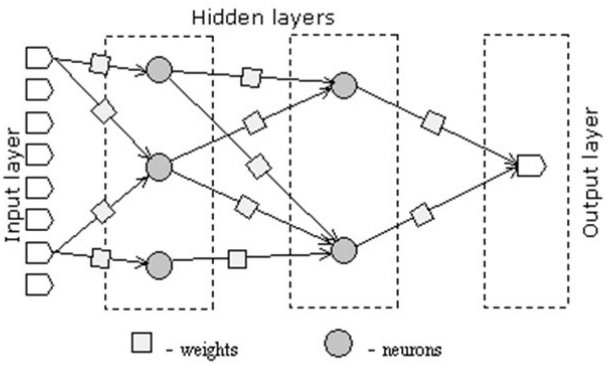

Fig. 2. General artificial neural network topology

Source: own study.

The neural network is the helmsman's brain making him able to make decisions based on actual situation which is represented by a vector of input signals. In every time step the network calculates its output from signals received on the input layer. These input signals are calculated from current situation of the environment (in this case: vessel manoeuvring in the coastal area). The main goal of the individuals in population is to maximize their fitness value. This value is calculated from helmsman behaviour during simulation. The best-fitted individuals become parents for next generation.

NEUROEVOLUTION OF ARTIFICIAL NEURAL TOPOLOGIES

Topology and Weight Evolving Artificial Neural Networks (TWEANNs) [5] have the advantage over neural networks with fixed structures that the correct topology need not be known at the design stage prior to evolution. NeuroEvolution of Augmenting Topologies (NEAT) is unique among other TWEANNS in that it begins evolution with a population of minimal networks and adds nodes and connections to them over generations, allowing complex problems to be solved gradually based on simple ones [7].

The modified NEAT method is based in four fundamental rules which deal with challenges that exist in evolving efficient neural network topology:

— begin with a minimal structure and add neurons and connections between them incrementally to discover most efficient solutions throughout evolution;
— breed disparate topologies in a meaningful way by matching up genes with the same historical markings;
— separate each innovative individual into a different species to protect it disappearing from the population prematurely;
— reduce oversized topologies by removing neurons and connection between them to provide and sustain good overall performance of a whole population of helmsmen.

GENETIC ENCODING

A flexible genetic encoding is required for meaningful evolution of neural structures. Dynamic and expandable representation of network topology allows it to increase its complexity and maintain its performance in given task [2]. Genes are grouped in two genomes: structural genome and connection genome (fig. 3). A structural genome in NEAT includes a number of inputs, neurons and outputs. A connection genome contains a list of connection genes, each of which refers to two nodes being connected with specified weighted value. Each connection gene has its innovation number which allows finding corresponding genes during crossover.

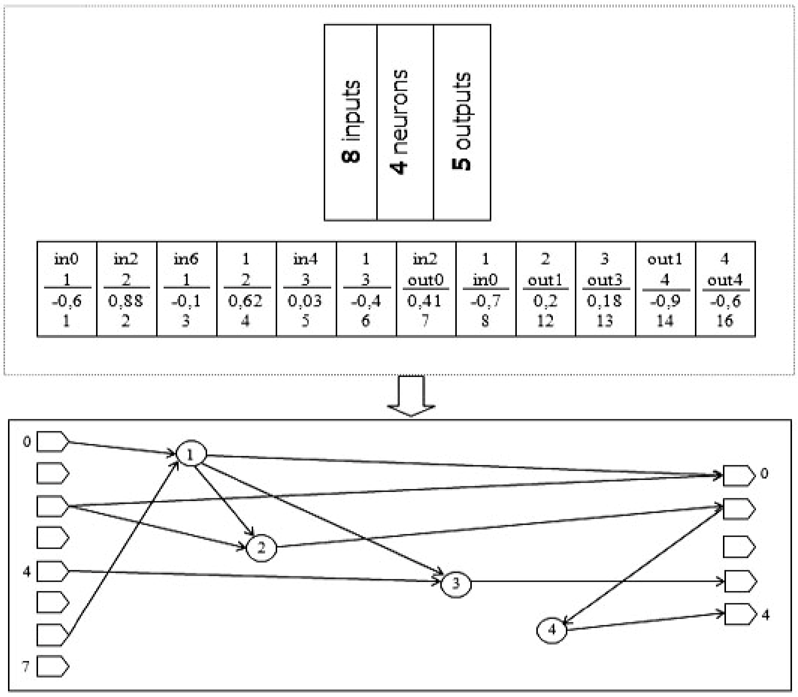

Fig. 3. Genotype and phenotype of evolutionary neural network

Source: own study.

In this approach each connection gene specifies the output layer and node, the input layer and node, the weight of the connection, and an innovation number, which allows finding corresponding genes during crossover.

MUTATION

Mutation in evolutionary neural networks can change both connection weights and network structures. Connection weights mutate as in most neuroevolutionary systems, with each connection either perturbed or not (fig. 4). Structural mutations, which form the basis of complexity, occur in two ways. Each mutation expands the size of the genome by adding genes or reduces it by removing genes from offspring chromosome.

In the add_connection mutation, a single new connection gene is added connecting two previously unconnected nodes. In the add_node mutation, an existing connection is split and the new node placed where the old connection used to be. The old connection is disabled and two new connections added to the genome. This approach allows changing topology slightly, without significant impact on efficiency of current topology but with possibility to add new connections to that new node in the future.

In modified NEAT method the remove_connection mutation removes single connection gene and the remove_node mutation removes single hidden neuron and all connection genes related to it.

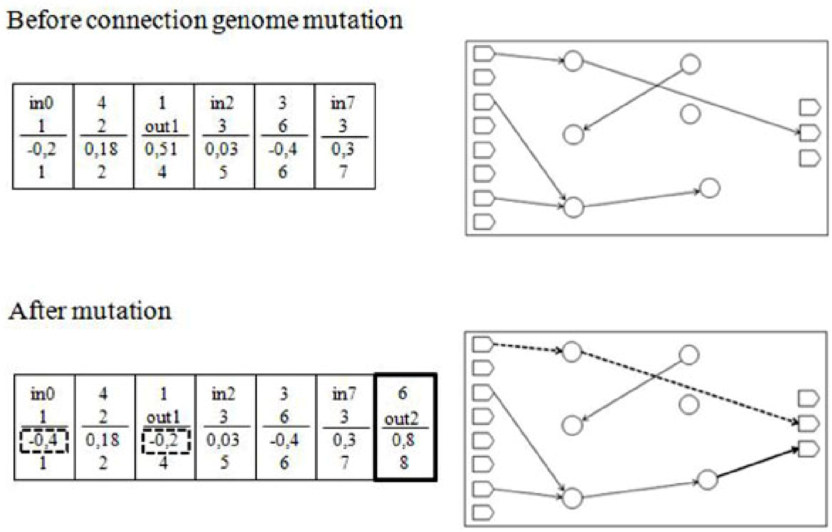

Fig. 4. An example of weights and connection mutation. In this case two connection genes had their weights mutated, while the new connection gene no. 8 has been added to the genome

Source: own study.

CROSSOVER

The system knows exactly which genes match up with which through innovation numbers. Genes that do not match are either disjoint or excess, depending on whether they occur within or outside the range of the other parent's innovation numbers.

In crossing over operation, the genes with the same innovation numbers are lined up. The offspring is then formed in one of three ways:

— in uniform crossover: matching genes are randomly chosen for the offspring genome, with all disjoints and excesses from both parents;
— in blended crossover: the connection weights of matching genes are averaged;
— in elite crossover: disjoints and excesses are taken from more fit parent only, all redundant genes from less fit parent are discarded. All matching genes are averaged.

These types of crossover were found to be most effective in evolution of neural networks in extensive testing compared to one-point crossover [6].

Disabled genes have a chance of being re-enabled during mutation, allowing networks to make use of older genes once again.

Evolutionary neural network can keep historic trails of the origin of every gene in the population, allowing matching genes to be found and identified even in different genome structures. Old behaviours encoded in the pre-existing network structure have a chance to not to be destroyed and pass their properties through evolution to the new structures, thus provide an opportunity to elaborate on these original behaviours.

Through mutation, the genomes in modified NEAT will gradually get larger for complex tasks and lower their size in simpler ones. Genomes of varying sizes will result, sometimes with different connections at the same positions. Any crossover operator must be able to recombine networks with differing topologies, which can be difficult. Historical markings represented by innovation numbers allow NEAT to perform crossover without analysing topologies. Genomes of different organizations and sizes stay compatible throughout evolution, and the variable-length genome problem is essentially solved. This methodology allows NEAT to increase complexity of structure while different networks still remain compatible.

Additionally different sizes and structures of networks group their genetic material into species.

Speciation of population can be seen as a result from the same process as adaptation [1], natural selection exerted by interaction among organisms, and between organisms and their environment [15].

Divergent adaptation of different populations would lead to speciation. Speciation of the population assures that individuals compete primarily within their own niches instead of competition within the whole population. In this way topological innovations of neural network are protected and have time to optimize their structure before they have to compete with other experienced helmsmen in the population [12].

Generally, during species assigning process, as described in [11], when a new helmsman appears in population, its genome must be assigned to one of the existing species or, if it is too innovative comparing to any other individuals, the new species is created.

Compatibility of genome g with particular species s is estimated accordingly to value of distance between two individuals which is calculated with formula 1:

$$\delta = \frac{c_1 E}{N} + \frac{c_2 D}{N} + c_3 \overline{W}, \tag{1}$$

where:

c_1, c_2, c_3 — weight (importance) coefficients;
E — number of excesses;
D — number of disjoints;
W — average weight differences of matching genes;
N — the number of genes in the larger genome.

There must be estimated a compatibility threshold δ_t at the beginning of the simulation and if $\delta \leq \delta_t$ then genome g is placed into this species. One can avoid the problem of choosing the best value of δ by making δ_t dynamic. The algorithm can raise δ_t if there are too many species in population, and lower δ_t if there are too few.

DECISION MAKING SUPPORT WITH EVOLUTIONARY NEURAL NETWORKS

The main goal of this research is to simulate a situation of ship manoeuvring through a restricted coastal area and improve decision making process in real time with Evolutionary Neural Network.

Safe navigation task for artificial helmsman can be described in many ways [8, 9, 10, 13]. Most important is to define proper state vector from available wide range of data signals and arbitrary determine fitness function values received by the helmsman. Fitness function determines the quality of each individual. Subsequently it defines helmsman's ability to sail safely toward designated goal.

In the simulation of safe passage through restricted waters with simplified channel structure there are no moving vessels in the area (fig. 5). Helmsman observes current situation which is encoded as input signals for his neural network and calculates the best, in his opinion, available action.

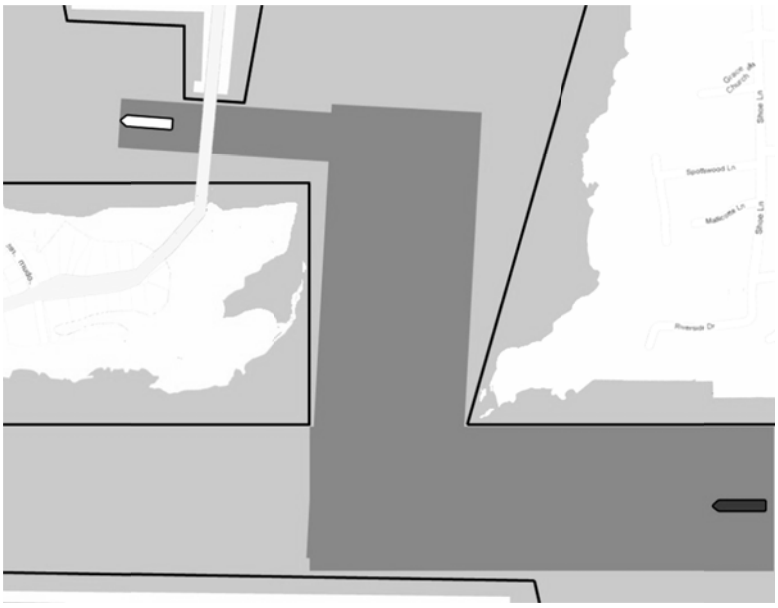

Fig. 5. Model of coastal environment;
manoeuvring area available for simulated vessel is shaded with dark grey colour

Source: own study.

The main input signals are gathered from on-board devices (GPS, Gyro, etc.) and available navigational support systems (ECDIS, AIS, etc.). Basic set of data needed for effective performance of real-time neuroevolutionary support system is as follows:

— ships course over ground;
— ships angular velocity;
— the ship is on the collision course with an obstacle;
— distance to collision;
— the ship is approaching destination;
— ships angle to destination;
— the ship is heading out of the area;
— distance to current channels' borders;
— distance to next channel;
— ship is heading on goal;
— distance to goal.

All the input signals are encoded either binary (0 or 1) or as a real values between 0 and 1. Some of the input signals may be calculated as multi-criteria values [3].

Neural network output values are signals for rudder angle (δ) and thrust control (rpm). It is crucial for usefulness of simulation to determine the number of neural network outputs.

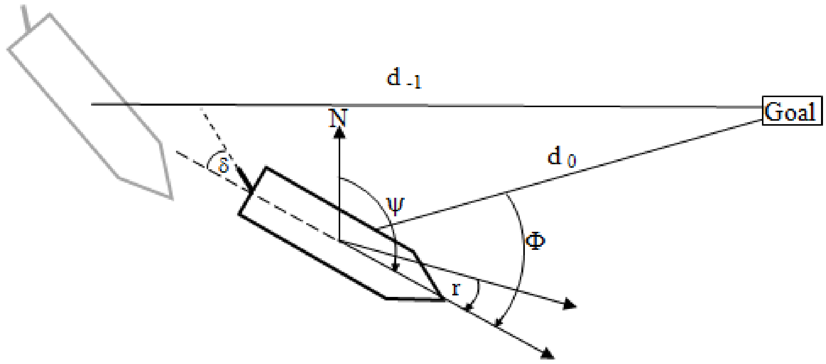

Fig. 6. Example of considered data signals in ship handling with ENN

Source: own study.

More outputs mean more calculations but on the other hand better accuracy and usefulness of designed decision support system. Additionally too many inputs and outputs may cause learning process to complicate, thus making a helmsman unable to quickly adapt to new situations. This accuracy vs. performance dilemmas were examined extensively in previous works [9, 10, 12, 13].

The fitness value of an individual is calculated from arbitrary set action values, i.e.: –1 for increase of the distance to goal in every time step, –10 when ship is on the collision course (with an obstacle or shallow waters), +10 when she's heading to goal without any obstacles on course, –100 when she hits an obstacle or run aground, +100 when ship reaches a goal and –50 when she depart from the determined restricted channel area in any other way, etc.

Sum of gained fitness values defines the quality of helmsmen and a chance of passing his genes to next generation. Selection in real-time NEAT (rtNEAT) exchange genetic material gradually [5]. It means that population is not replaced as a whole but only the worst fit individuals are replaced by offspring of the best fit ones. That allows an evolution to look more naturally. An example of gradual exchange of genes in population is presented on figure 7.

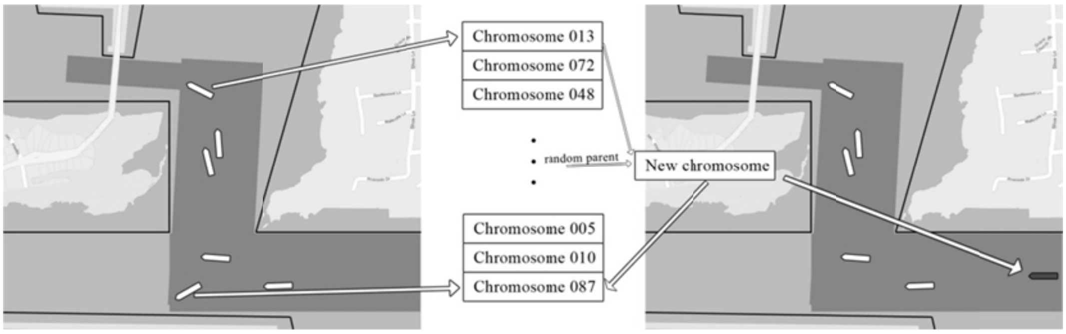

Fig. 7. Example of elitist selection and reproduction among individuals in ranked population
Source: own study.

In this example of elitist selection the best fitted helmsman (chromosome 13) exchanges his genetic material with one random individual. New chromosome gains its material through cross-over and changes it slightly with mutation operations. Then the worst helmsman (chromosome 87) is removed from population and new offspring takes his place. His vessel is placed on starting position in the restricted area and new helmsman starts to compete with older ones to achieve the best position in fitness ranking.

REMARKS

Neuroevolution approach to machine learning tasks can effectively improve learning and decision making processes in ship handling. Neural networks with evolving topology and weights based on modified NEAT can increase learning speed of automated helmsman and complexity of considered model of ship manoeuvring in restricted waters. One can use simulation models with much larger state space than it was possible in classic state machine learning algorithms without neural network function approximations. Properties of evolution process of artificial neural networks allow using this method in on-line training in real-time navigational decision support systems.

REFERENCES

[1] Beyer H.-G., Schwefel H.-P., *Evolution strategies — A comprehensive introduction*, 'Natural Computing', 2002, 1(1), pp. 3–52.

[2] Braun H., Weisbrod J., *Evolving Feedforward Neural Networks*, International Conference on Artificial Neural Networks and Genetic Algorithms, Innsbruck, Springer-Verlag, 1993.

[3] Filipowicz W., Łącki M., Szłapczyńska J., *Multicriteria decision support for vessels routing*, European Safety and Reliability, A. A. Balkema, Taylor & Francis Group, Gdańsk 2005.

[4] Kaelbling L. P., Littman M. L., Moore A. W., *Reinforcement Learning: A Survey*, 'Journal of Artificial Intelligence Research', 1996, cs.AI/9605, pp. 237–285.

[5] Kenneth O. S., Bryant B. D., Risto M., *Real-time neuroevolution in the NERO video game*, 'IEEE Transactions on Evolutionary Computation', 2005, 9(6), pp. 653–668.

[6] Kenneth O. S., Risto M., *Efficient evolution of neural network topologies*, Proceedings of the Evolutionary Computation on 2002. CEC '02, Proceedings of the 2002 Congress, IEEE Computer Society, 2002, Vol. 2, pp. 1757–1762.

[7] Kenneth O. S., Risto M., *Efficient Reinforcement Learning Through Evolving Neural Network Topologies*, Proceedings of the Genetic and Evolutionary Computation Conference, Morgan Kaufmann Publishers Inc., 2002, pp. 569–577.

[8] Łącki M., *Machine Learning Algorithms in Decision Making Support in Ship Handling*, TST Conference, WKŁ, Katowice — Ustroń 2007.

[9] Łącki M., *Neuroevolutionary approach towards ship handling*, TST Conference, WKŁ, Katowice — Ustroń 2008.

[10] Łącki M., *Ewolucyjne sieci NEAT w sterowaniu statkiem*, [in:] *Inżynieria wiedzy i systemy ekspertowe*, Akademicka Oficyna Wydawnicza EXIT, Warszawa 2009, pp. 535–544.

[11] Łącki M., *Speciation of population in neuroevolutionary ship handling*, [in:] *Marine Navigation and Safety of Sea Transportation*, A. Weintrit (ed.), CRC Press/Balkema, Taylor & Francis Group, Boca Raton — London — New York — Leiden, 2009, pp. 541–545.

[12] Łącki M., *Model środowiska wieloagentowego w neuroewolucyjnym sterowaniu statkiem*, 'Zeszyty Naukowe' AM w Gdyni, 2010, No 67, pp. 31–37.

[13] Łącki M., *Wyznaczanie punktów trasy w neuroewolucyjnym sterowaniu statkiem*, 'Logistyka', 2010, No 6.

[14] Singh S., Sutton R., *Reinforcement learning with replacing eligibility traces*, 'Mach. Learn.', 1996, 22(1–3), pp. 123–158.

[15] Spears W. M., *Speciation using tag bits*, Handbook of Evolutionary Computation, 1998.

[16] Sutton R., Barto A., *Reinforcement Learning: An Introduction*, The MIT Press, 1998.

[17] Sutton R. S., *Generalization in reinforcement learning: Successful examples using sparse coarse coding*, 1996.

[18] Tesauro G., *Temporal difference learning and TD-Gammon*, 'Communications of the ACM', 1995, 38(3), pp. 58–68.

ALGORYTMY SZKOLENIA W CZASIE RZECZYWISTYM W NEUROEWOLUCYJNYM SYSTEMIE WSPARCIA PODEJMOWANIA DECYZJI NAWIGACYJNYCH

STRESZCZENIE

Artykuł przedstawia koncepcję wykorzystania zaawansowanych algorytmów uczenia się maszyn dla wsparcia podejmowania decyzji manewrowania okrętem w czasie rzeczywistym. Do tego celu wykorzystywane są ewolucyjne sieci neuronowe. W symulowanym modelu manewrowania okrętem sternik jest traktowany jako jednostka w populacji konkurencyjnych sterników, którzy poprzez wyczuwanie środowiskowe i procesy ewolucyjne uczą się jak prowadzić nawigację bezpiecznie po ograniczonych akwenach.

Słowa kluczowe:

sztuczna inteligencja, ewolucyjne sieci neuronowe, nawigacja morska, wyznaczanie tras, manewrowanie, sterowanie bezpieczeństwem okrętu, symulacja komputerowa.