Jozef Waldhauser

Úloha 6

Evolučné učenie neurónovej siete

Obsah

[Úloha 4 – zadanie 2](#_Toc197116812)

[Úloha 4 – riešenie 2](#_Toc197116813)

[Design algoritmu 3](#_Toc197116814)

[Redesign sofistikovanej fitness funkcie 3](#_Toc197116815)

[Implementácia algoritmu 4](#_Toc197116816)

[Implementácia DEAP: 5](#_Toc197116817)

[Analýza 6](#_Toc197116818)

[Výstupy úlohy 4 7](#_Toc197116819)

[Podrobnejšia analýza fitness funkcie 8](#_Toc197116820)

[Experimenty s parametrami 10](#_Toc197116821)

[Mutácia 10](#_Toc197116822)

[Kríženie 10](#_Toc197116823)

[Súboj / Tournament 10](#_Toc197116824)

[Elitizmus 11](#_Toc197116825)

[Teoretické prepojenie 11](#_Toc197116826)

[Mutácia 11](#_Toc197116827)

[Kríženie 12](#_Toc197116828)

[Súboj / Tournament 12](#_Toc197116829)

[Elitizmus 12](#_Toc197116830)

[Fitness funkcia 12](#_Toc197116831)

[Záver 12](#_Toc197116832)

# Úloha 6 – zadanie

1. naimplementujte vlastní sentorické funkce, kterými budou agenti vnímat prosředí
2. navrhněte funkcinn\_function(inp, wei), která na základě vstupů inp a vektoru vah wei neuronové sítě (genomu neuronové sítě / zlinearizované sítě) provede výpočet výstupů této sítě
3. navrhněte funkci nn\_navigate\_me(me, inp), která pro agenta me a jeho senzorické vstupy inp provede výpočet jeho pohybu na základě jeho vnitřního genomu reprezentujícího jeho neuronovou síť (zda pojede up, down, left, right)
4. doimplementujte mechanismus výpočtu fitness jedinců ve funkci handle\_mes\_fitnesses(mes), která dostane seznam jedinců mes a vypočítá jim fitness na základě jednoho herního kola
5. nastavte vhodně parametry evolučního algoritmu (mutace, crossover, selekce)

**xxxxx**

# Úloha 6 – zdroje

<https://www.youtube.com/watch?v=aircAruvnKk>

<https://www.youtube.com/watch?v=jmmW0F0biz0>

<https://www.youtube.com/watch?v=rEDzUT3ymw4>

<https://www.youtube.com/watch?v=UZDiGooFs54>

citim sa jak na biologii

# Asdfg plajesjholdern

**1. Enhanced Sensory Functions**

* Expanded from 4 to 6 inputs for better environmental awareness
* Added obstacle detection to allow for more complex navigation
* Added Manhattan distance sensor for better goal-oriented behavior
* Normalized all sensors to [0,1] range for stable neural network processing

**2. Improved Neural Network Function**

* Optimized weight organization and neural processing
* Added softmax activation for output layer to better represent action probabilities
* Increased hidden layer size from 8 to 10 neurons for better learning capacity

**3. Enhanced Navigation Logic**

* More robust movement decisions based on neural network outputs
* Added obstacle avoidance logic
* Improved boundary checking

**4. Advanced Fitness Calculation**

* Rewards reaching the goal quickly with a significant bonus
* Properly penalizes loop behavior to encourage efficient paths
* Evaluates each agent across multiple game rounds from different starting positions
* Balances exploration (distance-based fitness) with exploitation (goal-reaching bonus)

**5. Optimized Evolutionary Parameters**

* Reduced mutation rate (0.3 → 0.2) for more stable evolution
* Added separate gene mutation probability (MUTATION\_INDPB = 0.1)
* Increased tournament size (3 → 5) for stronger selection pressure
* Doubled elitism size (10 → 20) to better preserve good solutions

**6. Major Bug Fixes**

* Fixed the AttributeError by properly implementing the generational loop
  + Replaced algorithms.eaSimple with custom evolution implementation
  + Properly tracked and updated fitness values
  + Fixed population handling between generations

**7. Added Visualization Features**

* Created a function to visualize the best agent's behavior
* Added path visualization to see how agents navigate to the goal
* Shows multiple trials to verify robustness

**How to Run**

Simply execute the script and it will:

1. Initialize a population of neural networks
2. Train them through evolutionary learning
3. Display a fitness-over-time graph
4. Visualize the best agent's navigation behavior