Gra "Snake" sterowana sztuczną siecią neuronową. Specyfikacja Implementacyjna

Patryk Peszko

25 marca 2021

Spis treści

1 Wstęp

Dokument ten jest kontynuacją dokumentu pt. "Gra "Snake" sterowana sztuczną siecią neuronową. Specyfikacja Funkcjonalna". W tym dokumencie postaram się opisać sposób w jaki mam zamiar napisać ten program, jak będę go wersjonować oraz z jakich modułów będzie się on składać.

2 Środowisko deweloperskie

2.1 Parametry komputera

- laptop Lenovo Ideapad 720s-14IKB z procesorem m Intel(R) Core(TM) i7-8555U CPU @ 1.80Ghz 2.00Ghz z 8 GB pamięci RAM, działającym na systemie Windows 10 Home wersja 1909

2.2 Programy wykorzystywane podczas projektu

- IntelliJ IDEA Community Edition 2019.3.2 x64

2.3 Środowisko kompilacji

- Microsoft Windows 10 Home wersja 20H2

2.4 Standard

Projekt pisany będzie zgodny z standardem Java 13.

3 Zasady wersjonowania

Wersje plików wrzucanych na gita będę zaznaczać w tag'ach. Krótkie komentarze, pisane w języku polskim, na temat zmian wprowadzonych w kodzie, będą się znajdować w commit'ach:

- "v n" wersja stabilna gdzie n = 0, 1, 2, ...
- "v n.m" wersja z drobnymi poprawami gdzie m = 1, 2, ...
- "v FINAL" ostateczna wersja programu

4 Struktura projektu

4.1 Opis modułów

Program będzie składał się z wymienionych modułów:

- Neuron.java - pojedynczy neuron
- Layer.java - warstwa sztucznej sieci neuronowej
- NeuronNetwork.java - cała sieć neuronowa
- ListsHolder.java - interfejs
- SnakeRunByNeuronNetwork.java - klasa zarządzająca przebiegiem programu
- ReadFromFile.java - klasa wczytująca z pliku
- WriteToFile.java - klasa zapisująca do pliku
- Cell.java - komórka planszy
- GameField.java - pole gry
- Apple.java - jabłko, dziedziczy po Cell.java
- Graphics.java - zarządza GUI
- KeyOperations.java - klasa obsługująca klawiaturę
- Snake.java - wąż
- Game.java - zarządza grą

4.2 Struktura

```
Snake
├── specyfikacje
│   ├── funkcjonalna
│   │   └── specyfikacja_funkcjonalna.tex
│   └── implementacyjna
│       ├── siec.png
│       ├── struktura.png
│       └── specyfikacja_implementacyjna.tex
└── src
    ├── main
    │   └── java
    │       ├── Neuron.java
    │       ├── Layer.java
    │       ├── NeuronNetwork.java
    │       ├── ListsHolder.java
    │       ├── SnakeRunByNeuronNetwork.java
    │       ├── ReadFromFile.java
    │       ├── WriteToFile.java
    │       ├── Cell.java
    │       ├── GameField.java
    │       ├── Apple.java
    │       ├── Graphics.java
    │       ├── KeyOperations.java
    │       ├── Snake.java
    │       └── Game.java
    ├── resources
    │       ├── apple.png
    │       ├── input.txt
    │       └── output.txt
    └── test
        └── java
            ├── SnakeRunByNeuronNetworkTest.java
            ├── NeuronTest.java
            ├── NeuronNetworkTest.java
            ├── CellTest.java
            ├── GameFieldTest.java
            ├── ReadFromFileTest.java
            ├── SnakeTest.java
            └── GameTest.java
```

5 Algorytmy i struktury danych

Sieć neuronowa będzie przechowywać warstwy w liście liniowej. Warstwy sieci neuronowej będą przechowywać neurony w liście liniowej. Czyli cała sieć będzie przechowywana w liście linowej list liniowych. Komórki, z których będzie się składać wąż, również będą przechowywane w liście liniowej. Listy

liniowe i stałe zmienne będą przechowywane w interfejsie.

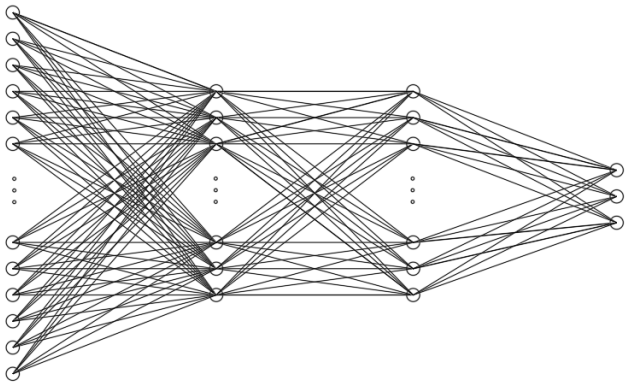
6 Sztuczna sieć neuronowa

Zaimplementuję jednokierunkową sieć neuronową. Wszystkie neurony będą mieć tę samą funkcję aktywacji ReLu (rektyfikowana jednostka liniowa). Pierwszy neuron wyjściowy będzie odpowiedzialny za skręcenie w lewo. Drugi za pójście prosto. Trzeci za skręcenie w prawo. Sieć będzie wybierać neuron o największej wartości i obierać kierunek odpowiadający danemu neuronowi.

6.1 Struktura sieci

Sieć neuronowa będzie zbudowana z 4 warstw. Pierwsza warstwa 24 neuronów, dwie kolejne będą miały po 18 neuronów a ostatnia będzie się składać z 3 neuronów wyjściowych.

6.2 Wizualizacja struktury sieci



6.3 Nienadzorowane nauczanie sieci

Pierwsza generacja węży będzie mieć sieci o współczynnikach losowych. Po zakończeniu gry przez wszystkie węże w danej generacji będzie wybierany najlepszy wąż. Każde jabłko zjedzone przez węża będzie warte 50 punktów a każdy ruch 1 punkt. Wąż będzie startował z możliwością wykonania 100 ruchów, za każde zjedzone jabłko będzie dostawać kolejne 50 ruchów. Po wybraniu najlepszego węża zostanie stworzona kolejna generacja węży, których sieci będą zawierać współczynniki losowe z zakresu <0.9x, 1.1x>, gdzie x jest współczynnikiem najlepszego węża z poprzedniej generacji.