

Využití umělé inteligence a genetického algoritmu pro autonomní řízení

Marek Bečvář

Červen 2020

1. O projektu

Tento projekt byl zpracován v roce 2020 jako maturitní práce 6.ročníku víceletého gymnázia Boženy Němcové v Hradci Králové.

Hlavní myšlenkou je využití strojového učení umělé inteligence genetickým algoritmem s cílem projetí uživatelem nastavené tratě. Výsledek snažení je postupně promítán do grafu. Schopnost by se měla, i po dosažení cíle (projetí tratí), stále rozvíjet.

2. Mechanismy programu

2.1 Umělá inteligence

Pojem **umělá inteligence** je dnes člověku prezentován v trochu zkreslené formě. Lidem se mohou vybavit některé sci-fi filmové scény s roboty, ale to není úplně přesné. Ve slově *intelligence* se doopravdy skrývá řada výpočtů zpracovávajících vstupní data na požadovaný výstup. Tyto výpočty se provádějí v předem definované struktuře tzv. **neuronové síti**.

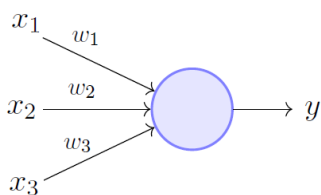


Schéma 1: Perceptron

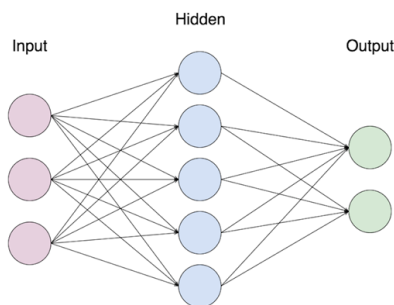


Schéma 2: Neuronová síť

Neuronová síť je modelována s inspirací v mozkových neuronech. Každý bod je považován za samostatný neuron a má řadu spojení, z biologie *axonů*. Právě tato spojení jsou důležitou funkční složkou neuronových sítí. V programu tedy vždy uchováváme číselnou hodnotu každého z nich. Tato hodnota je obvykle reálné číslo v rozsahu od -1.0 do 1.0. Neurony dělíme do vrstev (VSTUP, SKRYTÁ, VÝSTUP). Data ze sledovaného prostředí přenášíme do vstupní vrstvy a výsledek pozoruje na vrstvě výstupní. Existuje pravidlo, že každý neuron z nižší vrstvy je spojen se všemi neurony ve vyšší vrstvě. Nejjednodušším znázorněním neuronu je Schéma 1, skládající se z 3 vstupů a 1 výstupu. Jak se ale tvoří číselné hodnoty?

$$\sigma\left(\sum_{k=1}^n x_k * w_k\right) \quad (1)$$

$$0.5*0.1+1.0*0.5+0.25*0.3=0.625 \rightarrow \sigma(0.625)=0.625 \quad (2)$$

kde σ je *aktivační funkce*. Tyto funkce pouze upravují možný rozsah výstupních hodnot. Takto získáme finální hodnotu na jednom neuronu.

Příkladem takové aktivační funkce může být třeba **RELU** (Rectified Linear Unit). Ta výstupní hodnoty změní tak, že kladné nechá beze změny a záporné změní na nulu. Tyto změny pomáhají při přenosu signálu mezi neurony v síti.

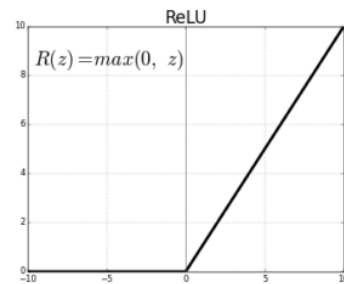


Schéma 3: RELU aktivační funkce

Složitější neuronové sítě jsou už jen opakováním tohoto zavedeného postupu. Dále se pro sítě tohoto typu zavádí jméno **hluboká neuronová síť** (*z ang. deep neural network*). To označuje schéma, ve kterém je použita více než jedna skrytá vrstva.

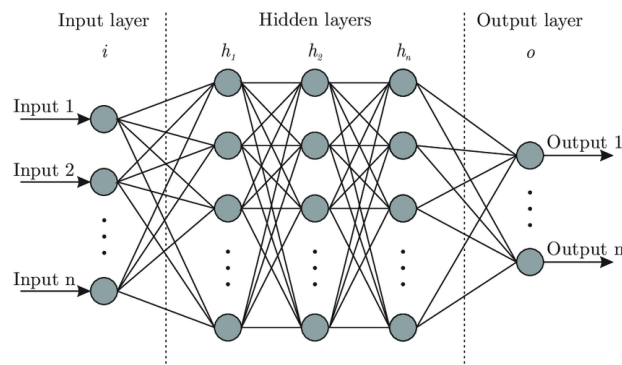


Schéma 4: Hluboká neuronová síť

Při práci s neuronovými sítěmi, abychom zvětšili přesnost výpočtů, upravujeme vstupy (první řadu neuronů) do určitého číselného rozsahu. Poté necháme proběhnout všechny matematické operace uvnitř sítě a sledujeme výsledek. To jak se program bude při daném výsledku chovat je opět pouze na volbě autora programu.

Poté nastává **proces učení**, při kterém upravujeme hodnoty všech spojení v síti tak, aby při určitých vstupních hodnotách nastal požadovaný výstup. Pro učení existuje řada algoritmů založených na různých matematických úpravách. Pro moji síť jsem ale zvolil postup učení, inspirovaný přírodou, tzv. **genetický algoritmus** (*"O původu druhů" Charles Darwin - evoluční teorie*).

Samotná architektura (počet neuronů v každé z vrstev) je pro funkčnost velmi důležitá a neexistuje předem daný, jednoznačně správný postup. Tato část je tedy často o spoustě drobných úprav při hledání dobře pracující sítě.

Architektura, která je použita v prezentovaném programu, byla docílena řadou testů různých sítí proti sobě.

Finální síť použitá v programu je ve schématu [6, 8, 4, 2]

5 vstupů měřících vzdálenost + 1 vstup měřící čas

1 výstup pro rychlost + 1 výstup pro zatačení

2.2 Genetický algoritmus

Genetický algoritmus má v informatice mnoho využití při řešení logických problémů. Ve svém projektu jsem ho použil na úpravu spojů v neuronové síti. Algoritmus je také známý svými procesy, které přímo kopírují fungování v přírodě. Teorie genetického algoritmu je založena na Darwinově teorii *"O původu druhů"*

Rozlišujeme jedince, jejichž genetickou informací jsou všechny hodnoty spojů v neuronové síti. Při učení procházíme stálým cyklem, při kterém se tato genetická informace upravuje tak, aby se výsledky zlepšovaly.

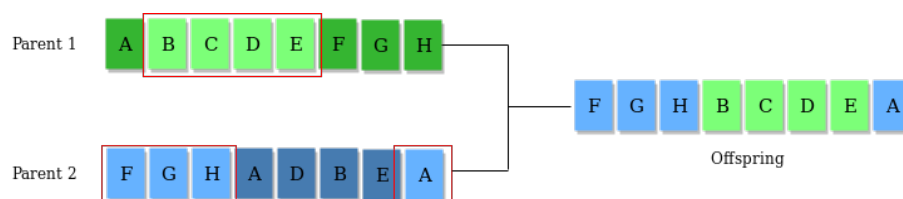


Schéma 5: Genetický crossover

Cyklus genetického algoritmu

1. Určíme, jak velký počet jedinců chceme testovat (**velikost populace**)
2. Naplníme populaci náhodně generovanými jedinci
 - každý jedinec je abstrakcí jedné náhodné neuronové sítě
 - všechna spojení sítě tvoří genetickou informaci
3. Celou populaci vyzkoušíme v testovacím prostředí a zapíšeme jejich **zdatnost** v zadaném problému
4. Z populace pseudo-náhodně vybereme skupinu rodičů nové generace
 - ti s **vyšší zdatností mají větší šanci** být zvoleni mezi rodiče
 - každý z jedinců může být zvolen opakovaně
5. Ze skupiny rodičů náhodně vybereme vždy dva jedince a náhodným poskládáním jejich genetických informací vytvoříme dva nové potomky (**genetický crossover** Schéma 5)
6. Každý gen každého potomka má určitou šanci náhodně zmutovat
 - hodnotu této náhody musíme určit
 - **pravděpodobnost mutace** se obvykle pohybuje mezi 1% - 5%
Příliš malá hodnota a populace není schopna prozkoumávat nová řešení.
Příliš velká hodnota a mutace může rozbít správná řešení.
7. Do splnění požadavků opakujeme kroky 3 - 6, při kterých by se měli objevovat stále lepší a lepší jedinci.

Základní myšlenkou při křížení genetických informací je předpoklad, že ze dvou úspěšných jedinců vznikne vždy potomek buď **zdatnější, nebo stejně dobrý**. V praxi toto není vždy pravda, už kvůli přítomnosti náhody ve většině krocích cyklu. Teorie ale funguje, a tak jsme tímto způsobem schopni dojít správných výsledků.

V projektu je genetický algoritmus založen s populací o **50 členech**. Každý člen představuje náhodnou neuronovou síť (stejně architektury [6, 8, 4, 2]). Kvůli zdánlivé obtížnosti úkolu a velikosti sítě jsem se rozhodl pro hodnotu **pravděpodobnosti mutace = 2,5%**. Zároveň pro snazší uchování nejlepší genetické informace se nejlepší jedinec poslední generace přenáší do další.