

NEAT - CARS

František Karel

ČVUT–FIT

karesfra@fit.cvut.cz

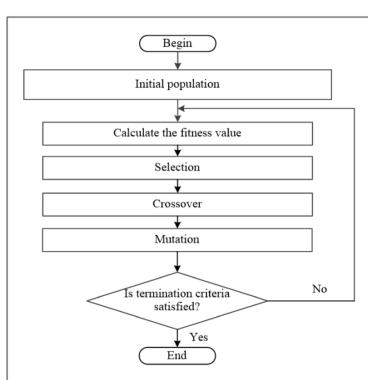
29. května 2025

1 Úvod

NEAT je genetický algoritmus, který se snaží optimalizovat populaci neuronových sítí nad danou úlohou. Mým cílem bylo pomocí NEAT algoritmu naučit řídit samostatně auto ve 2D prostředí, na různě obtížných tratích za co nejnížší čas. Dalším mým cílem bylo vizualizovat celkový běh algoritmu a jeho chování.

2 Genetický algoritmus

Genetický algoritmus je optimalizační metoda inspirováná evolucí. Často je využíván u úloh, kde není možné najít optimální řešení v polynomiálním čase, nebo neexistuje vůbec. Základní princip algoritmu je postaven na evoluci a metodě "pokus omyl". Postup algoritmu, jak je ilustrováno na obrázku je následující: Inicializuje se **populace** (o fixní velikosti), provede se výpočet **fitness funkce** (ohodnocení výkonnosti pro každý genom z populace), **selekcce** (například 5 genomů s nejvyšší fitness funkcí), **crossover**, neboli křížení a na závěr **mutace** (malá pravděpodobnost změny v genomu, která přispívá svojí náhodností a může se dostat z lokálního minima), pokud jsme nenašli optimální řešení, celý algoritmus pouštíme znova s novou "lepší" populací, nebo zastavujeme v případě jiné zastavovací podmínky (například maximální počet generací)



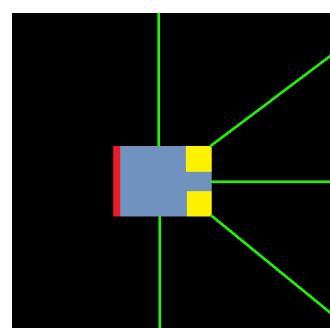
Obrázek 1: Postup genetického algoritmu

3 NEAT

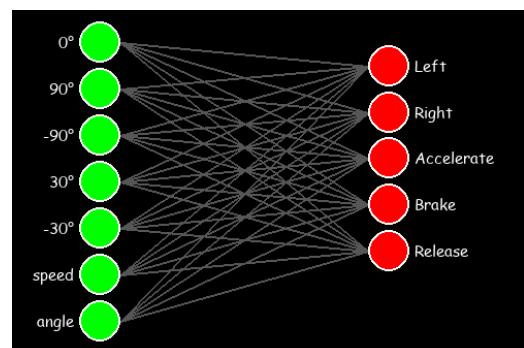
NeuroEvolution of Augmenting Topologies (NEAT), je genetický algoritmus, ve kterém jednotlivé genomy reprezentují neuronové sítě. Postupnou iterací se generují nové a výkonnější sítě. Nechci zde zabíhat do detailu implementace, vše je popsáno v tomto krátkém originálním paperu

4 Parametry

Jako vstupní vrstvu jsem zvolil tyto příznaky: 5 senzorů vyobrazených na obrázku, rychlosť a úhel kterým je auto zrovna otočeno. Ve výstupní vrstvě jsem nastavil 5 neuronů, s aktivační funkcí **tanh** a v jedné iteraci aktivují pouze jednu akci, která je maximem z těchto pěti neuronů. V procesu mutace se může v neuronové síti vyskytnout i hidden layer.



Obrázek 2: senzory



Obrázek 3: Feedforward neuronová síť

5 Procedurální generování tratí

Implementoval jsem velice primitivní, ale fungující způsob na generování tratí. Pomocí rovnoměrného rozdělení jsem vygeneroval několik desítek bodů, následně s využitím matematického aparátu Convex hull, který vybere ze všech bodů ty body, které pokud by byly spojené, tak by zahrnovaly všechny ostatní body. Tímto získané body použiju na vytvoření několika Beziérových křivek a trať je hotová.

6 Závěr

V této práci jsem implementoval algoritmus NEAT pro úlohu řízení auta ve 2D prostředí. Cílem bylo vyvinout agentního řidiče, který se dokáže co nejrychleji dostat do cíle na různě obtížných tratích. Podařilo se mi vytvořit funkční systém, ve kterém se populace neuronových sítí postupně zlepšovala díky evolučním principům jako je selekce, křížení a mutace.

Při experimentech jsem pozoroval, že NEAT dokáže najít rozumné strategie, i když se často dostane do lokálních minim, což je důsledek vysoké dimenze prostoru možných topologií neuronových sítí. Také se ukázalo, že vizualizace průběhu trénování (například aktuální pozice aut nebo struktura neuronové sítě) výrazně pomáhá při ladění a analýze výsledků.

Do budoucna bych rád algoritmus porovnal s jinými přístupy, zejména s metodami hlubokého učení jako je Deep Q-Network (DQN). Zajímalo by mě především srovnání efektivity tréninku, schopnosti přenosu naučeného chování na nové tratě a robustnosti při neznámých vstupních podmínkách. Dále bych chtěl přidat možnost učení více agentů současně, nebo zapojení penalizace za kolize, což by mohlo vést k bezpečnějším jízdním strategiím.

Celkově NEAT ukázal, že je schopný řešit i komplexnější úlohy a jeho výhodou je schopnost evolovat nejen váhy, ale i samotnou strukturu neuronové sítě.

Reference