

Pacman Labboek

Evelyne van Oers, Ruben van Heusden, Daniël Groen, Daniëlle Bijl

June 2016

1 Introductie

In dit project zal de computer leren hoe het pacman efficient kan spelen, met gebruik van machine learning (((voeg machine learning theorie kort toe))). Onze hypothese is dat met behulp van Q-learning, de computer kan leren om pacman efficient te spelen. De gebruikte software is python 3, in een 32 bit linux ubuntu virtual machine.

Het project is opgedeeld in een aantal stappen:

1. het maken van een kleine pacman omgeving
2. het implementeren van q-learning
3. het maken van een volledig pacman bord waar pacman kan leren

In eerste instantie moet een kleine omgeving gebouwd worden die de regels van pacman hanteert. Dit houdt in: pacman en spookjes kunnen niet door muren lopen, aangeraakt worden door een spookje is nadelig, en het pakken van voedsel is voordelig.

De tweede stap is het implementeren van q-learning in de kleine omgeving. De hypothese is dat Q-learning goed werkt voor een pad-probleem zoals pacman, maar dat het lastiger zal worden naarmate het bord groter wordt. Het idee is dus om het bord stap voor stap uit te breiden.

De laatste stap is het maken van een volledig pacman bord met alle tegenstanders, waar de Q-learning op toegepast wordt. De hypothese is dat wanneer alles werkt op kleine schaal, dit met een aantal aanpassingen ook op grote schaal zal werken.

2 Voortgangsrapportage

Maandag 20 Juni

Op maandag hebben wij in python de benodigde code voor pacman gemaakt, en een klein 2 bij 4 bord voor het uitvoeren van tests. Classes voor alle onderdelen in pacman zijn gemaakt met definities van wat deze onderdelen kunnen doen:

- Pacman
- Ghosts
- Dots
- Achtergrond zoals paden en blokjes

De ghosts bewegen richting pacman door de afstand tussen pacman en henzelf te schatten en deze te minimalizeren. We wilden de beweging van de ghosts gericht maken, maar ook simpel. Hierom hebben we ervoor gekozen dat de ghosts hun afstand tot pacman berekenen met behulp van pythagoras en deze afstand willen verkleinen. Dit werkt in vrijwel alle gevallen naar behoren.

We hebben ons ook verdiept in Q-learning, en hebben dit toegepast op het kleine 2 bij 4 bord met het gebruik van een tabel met Q-waardes.

Woensdag 22 Juni

Vandaag hebben we ons verder verdiept in Q-learning. We zijn van mening dat Q-learning met gebruik van een tabel niet praktisch zal zijn om pacman te spelen op een redelijk speelveld. Een redelijk speelveld is een speelveld met de volgende specificaties:

- Een veld van $3 * 3 = 9$ vakjes
- Één ghost als tegenstander
- Minimaal één dot dat pacman op moet pakken om te winnen.

We hebben het hier al over meer dan 9^2 states waar een tabel voor gemaakt moet worden met waarden. Dit achtten wij niet praktisch. In plaats van het opstellen van een grote tabel gaat onze interesse uit naar feature-based Q-learning, waarbij niet de states, maar specifieke features tijdens het leren in acht genomen worden. Een feature is een bepaald kenmerk in states van een wereld die relevant kunnen zijn voor de agent om zijn doel te bereiken. Feature-based Q-learning

updated de weging van een bepaalde feature om de Q-waarde te berekenen zodat deze zo dicht mogelijk bij de echte reward waarde komt te liggen. Het voordeel hiervan is dat pacman in verschillende contexten een 'goede' state van een 'slechte' state kan onderscheiden door kennis te gebruiken over gelijksoortige states.

De gekozen features zijn:

- De afstand tot de dichtst bijzijnde dot
- De afstand tot de ghost.

Er is gekozen voor deze features om het probleem simpel te houden. Ook zijn wij van mening dat voor de kleine schaal van het bord dat gebruikt wordt, meerdere features niet nodig zijn.