YILDIZ TEKNİK ÜNİVERSİTESİ BİLGİSAYAR MÜHENDİSLİĞİ

YAPAY ZEKA DERSİ DÖNEM PROJESİ

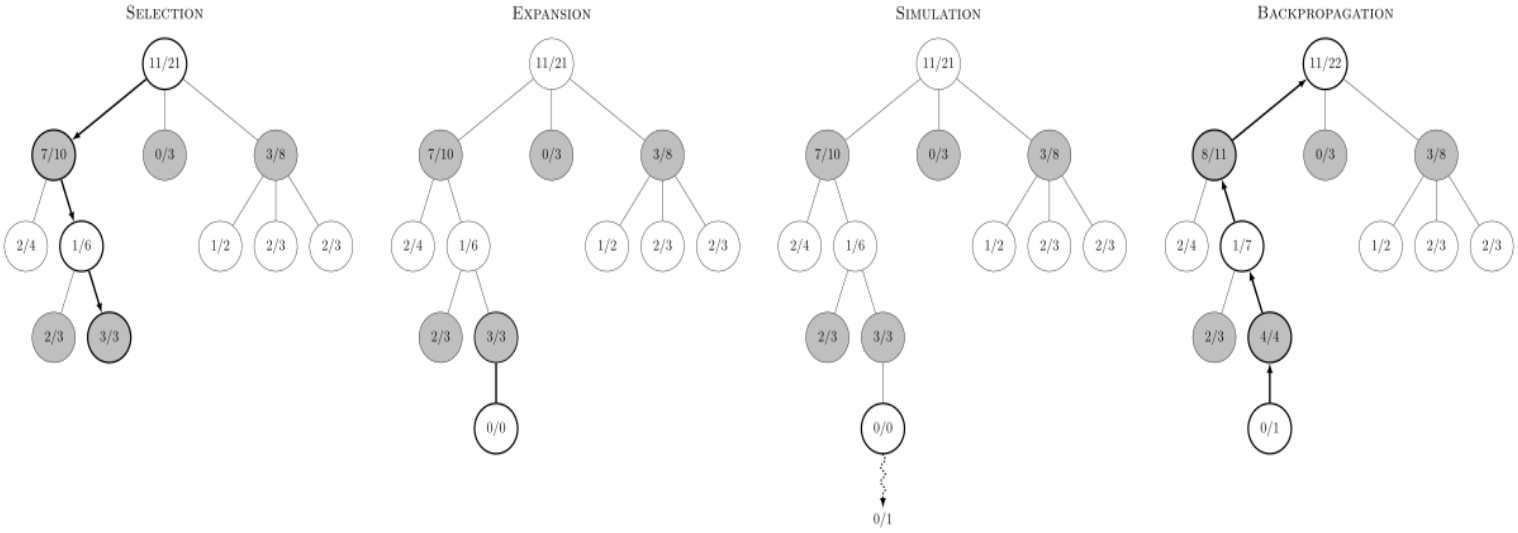
MONTE CARLO ARAMA AĞACI İLE TİC TAC TOE

18011613 – Mehmet Emre Gül

Mayıs, 2020

Monte Carlo Tree Search (Monte Carlo Ağacı Arama Algoritması)

Monte Carlo arama algoritması 2006 yılında Remi Coulom tarafından tanıtılmıştır. Algoritma, mevcut bir durumda yapılacak hamlenin, yapılabilecek en iyi hamle olması esasına dayanır. Bunun için mevcut durumdan rastgele hamlelerle sonuca ulaşır. Sonuç kaydedilir. Bu işlem epey bir miktar tekrar ettiğinde hangi hamlenin en iyi sonuca yaklaştırdığı belli olur.



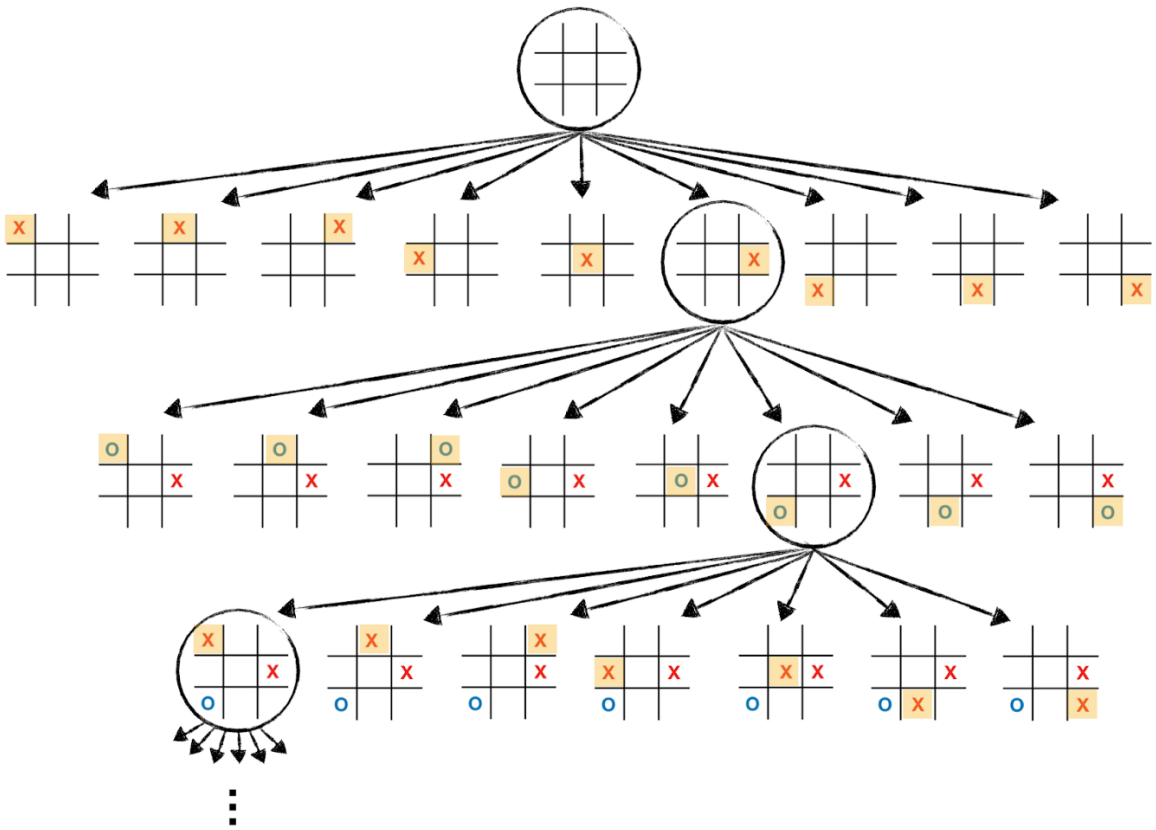
Monte Carlo algoritması 4 parçadan oluşur. Bunlar selection, expansion, simulation ve backpropagation kısımlarıdır. Mevcut düğümden gidilebilecek bütün düğümler (yani o durumdan yapılabilecek bütün hamleler) ziyaret edilmişse o düğüm fully expanded düğüm olarak adlandırılır. Selection kısmında, mevcut düğüm fully expanded ise aşağıdaki formülde en yüksek değer veren düğüm seçilir. Seçilen düğüm de fully expanded ise aynı şey tekrar edilir. Selection kısmı fully expanded olmayan bir düğüm bulana kadar devam eder.

UCT = (point / visits) + sqrt(2 * log(parent.visits) / visits)

Bu formül selection kısmında daha iyi hamlelerin seçilmesini sağlayarak algoritmanın daha optimize çalışmasını sağlar.

Fully expanded olmayan bir düğüm bulunduğunda expansion kısmına geçilir. Burada yapılabilecek hamle için yeni bir düğüm oluşturulur. Daha sonra simulation kısmına geçilir. Simulation kısmında yeni üretilen düğümden oyun sonuçlanana kadar rastgele hamleler yapılır. Oyun sonuçlandığında, sonuç yeni üretilen düğümden en üstteki root a kadar yazılır. Bu kısıma da backpropagation denir.

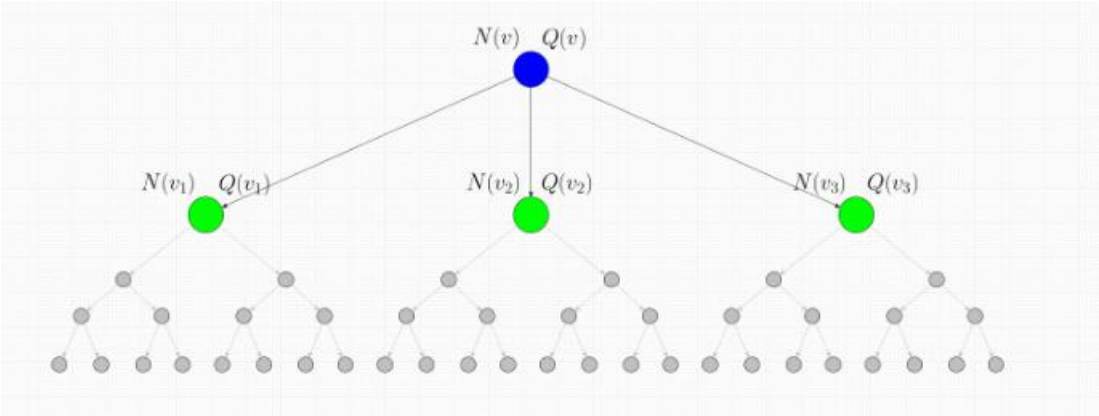
Bu işlemler defalarca kez tekrarlanır. En sonda root tan sonraki düğümlerden point / visits oranı en yüksek olan düğüme karşılık gelen hamle oynanır.



Örnek resim

Geliştirme Süreci

Projede iki farklı algoritma geliştirilmiştir. İlki yukarıda anlatıldığı gibidir. İkincisi sadece mevcut durumdan gidilebilecek durumlar için düğümler üretip simulation adımına geçilmesidir. Bu ikisini sırasıyla A ve B diye isimlendirdim.



B algoritmasında, mavi düğüm mevcut durumumuz, yeşiller ondan gidilebilecek düğümler olsun. Burada selection ve expansion kısımları bu kadardır. Daha aşağı inilmez. Onun yerine bu üç düğümden birine geçildikten sonra direkt simulation ve backpropagation kısımları gerçekleştirilir.

Algoritma Sonuçları

5x5'lik tahtada 4 tane yan yana veya çapraz aynı işaretten koyulması şeklinde oynanmıştır.

A için *(Kullanıcı galibiyeti, mağlubiyeti)*

Galibiyet	4
Beraberlik	4
Mağlubiyet	7

B için

Galibiyet	4
Beraberlik	3
Mağlubiyet	8

3x3 lik tahtada 3 tane yan yana veya çapraz aynı işaretten koyulması şeklinde oynanmıştır

A için

Galibiyet	3
Beraberlik	7
Mağlubiyet	5

B için

Galibiyet	0
Beraberlik	9
Mağlubiyet	6

Kaynaklar ve Faydalanılan Linkler

https://www.caktusgroup.com/blog/2015/09/24/introduction-monte-carlo-tree-search-1/

https://int8.io/monte-carlo-tree-search-beginners-guide/

https://sites.google.com/view/mfatihamasyali/yapay-zeka?authuser=0

https://www.geeksforgeeks.org/python-implementation-automatic-tic-tac-toe-game-using-random-number/

https://dev.to/jamesshah/the-classic-tictactoe-game-in-python-cpi

https://towardsdatascience.com/game-ais-with-minimax-and-monte-carlo-tree-search-af2a177361b0

https://stackoverflow.com/questions/49456415/monte-carlo-tree-search-tic-tac-toe-poor-agent

https://en.wikipedia.org/wiki/Monte_Carlo_tree_search

https://towardsdatascience.com/pathwayz-ai-3c338fb7b33