

دانشگاه صنعتي امیرکبیر  
(پلی­تکنیک تهران)

دانشكده ریاضی و علوم کامپیوتر

درس هوش مصنوعی و کارگاه

گزارش پروژه سوم

نگارش

نرگس گل محمدی

40113426

استاد اول

دکتر مهدی قطعي

استاد دوم

بهنام یوسفي مهر

ابان 1403

# چكيده

در این پروژه به بررسی الگوریتم مینیماکس پرداختیم و برای بهتر کردن آن از هرس الفا بتا استفاده کردیم تا بهتر و بهینه تر عمل کند در ادامه به بررسی الگوریتم اکسپکتیماکس میپردازیم.

در این پروژه هدف فهم الگوریتم ها و پیاده سازی این الگوریتم ها در بازی اتللو است.

اتللو شامل دو نوع مهره سفید و مشکی است که با کمک این الگوریتم ها بهترین حرکت برای هر کدام پیش بینی میشود.

واژه کلیدی: مینیماکس ، هرس الفا بتا ، اکسپکتیماکس ، اتللو

|  |  |
| --- | --- |
| فهرست مطالب | صفحه |

[چكيده ‌أ](#_Toc179490925)

[۱. فصل اول مقدمه 4](#_Toc179490926)

[۲. فصل دوم مینیماکس 5](#_Toc179490927)

[۱-۲- نحوه عملکرد 6](#_Toc179490928)

[۳. فصل سوم تابع ارزیابی 7](#_Toc179490930)

[۱-۳- نحوه عملکرد 8](#_Toc179490931)

[۴. فصل چهارم هرس الفا بتا 10](#_Toc179490934)

[۱-۴- نحوه عملکرد 11](#_Toc179490935)

[۵. فصل پنجم اکپکتیماس 13](#_Toc179490938)

[۱-5- نحوه عملکرد 14](#_Toc179490939)

[6. فصل کد 16](#_Toc179490942)

[۱-6- تحلیل 17](#_Toc179490943)

[7. فصل هفتم جمع بندی 19](#_Toc179490946)

# ۱. فصل اول مقدمه

**مقدمه**

اتللو یک بازی دو نفره روی یک تخته ۸ در ۸ است. هر بازیکن دیسک های دایره رنگی دارد که یک طرف آن سفید و طرف دیگر سیاه است. یکی از بازیکنان با دیسک های سیاه و دیگری با دیسک های سفید بازی می کند.تخته بازی در ابتدا ۴ دیسک در وسط خود دارد: دو دیسک سفید و دو دیسک سیاه که به شکل متقاطع قرار گرفته اند. با قرار دادن مهره یک رنگ دو طرف مهره بازیکن دیگر رنگ ان به رنگ بازیکن فعلی درمیاید .

بازی زمانی تمام می شود که یا تمام خانه های تخته پرشده باشد یا هیچ بازیکنی نتواند حرکت دیگری انجام دهد.

# ۲. فصل دوم مینیماکس

## ۱-۲- نحوه عملکرد

الگوریتم مینیماکس یک الگوریتم هوش مصنوعی برای بازی های دونفره ایکس او ، شطرنج و غیره است .

هدف این الگوریتم انتخاب بهترین حرکت برای بازیکنی است که می‌خواهد بیشترین امتیاز ممکن را کسب کند، در حالی که فرض می‌کند حریفش هم می‌خواهد امتیاز بازیکن را به حداقل برساند.

اول یک درخت بازی ساخته می‌شود که شامل تمام حرکات ممکن از وضعیت فعلی بازی است. هر گره در این درخت نشان‌دهنده یک وضعیت خاص از بازی است.

الگوریتم مینیماکس از پایین درخت به سمت بالا حرکت میکند و به هر گره، بهترین امتیاز ممکن را از دید بازیکنی که نوبت اوست می‌دهد. اگر نوبت بازیکن ماکزیمم باشد، بیشترین امتیاز انتخاب می‌شود و اگر نوبت مینیمم باشد، کمترین امتیاز انتخاب می‌شود.

این الگوریتم به صورت بازگشتی پیاده‌سازی می‌شود و هر بار به سراغ حرکت‌های ممکن می‌رود و امتیاز حرکت‌ها را محاسبه می‌کند تا بهترین حرکت را بیابد.



# ۳. فصل سوم تابع ارزیابی

## ۱-۳- نحوه عملکرد

در الگوریتم مینیماکس، تابع ارزیابی (Evaluation Function) برای تخمین و ارزیابی وضعیت یک گره یا حالت بازی استفاده می‌شود. این تابع به الگوریتم کمک می‌کند که بدون نیاز به بررسی تمامی حرکات تا انتهای بازی، وضعیت فعلی را به شکلی تخمین بزند که به بهترین حرکت ممکن دست یابد. تابع ارزیابی معمولاً در بازی‌هایی که فضای جستجوی بزرگی دارند (مانند شطرنج) کاربرد بسیاری دارد.

تابع ارزیابی یکی از بخش‌های حیاتی برای تعیین کیفیت تصمیم‌گیری در الگوریتم مینیماکس است.

درر بازی های ساده مثل ایکس و او میتوانیم بگوییم وقتی ایکس برد مثبت ده را برگرداند وقتی او برد منفی ده را و اگر مساوی شد صفر برمیگردد.

ولی در بازی های پیچیده تر مانند شطرنج جزئیات بیشتری داریم برای مثال :

تعداد مهره ها ، موقعیت هر مهره ، شرایط شاه و غیره

در بازی های پیچیده تر برای جلوگیری از محاسبات خیلی زیاد عمق محدودی را در نظر میگیریم و تا آنجا پیش میریم چون تعداد حالت ها خیلی زیاد میشود و غیر واقعی و غیر ممکن میشود .

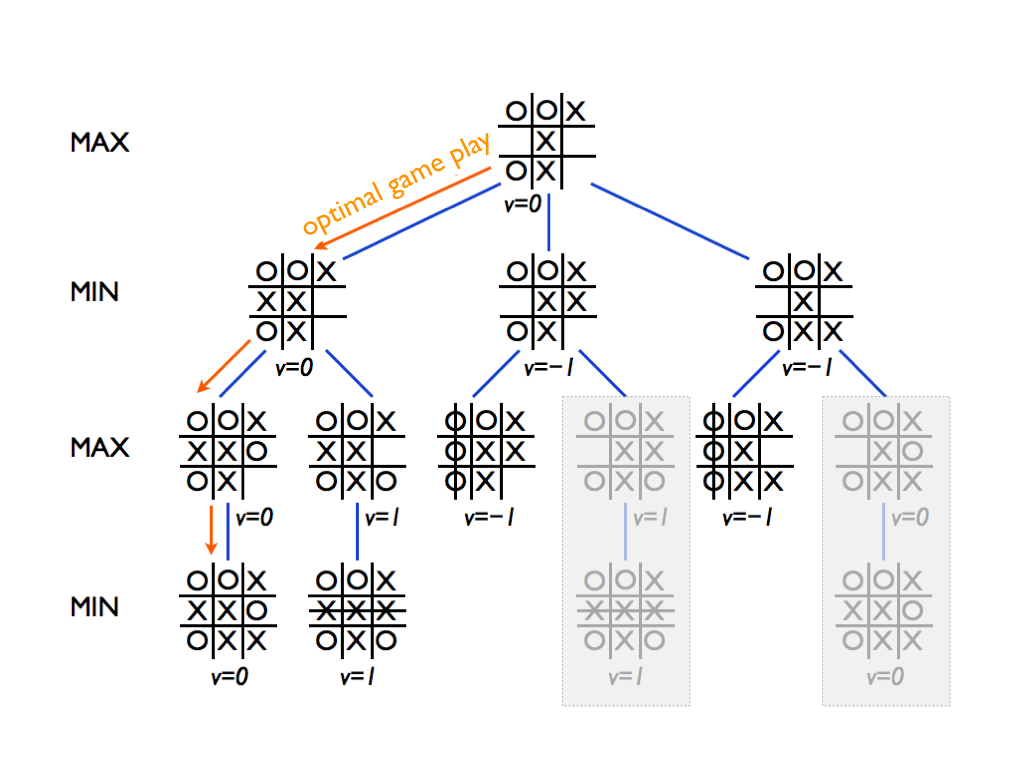
با استفاده از تابع ارزیابی، الگوریتم مینیماکس می‌تواند حرکات را سریع‌تر و با دقت بیشتری بررسی کند، به خصوص اگر از بهبودهایی مانند هرس آلفا-بتا هم استفاده کند که گره‌های غیرضروری را حذف می‌کند.

# ۴. فصل چهارم هرس آلفا بتا

## ۱-۴- نحوه عملکرد

هرس آلفا-بتا (Alpha-Beta Pruning) یک بهبود برای الگوریتم مینیماکس است که به سرعت و کارایی الگوریتم را افزایش و تعداد گره‌هایی که باید بررسی شوند را کاهش میدهد .این کار به الگوریتم اجازه می‌دهد تا بتواند بدون بررسی همه حالات ممکن در درخت جستجو، بهترین حرکت را پیدا کند و از بررسی گره‌های غیرضروری صرف‌نظر کند.

در هر مرحله از الگوریتم مینیماکس الفا نشان دهنده ماکسیمم حالتی که میتواند داشته باشد و بتا مینیمم ترین آن است. این جستوجو ادامه پیدا میکند تا مقدار آلفا بیشتر یا مساوی بتا شود. در نهایت میبینیم با این کار از گره های زیادی صرف نظر کردیم و باعث سرعت و دقت بیشتر ما شده.



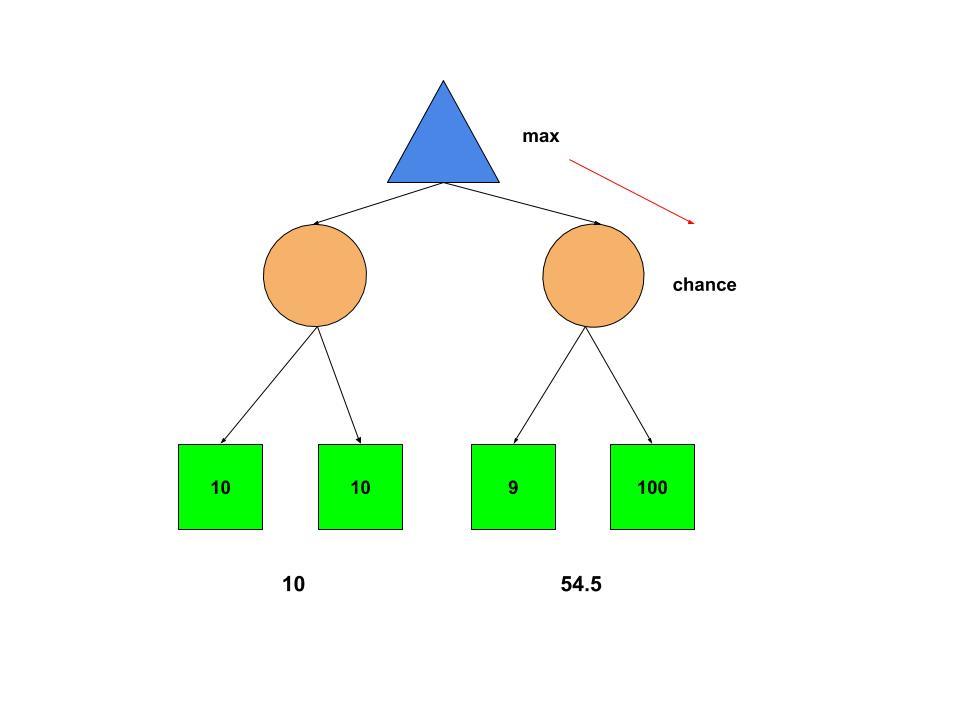
# ۵. فصل پنجم اکسپکتیماکس

## ۱-5- نحوه عملکرد

اکسپکتیماکس (Expectimax) یک الگوریتم بهینه‌سازی است که برای بازی‌ها و مسائلی استفاده می‌شود که شامل عدم قطعیت یا تصادف هستند، مثل بازی‌هایی که عنصر شانس در آنها وجود دارد (بازی‌های تخته‌ای مانند تخته‌نرد که تاس می‌ریزند). در این الگوریتم، علاوه بر گره‌های Max و Min یک نوع جدید گره به نام گره انتظاری (Expectation Node) وجود دارد که نشان‌دهنده حرکات احتمالی است.

در بازی هایی که عناصر تصادفی دارن مینیماکس موثر نیست چون که با حداقل وحداکثر کردن جواب میگیره و جواب قطعی میخواهد برای همین از expectimax استفاده میکنیم در اینجا بجز مین و ماکس گره انتظاری هم داریم که برای بدست اوردن مقدار آن میانگین نود های فرزند رو بدست میاریم و به صورت بازگشتی در گره بالایی میزاریم.

در الگوریتم مینیماکس، فرض بر این است که دو بازیکن دارند به بهترین شکل ممکن بازی می‌کنند؛ اما در اکسپکتیماکس، عنصر شانس اضافه می‌شود که نمی‌تواند تحت کنترل بازیکنان باشد.



# 6. فصل ششم کد

## ۱-6- تحلیل

قسمت اول الگوریتم مینیماکس را تعریف میکند :

یک بازیکن ماکسیمم میشود و دیگری مینیمم

ابتدا در این بخش چک میکنیم که برای بازیکن فعلی آیا میتونایم بیشتر در درخت تصمیم حرکت کنیم یا عمق داریم یا نه

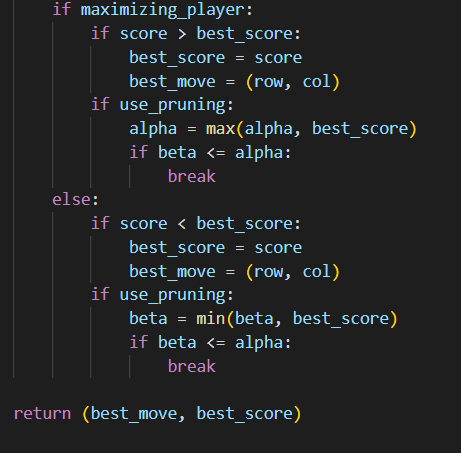
بعد چک میکنیم ک آیا تخته ما پر است یا نه

در نهایت چک میکنیم میتوانیم حرکتی ولید انجام بدهیم یا نه

سپس رنگ بازیکن را مشخص میکنیم / ابتدا بست موو را تعریف میکنیم و مقدار نان به آن میدهیم

سپس برای هر کدام که ماکس یا مین شده بست اسکور را تعریف و مقدار اولیه به آن میدهیم

حالا از تخته کپی میگیریم تا تغییرات را انجام بدیم و مهره بازیکن فعلی را قرار میدهیم تا نتیجه را شبیه سازی کنیم. / حالا مینیماکس را صدا میزنیم عمق یدونه کم میشه چون یه مرحله رفتیم جلو / بازکن فعلی تغییر میکنه و امتیاز بررسی میشه

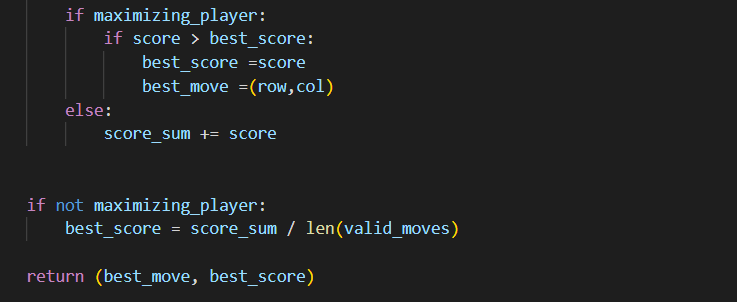


اینجا اگر بازکن فعلی ماکس بود اگر امتیاز بهتری پیدا کردیم آن را به عنوان بهترین حرکت تا اینجا معرفی میکنیم و اگر از هرس استفاده میکردیم تا زمانی که بتا بزرگ تر بود ادامه میدهیم برای بازیکن مینیمم هم همین کار را میکنیم .

در اینجا تابع ارزیابی امتیاز تخته رو برای هر بازیکن برمیگرداند و دو فاکتور اختلاف بین امتیاز ها و امتیاز های گوشه ای را داریم. ابتدا امتیاز های هر دورا میگیرد و به صورت لیست برمیگرداند حالا اختلاف انها را ذخیره میکند / حالا بعد از معرفی نقاط گوشه میگوییم اگر بازیکن در ان نقطه ها باشد تعداد یکی زیاد و اگر حریف بود و خالی نبود یکی کم و بعد با ضریب امتیاز را برمیگردانیم



تابع بعدی اکسپکتیماکسه که همون طور که توضیح دادیم همان مینیماکسه با یک نوع گره جدید که این گره با میانگین گره های پایینی بدست میاید.

همان طور که گفتیم الگوریتم مشابه مینیماکس است و کد هم همان توضیحات بالا را دارد

تنها تفاوت هایش در معرفی تابع است چون هرس ندارد و در نهایت اسکور سام به ما جمع اسکور را میدهد و تا بست اسکورمان را مساوی میانگین قرار دهیم.

# 7. فصل هفتم جمع بندی

با توجه به تمام نکات گفته شده نتیجه میگیریم که الگوریتم مینیماکس کاربرد زیادی دارد و میتواند در پیش بینی و انجام بهترین حرکت در هوش مصنوعی کمک ما کند. همچنین برای بهینه سازی این الگوریتم و حذف گره های غیر ضروری و شاخه های غیر مفید از هرس الفا بتا استفاده میکنیم که باعث میشود به بررسی نود های غیر مرتبط نپردازیم و سرعت را بالا ببریم.

در الگوریتم اکسپکتیماکس ما مینیماکس را گسترش میدیم و برای بازی های تصادفی از آن استفاده میکنیم که فرقش با مینیماکس در گره های شانس است که میانگین وزن دار گره های پایینی خود را برمیگرداند.

منابع

<https://courses.cs.washington.edu/courses/cse573/04au/Project/mini1/RUSSIA/miniproject1_vaishu_muthu/Paper/Final_Paper.pdf>

<https://www.geeksforgeeks.org/expectimax-algorithm-in-game-theory/>

<https://www.youtube.com/watch?v=zp3VMe0Jpf8>

<https://www.youtube.com/watch?v=SLgZhpDsrfc>

<https://www.youtube.com/watch?v=Q7Xz4ZpCgUY&t=6s>

<https://www.youtube.com/watch?v=xBXHtz4Gbdo&t=249s>

<https://www.youtube.com/watch?v=fY-9Kcf9ycI>