

طراحی و پیاده‌سازی مدل هوشمند بازی دوز با استفاده از شبکه عصبی کانولوشنی

چکیده

در این مقاله ما هدف داریم در مورد یک مدل هوشمند برای بازی "دوز" با استفاده از شبکه های عصبی درست کنیم. این مدل سطح بندی و به اصطلاح ورژن های مختلفی دارد که در این سطح ما برای ساخت یک مدل ساده (هدف آموزشی) سنگ بنا را قرار دادیم. اهمیت این موضوع یعنی مدل های هوشمند برای بازی ها و سرگرمی های مختلف جایی مشخص شود که بتوانند همانند انسان از خلاقیت انسانی تقلید کنند. مدل بر روی داده های جمع آوری شده تمرین داده شدند (خود برنامه بازی دوز با زبان سوپت نوشته شده و در این [لینک](#) قابل مشاهده است) و برای مراحل بعدی در خود همین برنامه هم تغییراتی را شاهد خواهیم بود.

مقدمه

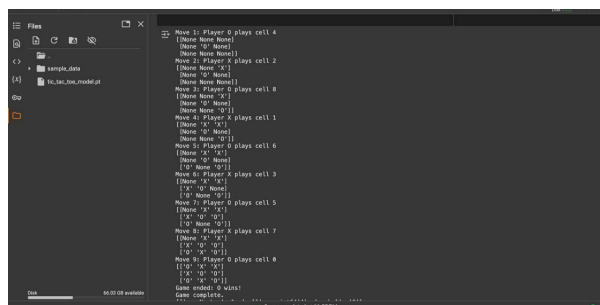
در سالهای اخیر با پیشرفت عظیم هوش مصنوعی و رشد مدل های مبتنی بر ساختار شبکه های عصبی و اهمیت بالای آن و همچنین قابل درک و شهودی بودن آن به دلیل ارتباط مستقیم با شبکه های عصبی و نورون های مغزی انسان، می توان گفت موضوع بسیار خوب برای شرح این بخش از یادگیری ماشین یا یادگیری عمیق است و میتواند با ترکیب شدن با یک موضوع ساده، یک مسیر یادگیری و تحقیق همواری را برای خواننده و محقق فراهم کند و ما همراه کردن این بخش از هوش مصنوعی یعنی همان شبکه های عصبی و بازی "دوز" سعی بر این داریم در طول طراحی و پیاده سازی این پروژه برای شما خواننده عزیز مفاهیم مورد نیاز رو شرح دهیم و در نهایت بتوانیم یک مدل هوشمند و با درصد خطای کم درست کنیم. قرار است در این مدل "تصمیم" بگیریم کدام خانه و با چه رویکردی باید پر شود.

روش تحقیق

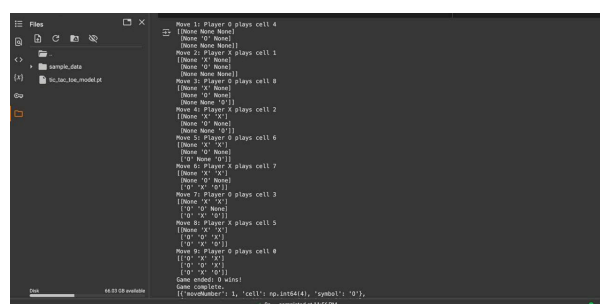
برای طراحی و پیاده سازی این مدل با استفاده از شبکه های عصبی از زبان برنامه نویسی پایتون استفاده شده است و سعی بر این بوده است با کمک از فریم ورک Torch بتوانیم یک شبکه عصبی کانولوشنی درست طراحی کنیم. برای جمع آوری داده با کمک از زبان Swift یک برنامه کاملاً شخصی برای این بازی (دوز) نوشتیم و برای جمع آوری آن از دوستان دانشگاه کمک گرفتیم و در طول کلاس با آنها برای جمع آوری داده بازی و به رقابت مشغول بودیم! بازی دور در یک محیط 3*3 طراحی شده است مانند تصویر برای مدل اولیه، داده ها به دو دسته تمرین و تست تقسیم شده اند.

نتایج

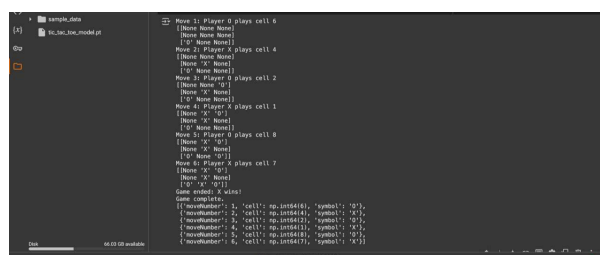
همانطور که در تصاویر مشخص است نتیجه در سه دور مسابقه تصویر برداری شده و این مدل نشان میدهد میتواند با دقتی حدود 60 درصد از هوشمندی خودش استفاده کند (مدل گاهی در حرکت بعدی خود اشتباهاتی دارد که در ورژن ابتدایی آن کاملاً قابل پیشبینی است) و دلیل خطای بالای آن از داده های ابتدایی آن است که نشان میدهد مدل روی برخی حرکات overfit شده است و نسبت به برخی دیگر از شرایط واکنشی ندارد لازم به ذکر است که مدل برای 50 بازی دقت 50 درصد را کسب کرده و روش تست آن هم بر اساس آن بوده است که آیا خانه مورد انتظار همان خانه‌ای است که مدل پر کرده یا خیر. (تمامی مراحل و کدهای آن در این لینک قابل مشاهده است)



```
Move 1: Player O plays cell 4
[Move Name Move]
[Move O Move]
[Move Name Move]
Move 2: Player X plays cell 2
[Move Name X]
[Move O Move]
[Move Name Move]
Move 3: Player O plays cell 8
[Move Name O]
[Move X Move]
[Move Name Move]
Move 4: Player X plays cell 1
[Move X X]
[Move O Move]
[Move Name Move]
Move 5: Player O plays cell 6
[Move X X]
[Move O Move]
[Move Name Move]
Move 6: Player X plays cell 3
[Move X X]
[Move O Move]
[Move Name Move]
Move 7: Player O plays cell 5
[Move X X]
[Move O Move]
[Move Name Move]
Move 8: Player X plays cell 7
[Move X X]
[Move O Move]
[Move Name Move]
Move 9: Player O plays cell 9
[Move X X]
[Move O Move]
[Move Name Move]
Game ended: O wins!
Game complete.
```



```
Move 11: Player O plays cell 4
[Move Name Move]
[Move O Move]
[Move Name Move]
Move 12: Player X plays cell 1
[Move Name X]
[Move O Move]
[Move Name Move]
Move 13: Player O plays cell 8
[Move Name O]
[Move X Move]
[Move Name Move]
Move 14: Player X plays cell 2
[Move X X]
[Move O Move]
[Move Name Move]
Move 15: Player O plays cell 6
[Move X X]
[Move O Move]
[Move Name Move]
Move 16: Player X plays cell 7
[Move X X]
[Move O Move]
[Move Name Move]
Move 17: Player O plays cell 3
[Move X X]
[Move O Move]
[Move Name Move]
Move 18: Player X plays cell 5
[Move X X]
[Move O Move]
[Move Name Move]
Move 19: Player O plays cell 9
[Move X X]
[Move O Move]
[Move Name Move]
Game ended: O wins!
Game complete.
```



```
Move 21: Player O plays cell 6
[Move Name Move]
[Move O Move]
[Move Name Move]
Move 22: Player X plays cell 4
[Move Name X]
[Move O Move]
[Move Name Move]
Move 23: Player O plays cell 2
[Move Name O]
[Move X Move]
[Move Name Move]
Move 24: Player X plays cell 1
[Move X X]
[Move O Move]
[Move Name Move]
Move 25: Player O plays cell 8
[Move X X]
[Move O Move]
[Move Name Move]
Move 26: Player X plays cell 7
[Move X X]
[Move O Move]
[Move Name Move]
Move 27: Player O plays cell 5
[Move X X]
[Move O Move]
[Move Name Move]
Move 28: Player X plays cell 9
[Move X X]
[Move O Move]
[Move Name Move]
Game ended: X wins!
Game complete.
```

نتیجه‌گیری

در این تحقیق، یک مدل هوشمند بر پایه شبکه عصبی کانولوشنی (CNN) برای بازی "دوز" توسعه داده شد. هدف از طراحی این مدل، ایجاد یک پایه آموزشی برای تصمیم‌گیری خودکار در بازی‌های ساده بود. با استفاده از داده‌های تولیدشده توسط برنامه‌ای که در محیط Swift توسعه یافته، مدل توانست با دقت قابل قبول تصمیمات مناسبی در طول بازی اتخاذ کند. نتایج نشان دادند که مدل قادر است در بیشتر موارد رفتار انسانی را تقلید کند و در موقعیت‌های بحرانی مانند برد یا باخت، تصمیمات منطقی اتخاذ نماید. این مدل می‌تواند در توسعه بازی‌های هوشمند، آموزش مفاهیم اولیه یادگیری ماشین، و طراحی سیستم‌های مبتنی بر تصمیم‌گیری خودکار مورد استفاده قرار گیرد. در ادامه، بهبود ساختار مدل و توسعه نسخه‌های پیشرفته‌تر با قابلیت پیش‌بینی حرکات پیچیده‌تر در نظر گرفته خواهد شد.