



KIRIKKALE ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

BİTİRME PROJE-1 DERSİ

Proje Adı : Magnus Carlsen satranç botu .

Danışman : Dr. Öğr. Üyesi Serkan Savaş

Proje Ekibi :190205056 ilyas demir

Bil. Müh. 4.sınıf N.Ö

Ders adı :Bitirme Projesi 1

Projenin Drive Linki : https://drive.google.com/file/d/1h8_HsMmJ5cxO-uNb5SPry892wI8poQvr/view?usp=share_link

[OCAK-2022]

Magnus Carlsen Satranç Botu

1.Özet:

Bu projede mağşuş Carlsen'in oyun tarzını bilgisayara öğretmek amacıyla oluşturuldu .Çeşitli platformdan toplanan veri setleri alınmıştır . Oyun tarzını öğretmek amacıyla makine öğrenmesi ve derin öğrenme teknikleri kullanıldı .Algoritma tasarımında Maia insansı satranç botu algoritmasından yararlanıldı . Kullanılan yöntemler sonucu 0.9165 doğruluk ve 0.1814 kayıp değeri elde edilmiştir .Elde edilen iyi sonuçlara rağmen proje istenilen düzeye çıkamamıştır .

Anahtar kelimeler:

Satranç hamle tahmini, satranç, derin öğrenme, makine öğrenmesi, Magnus Carlsen.

Abstract:

In this project, it was created to teach Magnus Carlsen's style of play to the computer. Data sets collected from various platforms were taken. Machine learning and deep learning techniques were used to teach the game style. Maia humanoid chess bot algorithm was used in the algorithm design. As a result of the methods used, 0.9165 accuracy and 0.1814 loss values were obtained. Despite the good results, the project could not reach the desired level.

Keywords:

Chess move prediction, chess, deep learning, machine learning, Magnus Carlsen.

2.Giriş :

Öncelikle bu projenin ilham kaynağı olan satranç ustası Magnus Carlsen'i tanıyalım ;

Sven Magnus Øen Carlsen (bilinen adıyla Magnus Carlsen) (30 Kasım 1990; Tønsberg), Norveçli satranç büyükustası.Satranç dâhisi Carlsen 2004 yılında (13,5 yaşındayken) büyükusta unvanını kazanmıştır ve bu onu tarihin en genç üçüncü büyük ustası yapmıştır. 1 Ocak 2010'da (19 yaş 32 gün) dünyanın en genç 1 numarası (FIDE listelerine göre)[1] olmuştur. Ocak 2013 FIDE derece listesine göre, Carlsen 2861 ELO'ya ulaşmış ve tarihin en yüksek derecesini elde etmiştir. Kasım 2013'te Viswanathan Anand'ı yenmiş ve 2013 Dünya Şampiyonluk Maçını kazanmıştır, böylece Dünya Satranç Şampiyonu olmuş ve bununla kalmayıp 2014, 2016, 2018 ve 2021 Dünya Satranç Şampiyonluğu'nu da kazanarak unvanını korumayı başarmıştır. [\[2\]\[3\]\[4\]](#)

Büyük ustayı tanıdığımıza göre artık projemin amacı olan sorumuza geçebiliriz. Bir makineye Magnus Carlsen gibi satranç oynamayı öğretmek mümkün müdür?

Buradaki kritik durum satranç oynamayı öğretmek değil, Magnus Carlsen gibi oynamayı öğretmektir .Yani burada kastedilen yenilmez bir algoritma yaratmak değil gerekli durum ve koşullarda Magnus Carlsen gibi yenilip, kazanmasını sağlamaktır .

peki bu projenin önemine değinecek olursak ;

kadın büyük satranç ustası Raluca Sgîrcea ve uluslararası usta Renier Castellanos'un satranç oyuncularını için yazdığı "10 Things We All Should Learn from Magnus Carlsen ^[5]" adlı yazıları örnek verilebilir .yazıda oyuncuların dikkat etmesi gereken şu 10 konuya değinmişlerdir ; oyun sonu tekniği ,Konumsal satranç ,Taktik oyun ,Esnek düşünme, Geniş açılış seçeneği, Basit satranç oynayın, Mücadele ruhu, Beden eğitimi, Psikolojik güç, Özgüven .

Başka bir faydası da Jorden Novet'in yazmış olduğu "What tech founders can learn from chess champion Magnus Carlsen ^[6]" yazısında anlatılan satranç maçlarında maksimum verimliliği nasıl ve hangi koşullarda aldığına dair durumlardan yola çıkarak şirketlere ve çalışanlara uygulanabilirliğine değinmiştir .

Yukarıdakilerden anlaşılabilceği gibi projenin birden çok yönlü faydası olabilmektedir. Bizim ilk amacımız Magnus Carlsen'nin tekniğini daha iyi anlamaları için yeni veya daha önce Magnus Carlsen ile oynama fırsatı bulamayan insanlar imkan sağlamaktır .Böylelikle kendi yöntem ve tekniklerini geliştirerek büyük ustayı yenebilecek varyasyonların oluşmasını sağlamaktır.

3.Benzer Çalışmalar:

Konu satranç oynayan yapay zeka modelleri olunca kaynak bulmak kolay iken, insan gibi satranç oynayan yapay zeka modellerine gelince kaynaklar sınırlı hale geliyor. Sınırlı kaynak olmasına rağmen bir model var ki tüm bu yükü tek başına rahatlıkla sırtlıyor diyebilirim .tabi bu kendi düşüncem olduğu için eleştirilebilir .Lafı fazla uzatmadan bu modelin adını verip inceleme kısmına geçelim . Maia insansı satranç botu .

MAIA CHESS:

"Maia, kendi kendine oynamak yerine çevrimiçi insan oyunlarından öğrenen AlphaZero/Leela benzeri bir derin öğrenme çerçevesidir. Maia milyonlarca oyun üzerinde eğitilmiştir ve görülen her pozisyonda oynanan insan hareketini tahmin etmeye çalışır. Eğitim sırasında, Maia'ya gerçek bir insan oyununda meydana gelen bir pozisyon verilir ve hangi hamlenin yapıldığını tahmin etmeye çalışır. Yüz milyonlarca pozisyonu gördükten sonra Maia, farklı seviyelerdeki insanların nasıl satranç oynadığını doğru bir şekilde yakalar." ^[7]

Yukarıda yapımcıları tarafından tanımı yapılmıştır. Maia, optimal oyun yerine insan oyununu anlamaya çalışan, insan odaklı bir satranç motorudur. Mevcut satranç motorları "Bu pozisyonda oynamak için en iyi hamle nedir?" diye sorarken, Maia bunun yerine "Bu pozisyonda bir insan ne oynardı?" Maia, bu soruyu belirli bir beceri düzeyindeki insanlar için veya hatta oynayan belirli bir kişi için yanıtlayabilir. Maia, insan kararlarını ayrıntılı bir düzeyde anlayarak, bir oyuncunun oyunlarını öğrenmelerine ve geliştirmelerine yardımcı olma hedefiyle, bir oyuncunun güçlü ve zayıf yönlerini belirlemeye yardımcı olabilir. Benimde amacım Magnus Carlsen'in güçlü ve zayıf yönlerini belirleyecek bir satranç yapay zeka modeli oluşturmak olduğu için Maia algoritmasından faydalandım.

4. Yöntem ve Metot:

Bizim istediğimiz Carlsen’in oyun tarzını öğrenmek ve bunu uygulamak olduğu için bunu uygulayabileceğimiz yöntem yapay zekadan geçiyor .yapay zeka ,makine öğrenmesi ve derin öğrenme bizim bu proje için bilmemiz gereken konulardır .uygulanılan yöntem aşama aşama aşağıda gösterilecektir .

Öncelikle bunlardan bahsedecek olursak;



“ Literatürde ve medyada sıkça işlenen ileri teknoloji ile ilgili farklı kavramların birbiri yerine kullanılması sıkça karşılaşılan bir olgudur. Ancak bir araştırma projesi ya da bir ürün geliştirme projesi için hazırlanacak algoritmanın; yapay zekâ, makine öğrenmesi ya da derin öğrenme algoritması olarak önceden net bir şekilde adlandırılması, ilgili projenin geleceğini olumlu yönde etkileyecektir. Nitekim yapay zekâ, makine öğrenmesi ve derin öğrenme sırasıyla dıştan içe doğru birbirini kapsayan ve aynı amaca yönelmiş teknolojilerdir. Ancak bunların çalışma şekilleri, ihtiyaç duydukları öğrenme veri miktarları ve son olarak çalışmaları için gereken komputasyonel kapasite farklıdır. Şimdi bu kavramları açıklayalım: Yapay Zekâ Yapay zekâ, ister makine öğrenmesi kullansın ister kullanmasın herhangi bir tahmin veya karar işlemini gerçekleştiren teknolojilerin genel adıdır. Genel kanaatin aksine yapay zekâ makine öğrenmesi veya derin öğrenme algoritmaları olmaksızın da çalışan bir algoritma olabilir. Makine öğrenmesi algoritmaları ortaya çıkana kadar yapay zekâ çalışmaları “hard-coded” olarak nitelendirilen yani tüm mantıksal ve matematiksel işlemlerin yazılımcı tarafından bizzat kodlandığı bir yapıya dayanmaktaydı. Örneğin ilk satranç oyuncusu yapay zekâ algoritmaları tamamen böyleydi. Yapay zekânın bu türü sembolik yapay zekâ olarak adlandırılmaktadır. **Makine Öğrenmesi Algoritmaları**, bilgisayarların açıkça programlanmadan bilişsel işlemler yapmasına olanak veren algoritmalar olarak nitelendirilir. Bu tanıma giren algoritmalar tarihsel olarak daha önce ortaya çıkan hard-coded yapay zekâ algoritmalarının yerini almıştır. Makine öğrenmesini hard-coded olarak kodlanmış sembolik yapay zekâ algoritmalarından ayıran özellik algoritmanın tamamen veriden öğrenmesidir. **Derin öğrenme algoritması**, makine öğrenmesi algoritmasının bir alt dalı olup öğrenmeye esas teşkil edilen verilerin makine öğrenmesi algoritması içinde birebir değil de katman olarak ifade edilen ve verinin özel bir tür temsili ile işlev görür² . Bu katmanlar bizim bir bütün olarak algıladığımız sözgelimi bir fotoğrafın en küçük bilgi

içerebilen parçasından tam olarak fotoğrafa dönüşene değin her aşamasını içeren temsili varlıklardır. [8]”.

Makine öğrenmesi ve derin öğrenmeden bahsettiğimize göre artık uygulama şekline geçebiliriz.

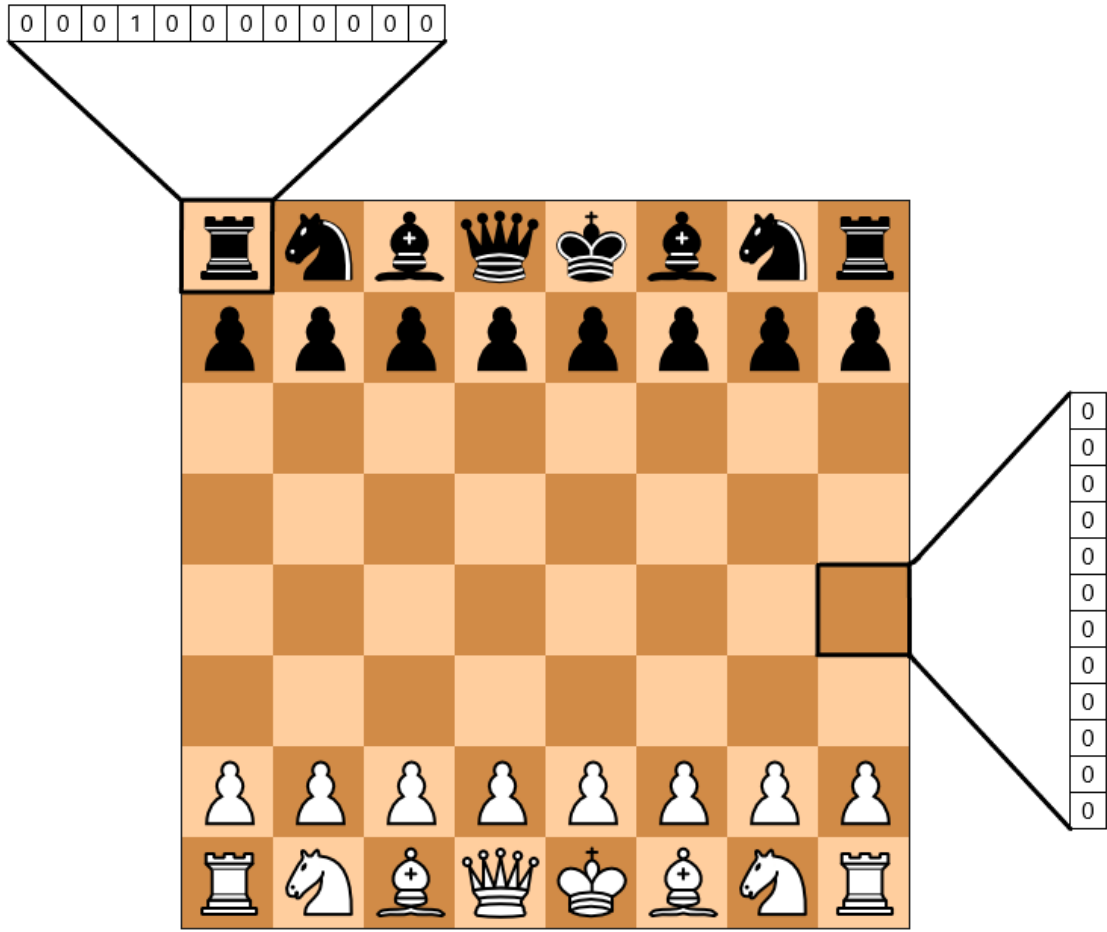


uygulama aşamaları.

Bu aşamaları açacak olursak;

1. Game State fonksiyonu:

burası kodlamanın ilk kısmı .Game State fonksiyonunu koşul ve durumlara göre 0 ve 1'lerden oluşan taş durum kodlamasını yapar .Bu 0-1 durumları taş türü(piyon ,vezir kale vb.) ,rengini(siyah ,beyaz),2 taraflı rok yapma durumunu ,geçerken alma ve oynayan kişiyi gibi değişkenlerdendir .Aşağıda kod bloğunu temsilen bir görsel eklenmiştir .



örnek görsel .

2. Training fonksiyonu:

Bu aşamada veri setindeki oyunlar tek tek alınır .tek tek alınan oyunların sonuçlar ayrı olarak tutulur .Oyunlar Game State fonksiyonu sayesinde encode edilerek bit haline getirilir ve yeni veri setine eklenir .burada 2 farklı yeni veri seti elde etmiş oluruz ilki yeni oyun veri seti .ikincisi oyun sonucu veri seti .yeni veri setlerinin oluştuğundan sonra içeriğinin formata uygunluğu kontrol edilir .kontrol edildikten sonra numpy'ın npy uzantısı halinde 2 veri setinde çevrilir ve kaydedilirler .

Aşağıda bir oyun dosyası formatı ve başlangıç durumunun encode edilmiş şeklini temsil eden iki görsele yer verilmiştir.

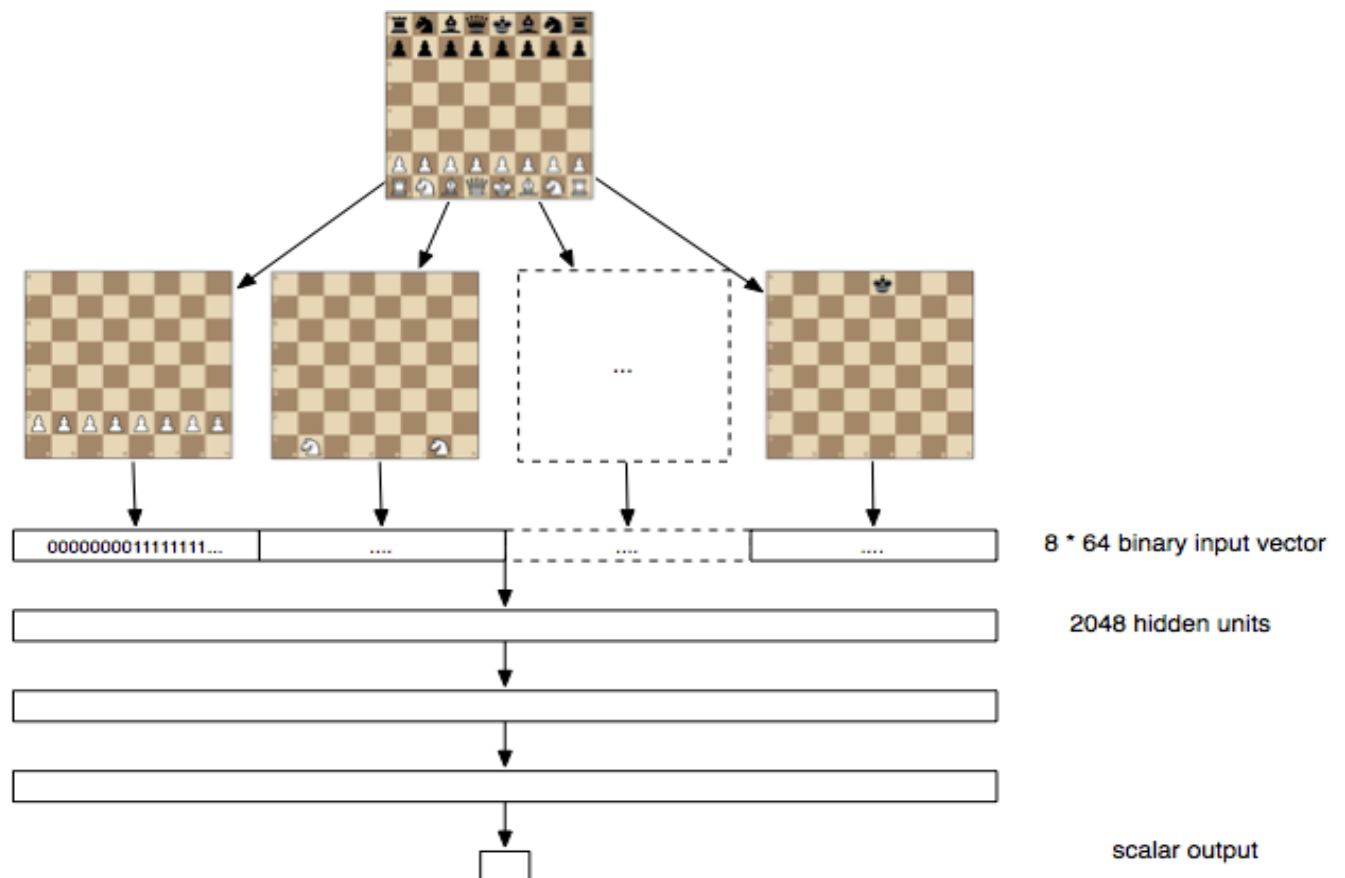
```

kramnik_b33.pgn - Notepad
File Edit Format View Help
[Event "GMA qual"]
[Site "Belgorod"]
[Date "1989.??.??"]
[Round "?"]
[White "Kramnik, Vladimir"]
[Black "Efimov, Igor"]
[Result "1/2-1/2"]
[ECO "B33"]
[BlackElo "2395"]

1. e4 c5 2. Nf3 Nc6 3. d4 cxd4 4. Nxd4 e5 5. Nb5 Nf6 6. N1c3 h6 7. Nd6+ Bxd6 8.
Qxd6 Qe7 9. Qxe7+ Kxe7 10. b3 d6 11. Ba3 a5 12. O-O-O Nb4 13. f3 Bd7 14. Bb2
Rhc8 15. a3 Na6 16. Bxa6 Rxa6 17. a4 Rac6 18. Rd3 Be6 19. Kb1 Nd7 20. Rc1 Nc5
21. Nd5+ Kf8 22. Rd2 f5 23. Re1 fxe4 24. fxe4 Bxd5 25. Rxd5 Ke7 26. Ba3 Rf8 27.
Bxc5 Rxc5 28. Rd2 b5 29. axb5 Rxb5 30. Kb2 Ke6 31. Red1 Rb6 32. Ka3 Rf4 33. Re2
Rb4 34. c4 Rb6 1/2-1/2

```

Temsili oyun dosyası formatı



Temsili oyun encode edilişi .

3. Main:

Main aşamasında yukarıda hazırlanan fonksiyonlar çalıştırılarak veri seti eğitim için uygun hale getirilmesi sağlanır. Veri setimiz oyun dosyalarından 134130 oyun durumu oluşturmaktadır.

4. Pretrained:

Prediction training aşamasında oyun veri setini denetimsiz öğrenme algoritması kullanarak derin öğrenmeden geçiriyoruz. Buradaki amaç konumların ağırlıklarını belirlemektir. bir pozisyonun gücünün kesin bir değeri olmadığı için bunu yapmak zordur. sinir ağıımızı TensorFlow ve Keras kullanarak iki konumu karşılaştıracak şekilde eğittik .Veriyi önce aşırı öğrenmeden kaçınmak için random.shuffle'dan geçirdim .ardından denetimsiz derin öğrenmeye geçtim.

derin öğrenmenin özellikleri ;

5 katmandan [774, 500, 250, 100, 100] .

0.0001 learnin rate (adam).

Loss olarak mse ve metrik olarak mae .

epochs = 25, batch_size = 40 .

Çıktı :

Layer -0

loss: 0.0033 - mae: 0.0034

Layer -1

loss: 0.0076 - mae: 0.0484

Layer -2

loss: 0.0254 - mae: 0.0579

Layer --3

loss: 0.0521 - mae: 0.0802

5. Supervised:

Denetimli öğrenme aşaması. Burada oyun ,oyun sonucu ve denetimsiz öğrenmenin çıktısını alır .denetimsiz öğrenmeden gelen çıktı ,yukarıdan aldığımız oyun sonuçları verilerine göre oyun dosyası üzerine denetimli öğrenme yapıyoruz .

yapacağımız öğrenmenin özellikleri ;

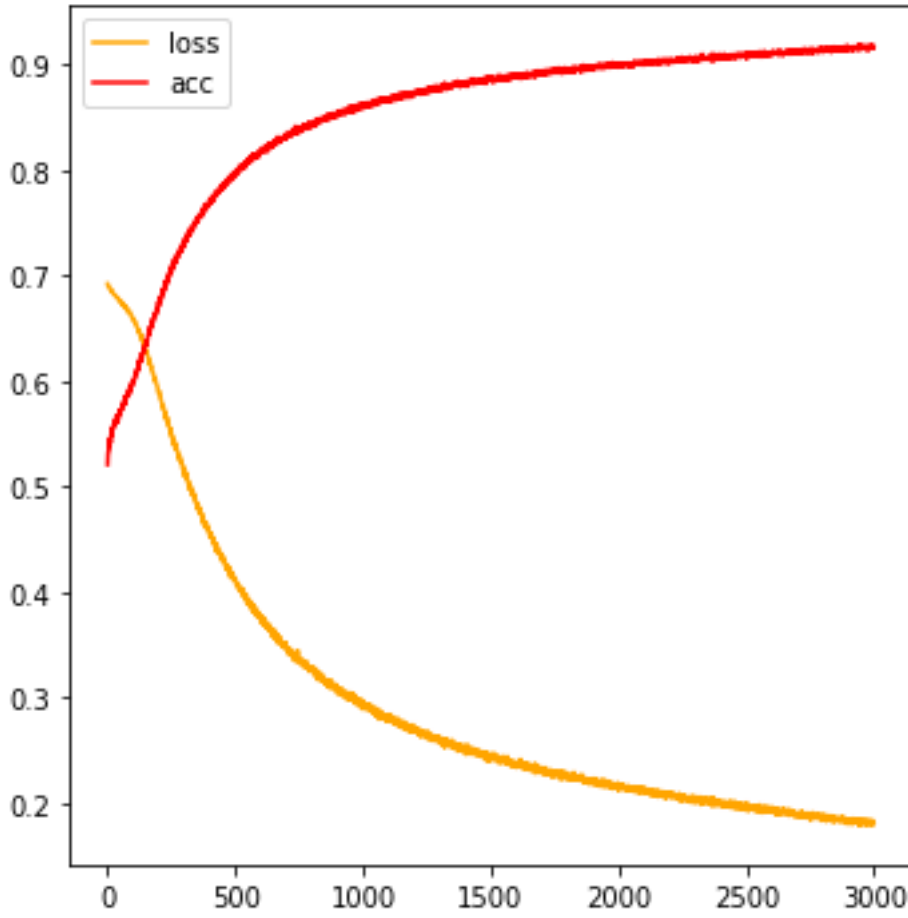
katmanlar : [700, 500, 250, 2] .

girdi katmanında denetimsiz öğrenmenin çıktısı alıyoruz .

0,001 learning rate .

Loss olarak binary crossentropy .metric olarak ise accury aldım .

Son olarak epoch sayısı 3000 olarak bıraktıdım .



Loss ve Accuracy değeri tablosu

Sonuç olarak 3000 epoch'ta Loss: 0.1823 , accuracy : 0.9153 olarak bulundu .

6. Play :

Burada alfa-beta budama yöntemiyle kullanıcının istediği derinlikte arama yapılır ve bu arama sonucu oluşan en iyi sonuç hamle olarak oynanır .Oyun insan ile algoritma arasında oynanacak şekilde tasarlanmıştır .beyaz bilgisayar siyah ise kullanıcı olarak belirlenmiştir .

Aşağıda program ile oynadığım örnek açılış hamleleri gösterilmiştir.







Yukarıda oynanan açılış pek bilinmeyen ve tercih edilmeyen amar açılışıdır. Amar'dan biraz bahsedecek olursak :

“Amar Açılışı (Paris Açılışı veya Sarhoş Şövalye Açılışı olarak da bilinir) , hamle ile tanımlanan bir satranç açılışıdır :

1. Ah3

Durkin Açılışını "Sodyum Saldırısı" olarak adlandırmaya benzer şekilde, bu açılış Amonyak Açılımı olarak adlandırılabilir, çünkü 1.Nh3 cebirsel notasyonu amonyak için NH3 kimyasal formülüne benzer. Parisli amatör Charles Amar 1930'larda oynadı. Muhtemelen bu açılış için her iki adı da kullanan Savielly Tartakower tarafından adlandırılmıştır, ancak satranç yazarı Tim Harding şaka yollu "Amar" ın "Kesinlikle çılgın ve saçma" nın kısaltması olduğunu öne sürmüştür. [9]

1.Ah3 düzensiz bir açılış olarak kabul edildiğinden, Encyclopaedia of Chess Openings'de A00 kodu altında sınıflandırılır.”

5. Sonuç:

Oluşturduğum bu projede 0.9165 doğruluk ve 0.1814 hata oranı almamıza rağmen model istediğim düzeyde gelememiştir. Kullandığım bilgisayar donanımı daha yüksek arama değerlerini kaldıramadığı için 4'ün üzerinde arama yapılamamıştır. Sonuç olarak projeyi istediğim düzeye getiremeden bitirmek durumunda kaldım. Daha yüksek arama derinliği ve farklı parametreler kullanılarak model şuan ki seviyesinin üzerine çıkarılabilir. Ayrıca veri seti yeterli miktarda bulunmadığından kaynaklı da böyle bir sonucun çıkmasına neden olmuş

olabilir. Bu çalışmada daha ilerisine gidilmeyecektir. Çalışmada bana yardımcı olan tüm Hocalarım ve arkadaşlarıma teşekkür ederim.

7. Kaynaklar:

1. Time Magazine (27 Aralık 2014). "Carlsen". Time. 22 Ağustos 2015 tarihinde kaynağından arşivlendi. Erişim tarihi: 27 Aralık 2014.
2. FIDE (Aralık 2013). "Chennai 2013 Dünya Satranç Şampiyonluk Maçı resmi ana sayfası". Time. 25 Şubat 2016 tarihinde kaynağından arşivlendi. Erişim tarihi: 29 Aralık 2013.
3. FIDE (Aralık 2014). "Sochi 2014 Dünya Satranç Şampiyonluk Maçı resmi ana sayfası". Time. 30 Ocak 2016 tarihinde kaynağından arşivlendi. Erişim tarihi: 27 Aralık 2014.
4. Graham, Bryan Armen (1 Aralık 2016). "Magnus Carlsen retains world chess title after quickfire tie-breaker". The Guardian (İngilizce). ISSN 0261-3077. 21 Aralık 2019 tarihinde kaynağından arşivlendi. Erişim tarihi: 23 Eylül 2020.
5. WGM Raluca Sgîrcea, IM Renier Castellanos(09.04.2017) ." 10 Things We All Should Learn from Magnus Carlsen ." , thechessworld.com .
6. Jordan Novet.(Ocak, 18, 2014) "What tech founders can learn from chess champion Magnus Carlsen" , Venture Beat .
7. Reid McIlroy-Young, Ashton Anderson, Siddhartha Şen , Jon Kleinberg . "Aligning Superhuman AI with Human Behavior: Chess as a Model System". arXiv:2006.01855v3 [cs.AI] 14 Jul 2020 .
8. STM ThinkTech & Başkent Üniversitesi(14.11.2018)." Derin Farklar: Yapay Zekâ, Makine Öğrenmesi ve Derin Öğrenme", thinktech.stm.com.tr .
9. Hopper, David ; Whyld, Kenneth (1996). The Oxford Companion To Chess (ikinci baskı). Oxford: Oxford Üniversitesi Yayınları. ISBN 0-19-280049-3.