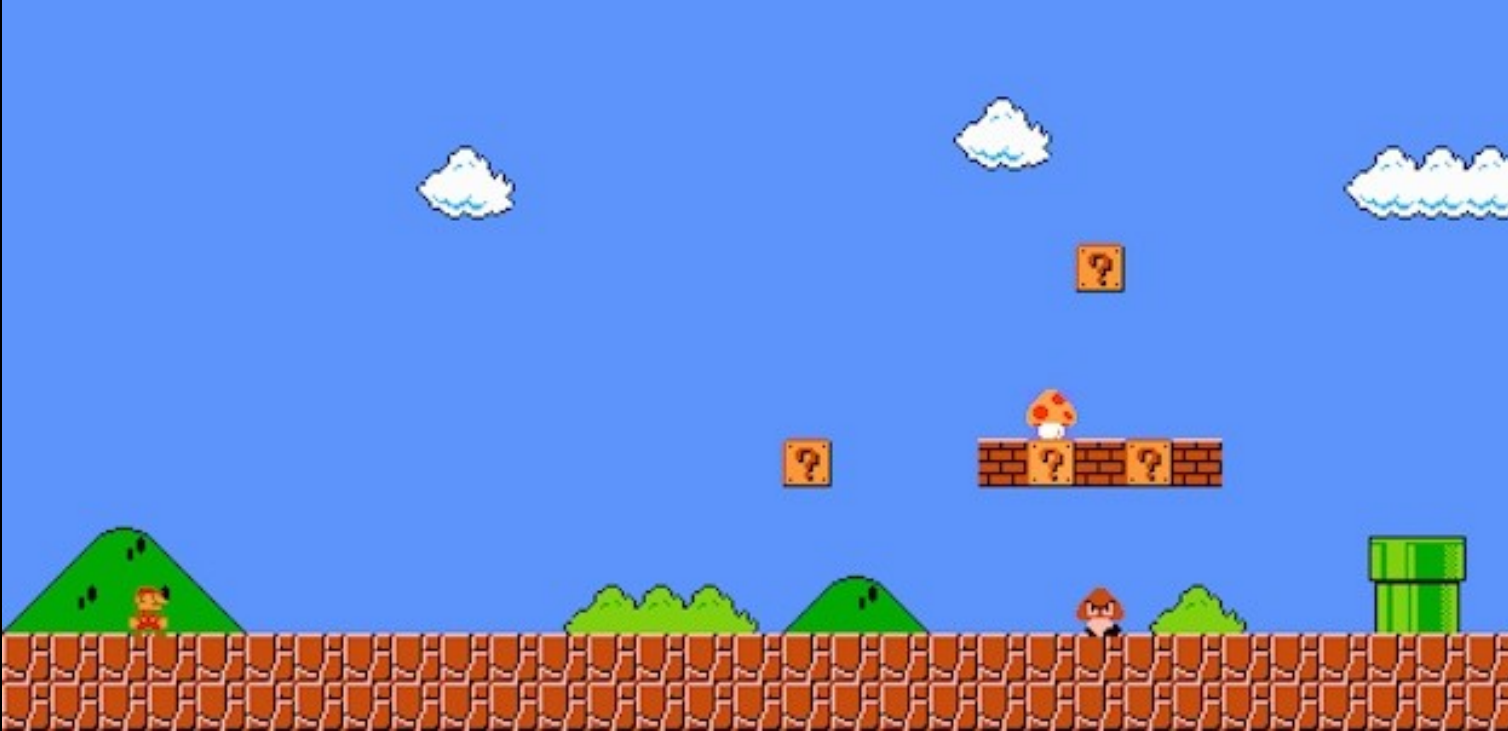


Mario AI

https://github.com/NeIseZitti/Mario_AI

Super Mario Bros diye bi oyun var. Mario var kořup zıplıyo.

Bizim projemizin amacı Mario'nun bir bölümü en kısa sürede bitirmesini sağlamak.



Bu isteğimizi gerçekleştirmenin birkaç yolu var.

Biz bu yollar arasından genetik algoritma ve sinir ağının birlikte kullanımını seçtik.

Peki neden bunu seçtik?

Bu isteğimizi gerçekleştirmenin birkaç yolu var.

Biz bu yollar arasından genetik algoritma ve sinir ağının birlikte kullanımını seçtik.

Peki neden bunu seçtik?

Cevap 1: Marionun içinde bulunduğu durumdan hareketle yapacağı hareketleri döndüren başka bir sistemin kurulmasının zor olması. Kısacası problemin karmaşıklığı.

Bu isteğimizi gerçekleştirmenin birkaç yolu var.

Biz bu yollar arasından genetik algoritma ve sinir ağının birlikte kullanımını seçtik.

Peki neden bunu seçtik?

Cevap 1: Marionun içinde bulunduğu durumdan hareketle yapacağı hareketleri döndüren başka bir sistemin kurulmasının zor olması. Kısacası problemin karmaşıklığı.

Cevap 2: Elimizde Mario'nun hangi durumlarda ne yapması gerektiğine dair önceden hazırlanmış bir veri bulunmaması. (Öyle olsaydı makine öğrenmesi kullanabilirdik.)

Bu isteğimizi gerçekleştirmenin birkaç yolu var.

Biz bu yollar arasından genetik algoritma ve sinir ağının birlikte kullanımını seçtik.

Peki neden bunu seçtik?

Cevap 1: Marionun içinde bulunduğu durumdan hareketle yapacağı hareketleri döndüren başka bir sistemin kurulmasının zor olması. Kısacası problemin karmaşıklığı.

Cevap 2: Elimizde Mario'nun hangi durumlarda ne yapması gerektiğine dair önceden hazırlanmış bir veri bulunmaması. (Öyle olsaydı makine öğrenmesi kullanabilirdik.)

Cevap 3: Youtube'daki abilerin videolarına özenmek.

Genetik algoritma biyolojik evrimden esinlenen ve taklit etmeye çalışan bir optimizasyon yöntemidir. Çözüm bulmak için doğal seçilim, çaprazlama (crossover), mutasyon ve elitizm gibi evrimsel mekanizmaları kullanır.

Genetik algoritma biyolojik evrimden esinlenen ve taklit etmeye çalışan bir optimizasyon yöntemidir. Çözüm bulmak için doğal seçilim, çaprazlama (crossover), mutasyon ve elitizm gibi evrimsel mekanizmaları kullanır.

Bir başka bakış açıysındansa genetik algoritma bir arama algoritmasıdır. Verilen problem için en iyi çözümü arar.

Genetik algoritma biyolojik evrimden esinlenen ve taklit etmeye çalışan bir optimizasyon yöntemidir. Çözüm bulmak için doğal seçilim, çaprazlama (crossover), mutasyon ve elitizm gibi evrimsel mekanizmaları kullanır.

Bir başka bakış açıysındansa genetik algoritma bir arama algoritmasıdır. Verilen problem için en iyi çözümü arar.

Ortada tek bir genetik algoritma yok. Tüm parçalarını isteğe göre tasarlayıp farklı algoritmalar elde edilebiliyo. Bu da onu esnek yapıyo.

Genetik algoritma biyolojik evrimden esinlenen ve taklit etmeye çalışan bir optimizasyon yöntemidir. Çözüm bulmak için doğal seçim, çaprazlama (crossover), mutasyon ve elitizm gibi evrimsel mekanizmaları kullanır.

Bir başka bakış açıysındansa genetik algoritma bir arama algoritmasıdır. Verilen problem için en iyi çözümü arar.

Ortada tek bir genetik algoritma yok. Tüm parçalarını isteğe göre tasarlayıp farklı algoritmalar elde edilebiliyo. Bu da onu esnek yapıyo.

Sinir ağı ise insan beynindeki nöronların çalışma prensibini taklit eden bir makine öğrenmesi modelidir. Verilen girdilere göre çıktı üretebilen matematiksel bir yapıdan oluşur.

Genetik algoritma biyolojik evrimden esinlenen ve taklit etmeye çalışan bir optimizasyon yöntemidir. Çözüm bulmak için doğal seçilim, çaprazlama (crossover), mutasyon ve elitizm gibi evrimsel mekanizmaları kullanır.

Bir başka bakış açıysındansa genetik algoritma bir arama algoritmasıdır. Verilen problem için en iyi çözümü arar.

Ortada tek bir genetik algoritma yok. Tüm parçalarını isteğe göre tasarlayıp farklı algoritmalar elde edilebiliyo. Bu da onu esnek yapıyo.

Sinir ağı ise insan beynindeki nöronların çalışma prensibini taklit eden bir makine öğrenmesi modelidir. Verilen girdilere göre çıktı üretebilen matematiksel bir yapıdan oluşur.

Evrensel yakınsama teoremi: Bir başka açıdan bakıldığında sinir ağları; yeterince gizli katmana(hidden layer) ve nörona sahip olduğunda, sürekli bir fonksiyonu, verilen doğruluk seviyesinde yaklaşık olarak modelleyebilir. Bu, sinir ağlarının teorik olarak herhangi bir fonksiyonu öğrenebileceğini gösterir. Örneğin: Lineer, lineer olmayan, trigonometrik, polinomiye, logaritmik veya DAHA KARMAŞIK fonksiyonlar.

Kısaca bizim projemiz bazında;

Kısaca bizim projemiz bazında;

Genetik algoritma çözümü çok karmaşık fonksiyonları optimize eden bir zimbırtı.

Kısaca bizim projemiz bazında;

Genetik algoritma çözümü çok karmaşık fonksiyonları optimize eden bir zımbırtı.

Fakat elimizde problem var kesin bir fonksiyon yok.

Kısaca bizim projemiz bazında;

Genetik algoritma çözümü çok karmaşık fonksiyonları optimize eden bir zımbırtı.

Fakat elimizde problem var kesin bir fonksiyon yok.

Sinir ağı ise karmaşıklığı yeterli düzeydeyse herhangi bir fonksiyonun yerine konabilecek ve onu taklit edebilecek bir replika.

Kısaca bizim projemiz bazında;

Genetik algoritma çözümü çok karmaşık fonksiyonları optimize eden bir zımbırtı.

Fakat elimizde problem var kesin bir fonksiyon yok.

Sinir ağı ise karmaşıklığı yeterli düzeydeyse herhangi bir fonksiyonun yerine konabilecek ve onu taklit edebilecek bir replika.

Bu iki zımbırtı birbirlerini tamamlar nitelikte.

Genetik algoritmadan önce sinir ağına değinirsek:

Genetik aloritmadan önce sinir ağına değinirsek:

Bize girdi lazım.

Genetik algoritmadan önce sinir ağına değinirsek:

Bize girdi lazım.

Girdi için oyunla etkileşime girmek lazım.

Genetik algoritmadan önce sinir ağına değinirsek:

Bize girdi lazım.

Girdi için oyunla etkileşime girmek lazım.

Elde edeceğimiz girdi sayısına uygun bi sinir ağı seçmek lazım.

Genetik aloritmadan önce sinir ağına değinirsek:

Bize girdi lazım.

Girdi için oyunla etkileşime girmek lazım.

Elde edeceğimiz girdi sayısına uygun bi sinir ağı seçmek lazım.

Bu sinir ağını eğitmek lazım.

Genetik aloritmadan önce sinir ağına değinirsek:

Bize girdi lazım.

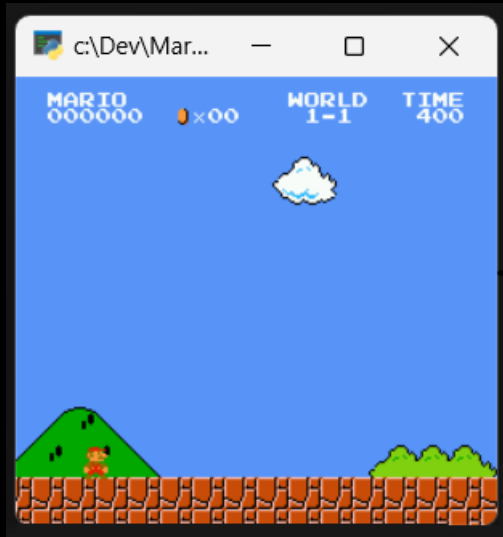
Girdi için oyunla etkileşime girmek lazım.

Elde edeceğimiz girdi sayısına uygun bi sinir ağı seçmek lazım.

Bu sinir ağını eğitmek lazım.

Sinir ağından çıkacak çıktıları oyuna vermek lazım.

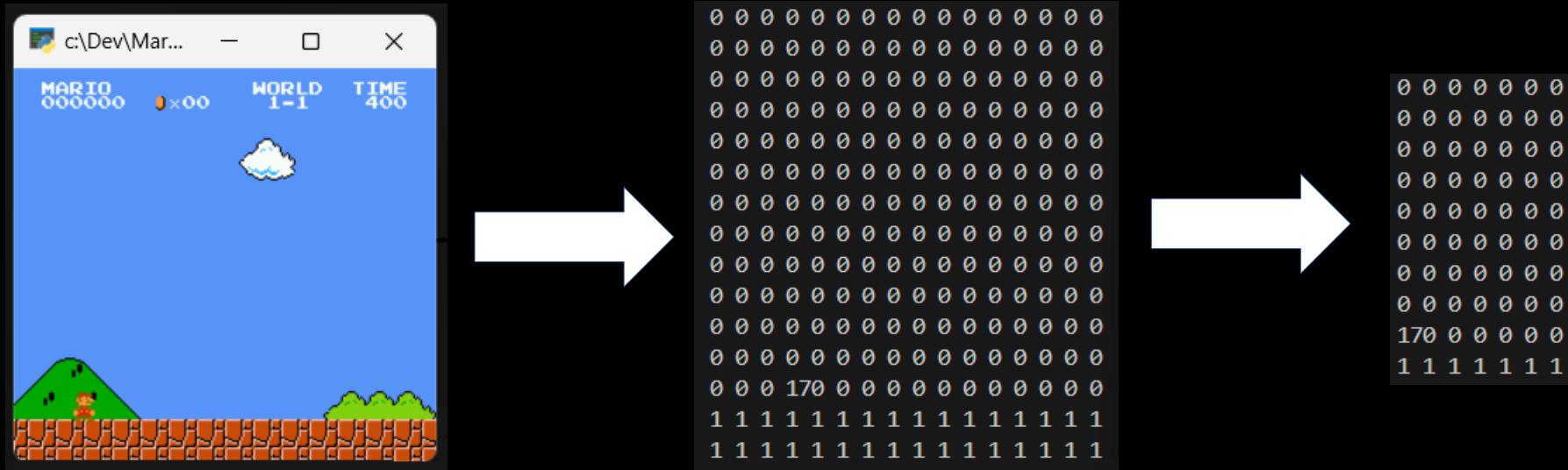
Mario oyunu bloklardan oluşuyo görüldüğü gibi.



Mario oyunu bloklardan oluşuyo görüldüğü gibi.

Biz bu blokları sınıflayıp sayı karşılıklarını elde ettik.

Tüm bu sayıları girdi olarak vermemek için önemli olduklarını düşündüğümüz bir kısmını ayırdık ve onları girdi olarak aldık.



0 boşluğu, 1 yapıları, 170 mario'yu, 255 de canavarları temsil edecek şekilde seçmiş olduk.

0 boşluğu, 1 yapıları, 170 mario'yu, 255 de canavarları temsil edecek şekilde seçmiş olduk.

Fakat bu değerler büyük değerler olduğundan sinir ağının düzgün çalışmamasına neden olabilirdi. Bu yüzden biz de bu değerleri normalize ettik. Yani 0 ve 1 arasına sıkıştırdık.

0 boşluğu, 1 yapıları, 170 mario'yu, 255 de canavarları temsil edecek şekilde seçmiş olduk.

Fakat bu değerler büyük değerler olduğundan sinir ağının düzgün çalışmamasına neden olabilirdi. Bu yüzden biz de bu değerleri normalize ettik. Yani 0 ve 1 arasına sıkıştırdık.

70 girdi değerine ve 6 çıktı değerine(basılacak tuşlar) uygun bir sinir ağı belirlemek lazımdı. Biz de 70x64x32x6 şeklinde bir sinir ağına karar verdik. (6720 weight ve 102 bias)

0 boşluğu, 1 yapıları, 170 mario'yu, 255 de canavarları temsil edecek şekilde seçmiş olduk.

Fakat bu değerler büyük değerler olduğundan sinir ağının düzgün çalışmamasına neden olabilirdi. Bu yüzden biz de bu değerleri normalize ettik. Yani 0 ve 1 arasına sıkıştırdık.

70 girdi değerine ve 6 çıktı değerine(basılacak tuşlar) uygun bir sinir ağı belirlemek lazımdı. Biz de 70x64x32x6 şeklinde bir sinir ağına karar verdik. (6720 weight ve 102 bias)

İlk katmandan ikincisine ve ikinci katmandan üçüncüsüne relu, üçüncü katmandan son katmana da sigmoid aktivasyon fonksiyonunu seçtik.

0 boşluğu, 1 yapıları, 170 mario'yu, 255 de canavarları temsil edecek şekilde seçmiş olduk.

Fakat bu değerler büyük değerler olduğundan sinir ağının düzgün çalışmamasına neden olabilirdi. Bu yüzden biz de bu değerleri normalize ettik. Yani 0 ve 1 arasına sıkıştırdık.

70 girdi değerine ve 6 çıktı değerine(basılacak tuşlar) uygun bir sinir ağı belirlemek lazımdı. Biz de 70x64x32x6 şeklinde bir sinir ağına karar verdik. (6720 weight ve 102 bias)

İlk katmandan ikincisine ve ikinci katmandan üçüncüsüne relu, üçüncü katmandan son katmana da sigmoid aktivasyon fonksiyonunu seçtik.

Çıktımızı, tuşa basılıp basılmadığını temsil edecek 0 ya da 1 cinsinden istediğimizden sigmoid fonksiyonunun çıktılarının; 0.5'ten küçük olanlarını 0 , 0.5'ten büyük olanlarını 1 aldık.

Sinir ađımızı bitirdikten sonra sıra genetik algoritmaya geldi.

Sinir ağıımızı bitirdikten sonra sıra genetik algoritmaya geldi.

Genetik algoritmayı kullanma amacımız sinir ağının weight ve bias değerlerinin en optimum halini bulmak.

Sinir ağıımızı bitirdikten sonra sıra genetik algoritmaya geldi.

Genetik algoritmayı kullanma amacımız sinir ağının weight ve bias değerlerinin en optimum halini bulmak.

Bir nesildeki mario sayısını ve seçilecek mario sayısını belirlemek lazımdı. Biz de her nesil için 64 ve seçilecek 16 mario belirledik.

Sinir ağıımızı bitirdikten sonra sıra genetik algoritmaya geldi.

Genetik algoritmayı kullanma amacımız sinir ağının weight ve bias değerlerinin en optimum halini bulmak.

Bir nesildeki mario sayısını ve seçilecek mario sayısını belirlemek lazımdı. Biz de her nesil için 64 ve seçilecek 16 mario belirledik.

Bu iş çok uzun süreceğinden ve elde edeceğimiz sonuç değerli olacağından ve eğitimi yarıda bırakıp tekrar kaldığımız yerden devam etmek istediğimizden bu değerleri kaydetmek istedik. Bu kayıt etmek istediğimiz verileri bir json dosyasında tutmaya karar verdik.

Sinir ağıımızı bitirdikten sonra sıra genetik algoritmaya geldi.

Genetik algoritmayı kullanma amacımız sinir ağının weight ve bias değerlerinin en optimum halini bulmak.

Bir nesildeki mario sayısını ve seçilecek mario sayısını belirlemek lazımdı. Biz de her nesil için 64 ve seçilecek 16 mario belirledik.

Bu iş çok uzun süreceğinden ve elde edeceğimiz sonuç değerli olacağından ve eğitimi yarıda bırakıp tekrar kaldığımız yerden devam etmek istediğimizden bu değerleri kaydetmek istedik. Bu kayıt etmek istediğimiz verileri bir json dosyasında tutmaya karar verdik.

Başlangıç popülasyonunu oluşturmak için chromosome generator isminde bir kod yazdık. Bu kod 16 mario için -1 ve 1 aralığında tamamen rastgele $16 \times (6720 + 108) = 109152$ değer oluşturdu. Sonra bu değerleri bir chromosomes isimindeki json dosyasına kaydetti.

Kod çalışmaya başladığında json dosyasından değerleri alıp bir arraye koyduk.

Kod çalışmaya başladığında json dosyasından değerleri alıp bir arraye koyduk.

Sonra o değerlere çaprazlama uygulayarak 64 mario'ya yükselttik.
(her mario 6822 değer olduğundan $64 \times 6822 = 436608$ değişkenimiz oldu.)

Kod çalışmaya başladığında json dosyasından değerleri alıp bir arraye koyduk.

Sonra o değerlere çaprazlama uygulayarak 64 mario'ya yükselttik.
(her mario 6822 değer olduğundan $64 \times 6822 = 436608$ değişkenimiz oldu.)

Yeni mario'lar üzerine mutasyon uyguladık.

Kod çalışmaya başladığında json dosyasından değerleri alıp bir arraye koyduk.

Sonra o değerlere çaprazlama uygulayarak 64 mario'ya yükselttik.
(her mario 6822 değer olduğundan $64 \times 6822 = 436608$ değişkenimiz oldu.)

Yeni mario'lar üzerine mutasyon uyguladık.

Sırayla her mario için elde ettiğimiz 6720 weight ve 102 bias'i sinir ağına yerleştirdik.

Kod çalışmaya başladığında json dosyasından değerleri alıp bir arraye koyduk.

Sonra o değerlere çaprazlama uygulayarak 64 mario'ya yükselttik.
(her mario 6822 değer olduğundan $64 \times 6822 = 436608$ değişkenimiz oldu.)

Yeni mario'lar üzerine mutasyon uyguladık.

Sırayla her mario için elde ettiğimiz 6720 weight ve 102 bias'i sinir ağına yerleştirdik.

Sinir ağına her frame için girdilerimizi verdik ve çıktılarımızı aldık.

Kod çalışmaya başladığında json dosyasından değerleri alıp bir arraye koyduk.

Sonra o değerlere çaprazlama uygulayarak 64 mario'ya yükselttik.
(her mario 6822 değer olduğundan $64 \times 6822 = 436608$ değişkenimiz oldu.)

Yeni mario'lar üzerine mutasyon uyguladık.

Sırayla her mario için elde ettiğimiz 6720 weight ve 102 bias'i sinir ağına yerleştirdik.

Sinir ağına her frame için girdilerimizi verdik ve çıktılarımızı aldık.

Aldığımız çıktıları oyuna verdik ve sonraki frame'i elde ettik.

Kod çalışmaya başladığında json dosyasından değerleri alıp bir arraye koyduk.

Sonra o değerlere çaprazlama uygulayarak 64 mario'ya yükselttik.
(her mario 6822 değer olduğundan $64 \times 6822 = 436608$ değişkenimiz oldu.)

Yeni mario'lar üzerine mutasyon uyguladık.

Sırayla her mario için elde ettiğimiz 6720 weight ve 102 bias'i sinir ağına yerleştirdik.

Sinir ağına her frame için girdilerimizi verdik ve çıktılarımızı aldık.

Aldığımız çıktıları oyuna verdik ve sonraki frame'i elde ettik.

Mario'nun işi bitene kadar bu işlemi sürdürdük.

Kod çalışmaya başladığında json dosyasından değerleri alıp bir arraye koyduk.

Sonra o değerlere çaprazlama uygulayarak 64 mario'ya yükselttik.
(her mario 6822 değer olduğundan $64 \times 6822 = 436608$ değişkenimiz oldu.)

Yeni mario'lar üzerine mutasyon uyguladık.

Sırayla her mario için elde ettiğimiz 6720 weight ve 102 bias'i sinir ağına yerleştirdik.

Sinir ağına her frame için girdilerimizi verdik ve çıktılarımızı aldık.

Aldığımız çıktıları oyuna verdik ve sonraki frame'i elde ettik.

Mario'nun işi bitene kadar bu işlemi sürdürdük.

Mario'nun işi bittiğinde onun uygunluk değerini hesapladık.

Uygunluk deęerini hesaplamak bařlı bařına bir problem. Eęer uygunluęu hesaplayan fonksiyon dűzgűn tasarlanmamıřsa alacaęımız sonu da dűzgűn olmaz.

Uygunluk deęerini hesaplamak bařlı bařına bir problem. Eęer uygunluęu hesaplayan fonksiyon dűzgűn tasarlanmamıřsa alacaęımız sonu da dűzgűn olmaz.

Biz uygunluęu marionun gittięi mesafeye, o mesafeyi ne kadar sűrede gittięine ve marionun oyunu bitirip bitiremedięine gűre belirledik.

Uygunluk değerini hesaplamak başlı başına bir problem. Eğer uygunluğu hesaplayan fonksiyon düzgün tasarlanmamışsa alacağımız sonuç da düzgün olmaz.

Biz uygunluğu marionun gittiği mesafeye, o mesafeyi ne kadar sürede gittiğine ve marionun oyunu bitirip bitiremediğine göre belirledik.

Her mario için sırayla uygunluk değerini hesapladık.

Uygunluk değerini hesaplamak başlı başına bir problem. Eğer uygunluğu hesaplayan fonksiyon düzgün tasarlanmamışsa alacağımız sonuç da düzgün olmaz.

Biz uygunluğu marionun gittiği mesafeye, o mesafeyi ne kadar sürede gittiğine ve marionun oyunu bitirip bitiremediğine göre belirledik.

Her mario için sırayla uygunluk değerini hesapladık.

İstatistik tutmak için her neslin sonunda bazı verileri kaydettik.

Uygunluk değerini hesaplamak başlı başına bir problem. Eğer uygunluğu hesaplayan fonksiyon düzgün tasarlanmamışsa alacağımız sonuç da düzgün olmaz.

Biz uygunluğu marionun gittiği mesafeye, o mesafeyi ne kadar sürede gittiğine ve marionun oyunu bitirip bitiremediğine göre belirledik.

Her mario için sırayla uygunluk değerini hesapladık.

İstatistik tutmak için her neslin sonunda bazı verileri kaydettik.

Uygunluk değerlerine göre 64 tane mario'nun içinden 16 tanesini seçtik.

Uygunluk deęerini hesaplamak bařlı bařına bir problem. Eęer uygunluęu hesaplayan fonksiyon dűzgűn tasarlanmamıřsa alacaęımız sonu da dűzgűn olmaz.

Biz uygunluęu marionun gittięi mesafeye, o mesafeyi ne kadar sűrede gittięine ve marionun oyunu bitirip bitiremedięine gűre belirledik.

Her mario iin sırayla uygunluk deęerini hesapladık.

İstatistik tutmak iin her neslin sonunda bazı verileri kaydettik.

Uygunluk deęerlerine gűre 64 tane mario'nun iinden 16 tanesini setik.

Bu seim iřlemini 4 tanesini deterministik seimle kalan 12 tanesini de rulet seimiyle belirledik. Kısaca elitist rulet seimi uygulamıř olduk.

Seçilen mario'ları tuttuk ve geri kalanını attık.

Seçilen mario'ları tuttuk ve geri kalanını attık.

Yeni marioların kromozomlarını(weight ve bias'lerini) chromosomes adlı json dosyamıza kaydettik.

Seçilen mario'ları tuttuk ve geri kalanını attık.

Yeni marioların kromozomlarını(weight ve bias'lerini) chromosomes adlı json dosyamıza kaydettik.

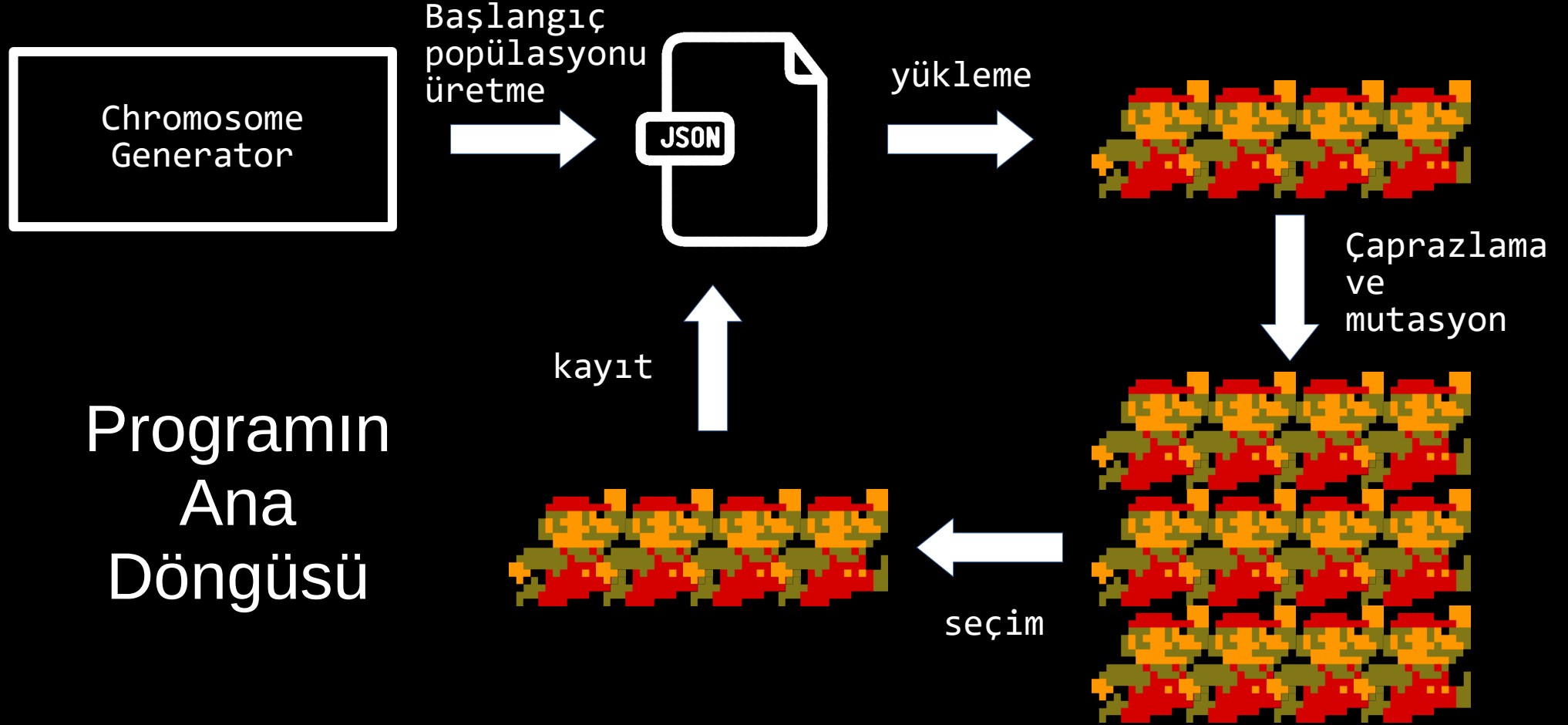
Kalan 16 mario 2. nesil için ebeveyn oldu ve tekrar çaprazlama uyguladık...

Seçilen mario'ları tuttuk ve geri kalanını attık.

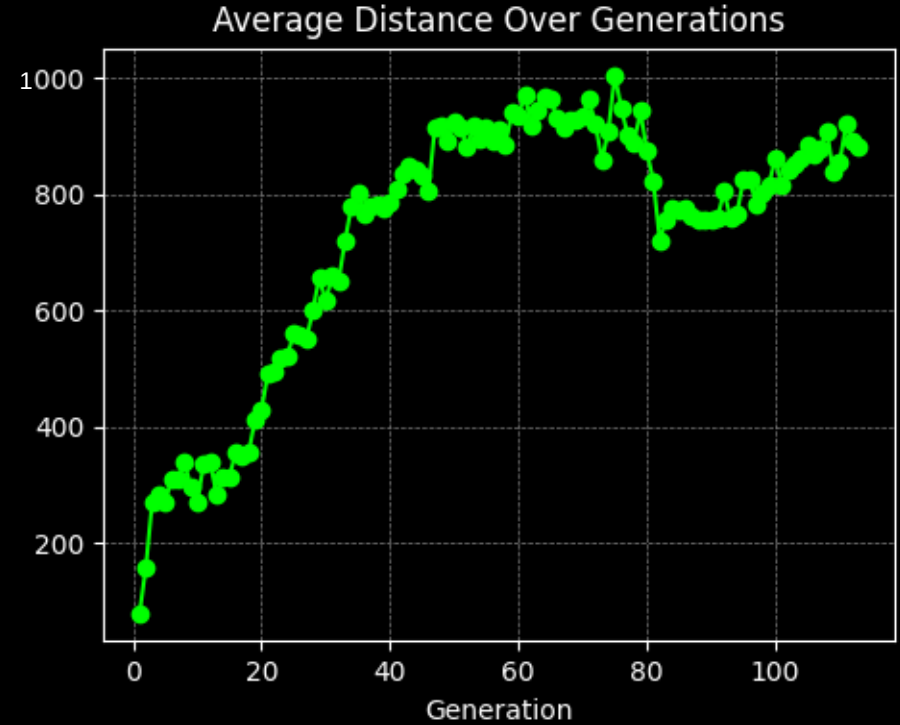
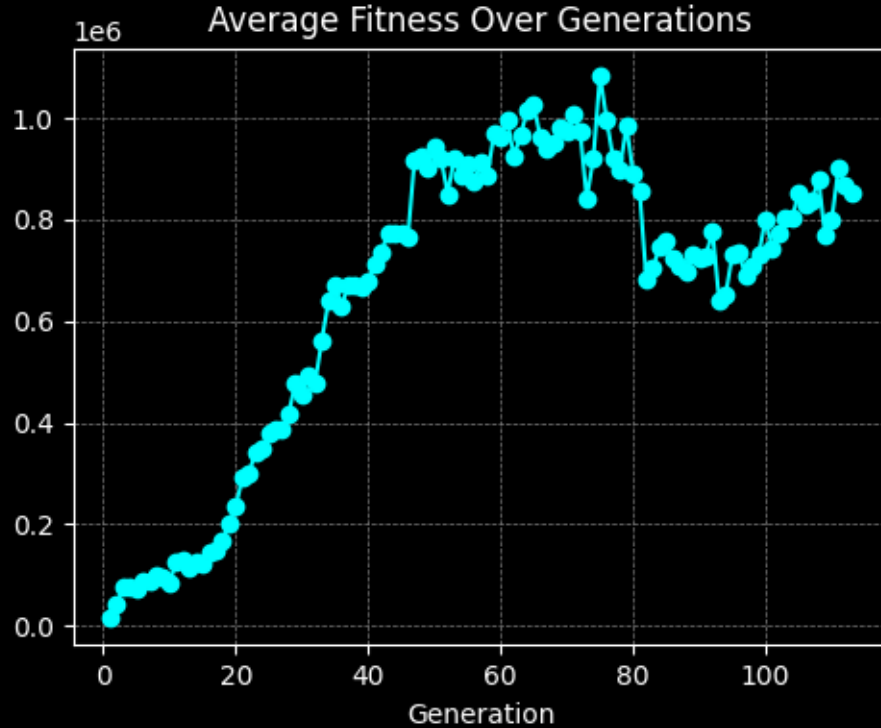
Yeni marioların kromozomlarını(weight ve bias'lerini) chromosomes adlı json dosyamıza kaydettik.

Kalan 16 mario 2. nesil için ebeveyn oldu ve tekrar çaprazlama uyguladık...

Bu şekilde nesilden nesile döngü devam etti. Ve her neslin bir öncekinden daha iyi olmasını umduk ve öyle de oldu.



Anlattığımız şekilde yapay zekayı eğittik ve şöyle bir sonuç elde ettik.



Günün sonunda mario bölümü tamamlayamasa da hayli yol katetti.

Günün sonunda mario bölümü tamamlayamasa da hayli yol katetti.

Tamamlayamamasının olası sebepleri:

Günün sonunda mario bölümü tamamlayamasa da hayli yol katetti.

Tamamlayamamasının olası sebepleri:

Daha fazla zamana veya işlem gücüne ihtiyaç olması

Günün sonunda mario bölümü tamamlayamasa da hayli yol katetti.

Tamamlayamamasının olası sebepleri:

Daha fazla zamana veya işlem gücüne ihtiyaç olması

Yeterince karmaşık bir sinir ağı belirlemememiz

Günün sonunda mario bölümü tamamlayamasa da hayli yol katetti.

Tamamlayamamasının olası sebepleri:

Daha fazla zamana veya işlem gücüne ihtiyaç olması

Yeterince karmaşık bir sinir ağı belirlemememiz

Genetik algoritmanın tasarımının kötü olması nedeniyle çeşitliliğin azalması ve yerel optimuma sıkışmamız

Günün sonunda mario bölümü tamamlayamasa da hayli yol katetti.

Tamamlayamamasının olası sebepleri:

Daha fazla zamana veya işlem gücüne ihtiyaç olması

Yeterince karmaşık bir sinir ağı belirlemememiz

Genetik algoritmanın tasarımının kötü olması nedeniyle çeşitliliğin azalması ve yerel optimuma sıkışmamız

Uygunluk fonksiyonunu yanlış tasarlamamız

Günün sonunda mario bölümü tamamlayamasa da hayli yol katetti.

Tamamlayamamasının olası sebepleri:

Daha fazla zamana veya işlem gücüne ihtiyaç olması

Yeterince karmaşık bir sinir ağı belirlemememiz

Genetik algoritmanın tasarımının kötü olması nedeniyle çeşitliliğin azalması ve yerel optimuma sıkışmamız

Uygunluk fonksiyonunu yanlış tasarlamamız

Seçtiğimiz girdilerin eksik veya alakasız olması

Projenin eksikleri:

Projenin eksikleri:

Kullandığımız algoritmaları ve parçalarını gelişigüzel keyfimize göre tasarladık.

Projenin eksikleri:

Kullandığımız algoritmaları ve parçalarını gelişigüzel keyfimize göre tasarladık.

CPU ile tek thread üzerinde çalıştırdık. Kolayca paralellenebilecek bir problem olduğundan GPU kullanabilirdik.

Projenin eksikleri:

Kullandığımız algoritmaları ve parçalarını gelişigüzel keyfimize göre tasarladık.

CPU ile tek thread üzerinde çalıştırdık. Kolayca paralellenebilecek bir problem olduğundan GPU kullanabilirdik.

Kod; estetik değil, optimize değil, sürdürülebilir değil, çok amaçlı değil. GUI yok. Paralellik yok. C++ ve Rust gibi düşük seviyeli bir dil yerine Python kullandık.



Son