

NÖROEVRİM ALGORİTMASIYLA OYUN OYNAYABİLEN YAPAY ZEKA GELİŞTİRME



BİTİRME PROJESİ 2

Furkan Kaya

191216002

DANIŞMAN

Öğr. Gör. Ezgi Özer

17 Nisan 2021

ÖZET

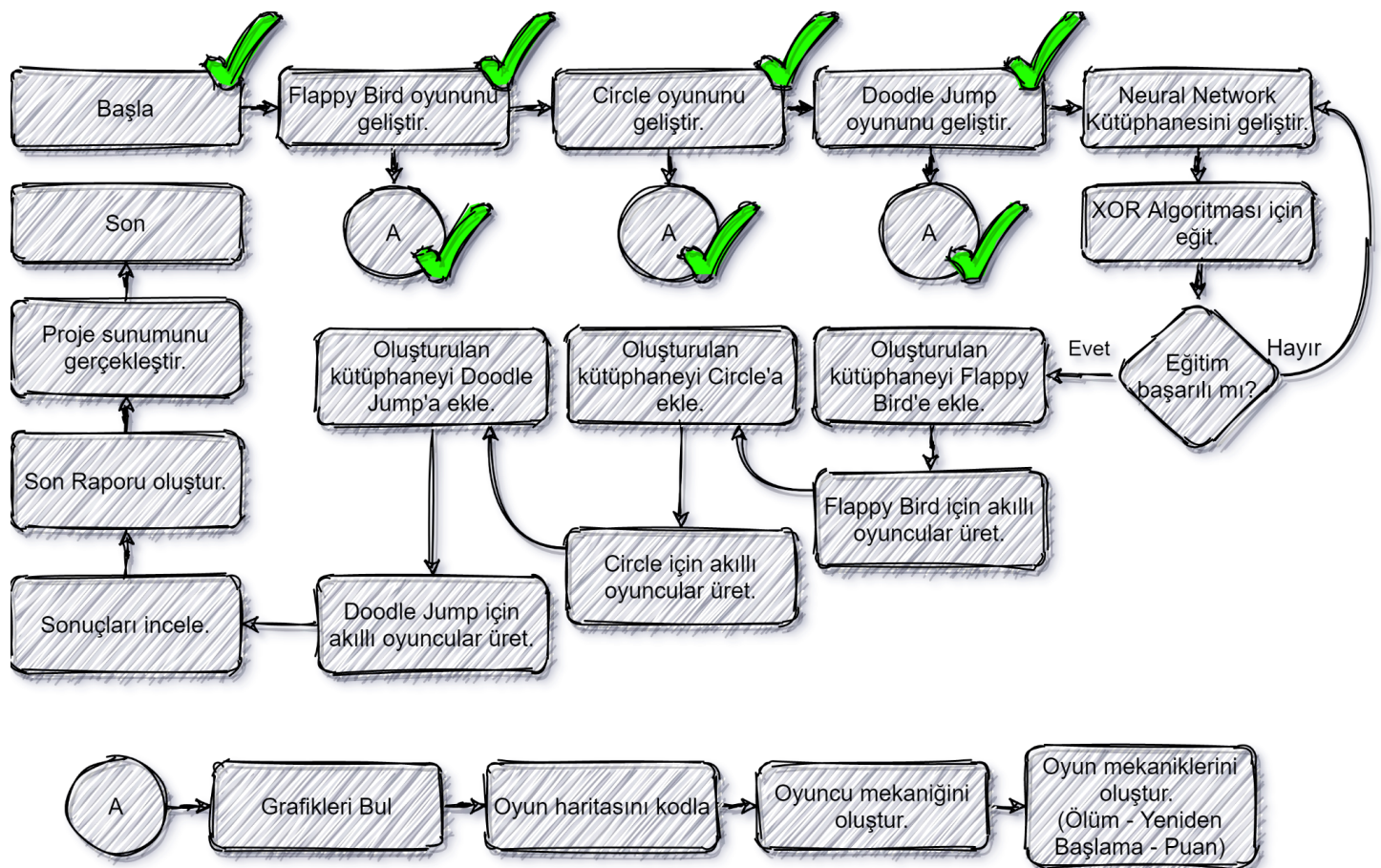
Bu raporda, çalışmada kullanılan iş akış şeması gösterilip projede bulunulan konum belirtilmiştir. Projede kullanılması planlanan yöntemler, bu yöntemlerin sözde kodu ve matematiksel formülleri yer almaktadır.

Proje İş Akış Şeması:

Proje boyunca aşağıdaki akış diyagramı ile hareket edilmiştir.

Gerçekleştirilen adımlar tik atılarak belirtilmiştir. Geliştirme süreci

projenin github reposu üzerinden takip edilebilir.



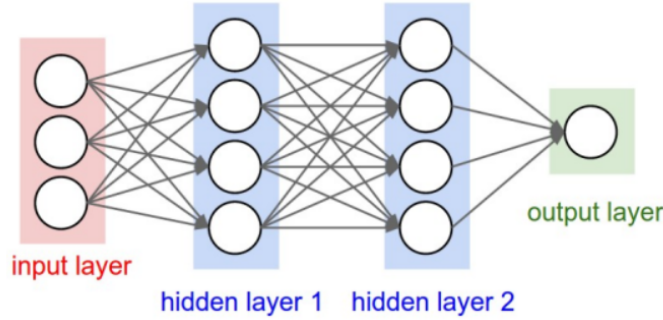
Yöntemler:

Bu bölümde çalışmada kullanılacak ve literatürde hali hazırda bulunan algoritmalar listelenmiş ve anlatılmıştır.

1. Neural Networks:

1.1. Mantığı:

Literatürde birçok makine öğrenmesi metodu mevcuttur. Nöral Ağlar da bunlardan biridir. İnsan beynindeki nöron-sinaps ilişkisinden esinlenilerek tasarlanmış ve çalıştırılmıştır. Genel olarak 3 bölümden oluşmaktadır. Bunlar; Girdi Katmanı - Gizli Katmanlar - Çıkış Katmanı



1.2. Matematiksel Formülü ve Sözde Kodu:

1. Initialize the input layer l_0
Set the values of the outputs o_i^0 for nodes in the input layer l_0 to their associated inputs in the vector $\vec{x} = \{x_1, \dots, x_n\}$, i.e. $o_i^0 = x_i$.
2. Calculate the product sums and outputs of each hidden layer in order from l_1 to l_{m-1}
For k from 1 to $m - 1$,
 - a. compute $h_i^k = \vec{w}_i^k \cdot \vec{o}^{k-1} + b_i^k = b_i^k + \sum_{j=1}^{r_{k-1}} w_{ji}^k o_j^{k-1}$ for $i = 1, \dots, r_k$;
 - b. compute $o_i^k = g(h_i^k)$ for $i = 1, \dots, r_k$.
3. Compute the output y for output layer l_m
 - a. Compute $h_1^m = \vec{w}_1^m \cdot \vec{o}^{m-1} + b_1^m = b_1^m + \sum_{j=1}^{r_{m-1}} w_{j1}^m o_j^{m-1}$.
 - b. Compute $o = o_1^m = g_o(h_1^m)$.

2. Genetic algorithm:

2.1. Mantığı:

Biyolojide bulunan doğal seçim kavramı örnek alınarak geliştirilmiştir. Genellikle rastgele oluşturulan bir grup birey ortamda test edilerek ne kadar başarılı oldukları gözlemlenmektedir. Bunların içinden ortama en çok uyum sağlayan bireyler ile yeni bir grup oluşturulur ve ne kadar uyum sağladıkları tekrar gözlenir. İstenilen görevi yerine getiren bir birey elde edilene kadar bu döngü tekrar edilir.

2.2. Sözde Kodu:

Given:

- nP: base population size.
- nI: number of iterations.
- rC: rate of crossover.
- rM: rate of mutation.

Generate initial population of size nP.

Evaluate initial population according to the fitness function.

While (*current_iteration* \leq nI)

 //Breed $rC \times nP$ new solutions.

 Select two parent solutions from current population.

 Form offspring's solutions via crossover.

IF(*rand*(0.0, 1.0) < rM)

 Mutate the offspring's solutions.

end IF

 Evaluate each child solution according to the fitness function.

 Add offspring's to population.

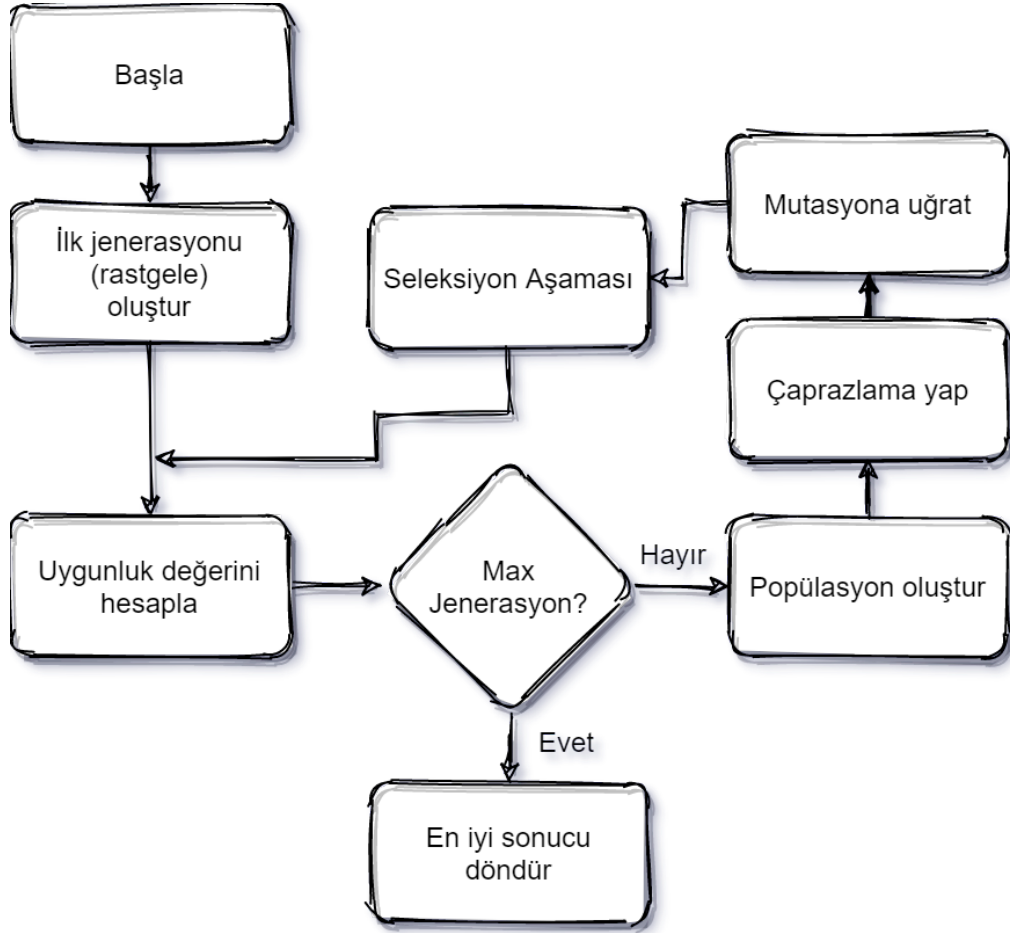
 //population size is now $MaxPop = nP \times (1 + rC)$.

 Remove the $rC \times nP$ least-fit solutions from population.

end While

Output the global best solution

2.3. Akış Şeması:



3. NEAT Algorithm:

3.1. Mantığı:

Bu algoritmayı kısaca özetlemek gerekirse; genetik algoritmasının nöral ağlar için özelleştirilmiş halidir. Sadece ağırlıkları optimize etmenin yanında topolojilerini ve nöron sayılarını da mutasyona uğratarak en optimize nöral ağı oluşturmak amaçlanmıştır.

3.2. Akış Şeması:

