



BURSA TEKNİK ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ (T) BÖLÜMÜ
BİLGİSAYAR OYUNLARDA YAPAY ZEKA

Ödev-8 Raporu

Github: <https://github.com/MoussaBane/BOYZ-ANN-Go-Kart>

MOUSSA BANE

24435004029

1. Yeni 3D Proje Oluşturma ve Temel Yapı:

İlk olarak, Unity'de yeni bir 3D proje oluşturun. Bu projede sahnede iki go-kart arabası yer alacak: birincisi oyuncu tarafından klavye ile kontrol edilecek, ikincisi ise Yapay Sinir Ağı (ANN) tarafından kontrol edilecek.

Adımlar:

- Yeni 3D proje oluşturun ve ANNRacing projesini içe aktarın.
- Starter sahnesini açın.
- **Sahne Elemanları:** İki go-kart aracı sahnede bulunmalı.
- **Araç Hareketi:**

Klavye ile Hareket: İlk araç (Kart) için klavye kontrolü eklenir. Bu, araç hareketini sağlamak için klavye tuşlarını kullanmanızı sağlar.

ANN ile Hareket: İkinci araç (ANNKart), ANN tarafından kontrol edilecek.

2. Script Oluşturma ve Araba Kontrolü

Yeni bir script oluşturun ve bu script'i ilk araca ekleyin. Bu script, oyuncu tarafından kontrol edilen aracın hareketini sağlayacak.

Adımlar:

- **Yeni Script:** Drive_sc adında bir script oluşturun ve bu script'i Kart'a ekleyin.
- **Kaydedin ve Çalıştırın:** Script'i kaydedin ve çalıştırarak aracın hareket ettiğinden emin olun.

3. Veri Toplama ve Mesafe Hesaplamaları

Yarış sırasında aracın hareket verilerini toplayarak ANN modeli için eğitim verisi oluşturacağız.

Adımlar:

- **Veri Toplama:** Aracın her hareketi sırasında translation ve rotation verilerini toplanır.
- **Mesafe Hesaplamaları:** Çeşitli yönlerde ışınlar göndererek çevredeki engellere olan mesafeyi ölçülür. Bu mesafeler, ANN'ye girdi olarak sağlanacaktır:

İleri Mesafe: Aracın önüne bir ışın gönderilir.

Sağ ve Sol Mesafe: Araç sağ ve sol yönlerde ışınlar göndererek mesafeyi ölçülür.

Çapraz Mesafeler: Sağ ve sol ileri 45 derecelik açılarda ışın gönderilir.

Script için Değişkenler:

- **visibleDistance:** Aracın etrafında ışın gönderme mesafesini belirleyecek değişken.

- **Mesafe Değişkenleri:** fDist, rDist, lDist, r45Dist, l45Dist gibi mesafe değişkenleri kullanarak çevredeki engellere olan mesafeyi hesaplanır.

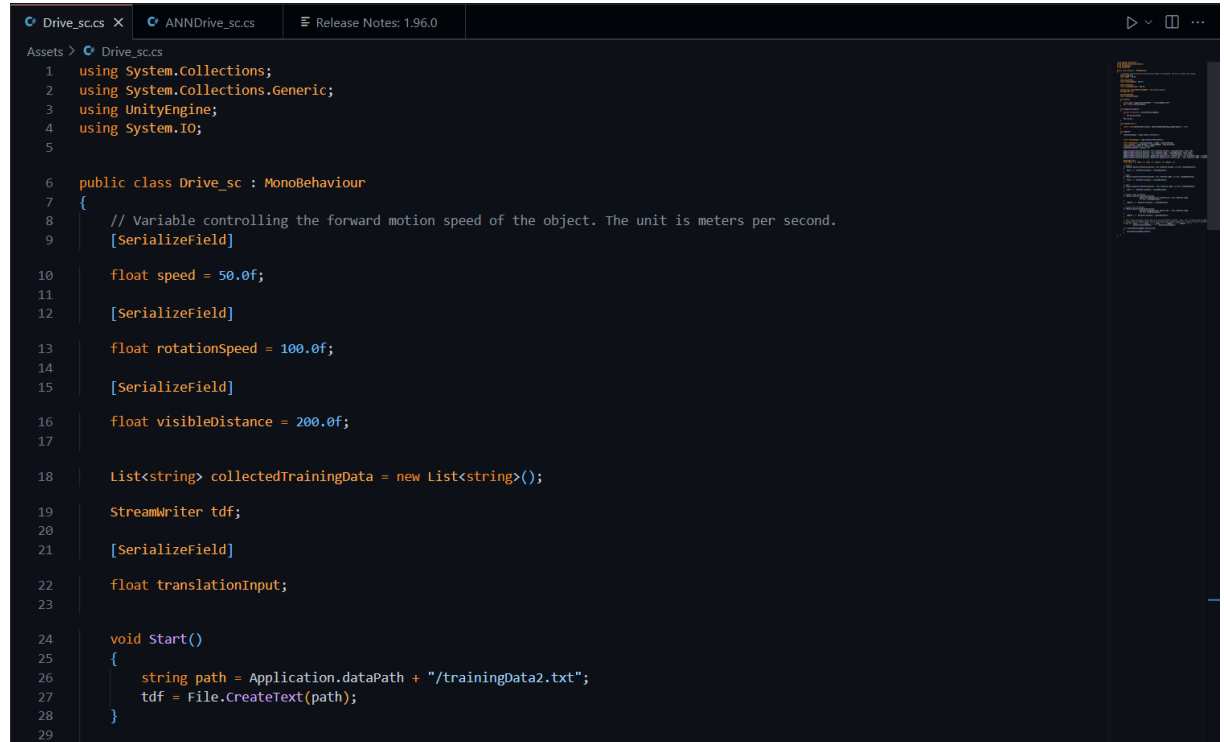
4. Veri Setinin Düzenlenmesi ve Normalizasyon

Toplanan veri seti ham olacak ve gürültü içerecektir. Bu gürültüyü azaltarak veriyi daha temiz hale getireceğiz.

Adımlar:

- **Gürültü Azaltma:** Toplanan veriyi yuvarlayarak yalnızca yeni verilerin listeye eklenmesini sağlar.
- **Normalizasyon:** Mesafeleri normalize ederek 0 ile 1 arasında bir değer elde edilir. Bu, verilerin daha uyumlu olmasını sağlar. Normalizasyon için mesafeyi visibleDistance ile bölünür.
- **Değer Tersine Çevirme:** Engellere yakınlaştıkça değerlerin artmasını sağlamak için mesafe değerlerini tersine çevirilir.

5. Drive_sc Script'in Son Hali



```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4 using System.IO;
5
6 public class Drive_sc : MonoBehaviour
7 {
8     // Variable controlling the forward motion speed of the object. The unit is meters per second.
9     [SerializeField]
10     float speed = 50.0f;
11
12     [SerializeField]
13     float rotationSpeed = 100.0f;
14
15     [SerializeField]
16     float visibleDistance = 200.0f;
17
18     List<string> collectedTrainingData = new List<string>();
19     StreamWriter tdf;
20
21     [SerializeField]
22     float translationInput;
23
24     void Start()
25     {
26         string path = Application.dataPath + "/trainingData2.txt";
27         tdf = File.CreateText(path);
28     }
29 }
```

```
Drive_sc.cs 1 x ANNDrive_sc.cs 7 Release Notes: 1.96.0
Assets > Drive_sc > ...
6 public class Drive_sc : MonoBehaviour
30 void OnApplicationQuit()
31 {
32     foreach (string td in collectedTrainingData)
33     {
34         tdf.WriteLine(td);
35     }
36     tdf.Close();
37 }
38
7 references
39 float Round(float x)
40 {
41     return (float)System.Math.Round(x, System.MidpointRounding.AwayFromZero) / 2.0f;
42 }
43
0 references
44 void Update()
45 {
46     translationInput = Input.GetAxis("Vertical");
47
48     float rotationInput = Input.GetAxis("Horizontal");
49
50     float translation = translationInput * speed * Time.deltaTime;
51     float rotation = rotationInput * rotationSpeed * Time.deltaTime;
52     transform.Translate(0, 0, translation);
53     transform.Rotate(0, rotation, 0);
54
55     Debug.DrawRay(transform.position, this.transform.forward * visibleDistance, Color.red);
56     Debug.DrawRay(transform.position, this.transform.right * visibleDistance, Color.blue);
57     Debug.DrawRay(transform.position, -this.transform.right * visibleDistance, Color.blue);
58     Debug.DrawRay(transform.position, Quaternion.AngleAxis(-45, Vector3.up) * this.transform.right * visibleDistance, Color.green);
59     Debug.DrawRay(transform.position, Quaternion.AngleAxis(45, Vector3.up) * -this.transform.right * visibleDistance, Color.green);
60
61     RaycastHit hit;
62     float fDist = 0, rDist = 0, lDist = 0, r45Dist = 0, l45Dist = 0;
63 }

Drive_sc.cs 1 x ANNDrive_sc.cs 7 Release Notes: 1.96.0
Assets > Drive_sc > ...
6 public class Drive_sc : MonoBehaviour
44 void Update()
65 // Forward
66 if (Physics.Raycast(transform.position, this.transform.forward, out hit, visibleDistance))
67 {
68     fDist = 1 - Round(hit.distance / visibleDistance);
69 }
70
71 // Right
72 if (Physics.Raycast(transform.position, this.transform.right, out hit, visibleDistance))
73 {
74     rDist = 1 - Round(hit.distance / visibleDistance);
75 }
76
77 // Left
78 if (Physics.Raycast(transform.position, -this.transform.right, out hit, visibleDistance))
79 {
80     lDist = 1 - Round(hit.distance / visibleDistance);
81 }
82
83 // Forward right 45 degrees
84 if (Physics.Raycast(transform.position,
85     Quaternion.AngleAxis(-45, Vector3.up) * this.transform.right,
86     out hit, visibleDistance))
87 {
88     r45Dist = 1 - Round(hit.distance / visibleDistance);
89 }
90
91 // Forward left 45 degrees
92 if (Physics.Raycast(transform.position,
93     Quaternion.AngleAxis(45, Vector3.up) * -this.transform.right,
94     out hit, visibleDistance))
95 {
96     l45Dist = 1 - Round(hit.distance / visibleDistance);
97 }
98

Drive_sc.cs 1 x ANNDrive_sc.cs 7 Release Notes: 1.96.0
Assets > Drive_sc > ...
6 public class Drive_sc : MonoBehaviour
44 void Update()
98
99 // This string represents sensor data from five directions (forward, right, left, forward-right 45 degrees, and forward-left
100 // along with the rounded values of the translation and rotation inputs. It is used to record training data for the model.
101 string td = fDist + "," + rDist + "," + lDist + "," + r45Dist + "," + l45Dist + "," +
102     Round(translationInput) + "," + Round(rotationInput);
103
104 if (!collectedTrainingData.Contains(td))
105 {
106     collectedTrainingData.Add(td);
107 }
108 }
109 }
110
```

6. ANN Modeli: Yapay Sinir Ağı

Artık toplanan verilerle bir ANN modeli oluşturacağız. Bu model, aracın hareketini kontrol etmek için eğitim alacak.

Adımlar:

- **Yeni Script:** ANNDrive_sc adında bir script oluşturun ve ANN kontrolündeki ANNKart'a ekleyin.
- **İleri ve Geri Yön Kontrolü:** translation ve rotation verilerini kullanarak ANN modelini eğitin. Bu veriler, aracın hızını ve yönünü belirleyecektir.

7. Adaptive Learning ve Ağırlık Yönetimi

ANN'nin başarısını iyileştirmek için, öğrenme süreci boyunca ağırlıkları güncelleyeceğiz.

Adımlar:

- **Epoch Başlangıcı:** Her epoch başında ağırlıkları kaydedin. Eğer hata iyileşmezse, önceki ağırlıkları kullanarak öğrenme oranını düşürün.
- **Ağırlık Kaydetme:** Eğitim sırasında ağırlıkları kaydedin ve dosyaya yazın.
- **Dosyadan Okuma:** Eğitim sonunda öğrenilen ağırlıkları dosyadan okuyarak devam edin.

8. Veri Setinin Kaydedilmesi ve İncelenmesi

Toplanan verileri kaydedip, gürültüyü azaltarak daha temiz bir veri seti elde edeceğiz.

Adımlar:

- **Veri Kaydetme:** Toplanan veriyi dosyaya kaydedin. System.IO kütüphanesini kullanarak CSV formatında veri kaydedebilirsiniz.
- **Veri İnceleme:** Veri setinde gereksiz verileri (noise) temizleyin ve sadece önemli verileri saklayın.

9. Sonuç ve İyileştirme

Oyun sonunda, ANN modelini kullanarak go-kart aracının daha doğru hareket etmesini sağlamak için modelin başarısını izleyin. Gerekirse, veriyi manuel olarak düzenleyerek iyileştirmeler yapın.

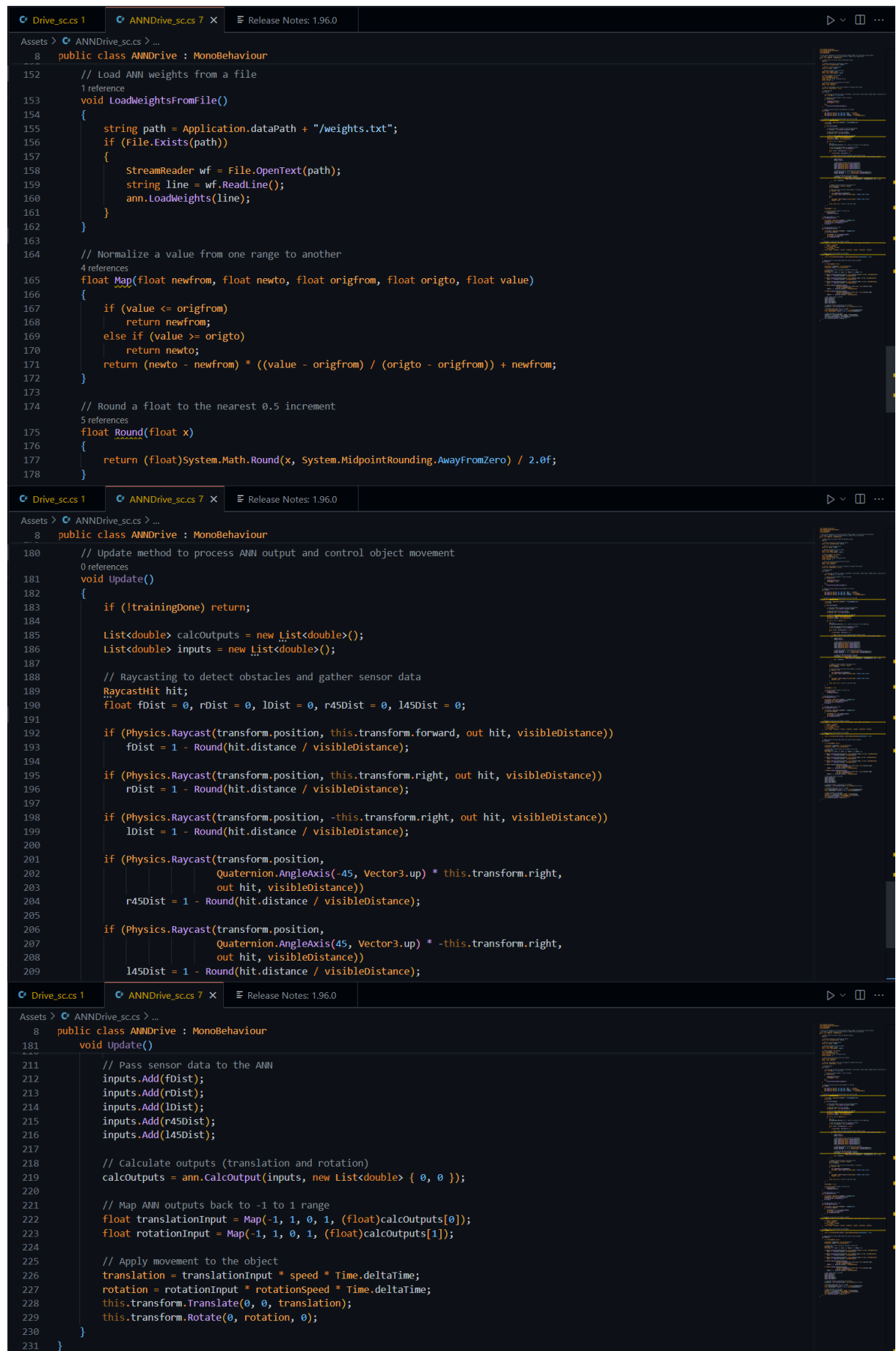
Adımlar:

- **Eğitim Sonrası Değerlendirme:** Modelin hata oranını değerlendirin ve gerektiğinde yeniden eğitim uygulayın.
- **Modeli İyileştirme:** Kullanıcı verisini Excel'de açarak çelişen verileri düzeltin ve modeli optimize edin.

10. ANNDrive_sc Script'in Son Hali

```
Assets > ANNDrive_sc.cs > ...
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4 using System.IO;
5
6 // This script implements an Artificial Neural Network (ANN) for controlling a Unity object.
7 // The ANN is trained using pre-recorded data or loaded from a saved state.
8 public class ANNDrive : MonoBehaviour
9 {
10     // ANN instance to process input and generate output
11     ANN ann;
12
13     // Visible distance for raycasting in meters
14     public float visibleDistance = 200.0f;
15
16     // Number of training iterations
17     public int epochs = 50000;
18
19     // Movement parameters for the object
20     public float speed = 50.0f;
21     public float rotationSpeed = 100.0f;
22
23     // State variables for training
24     bool trainingDone = false;
25     float trainingProgress = 0;
26     double sse = 0; // Sum of Squared Errors
27     double lastSSE = 1;
28
29     // Stores the current translation and rotation
30     public float translation;
31     public float rotation;
32
33     // Flag to load weights from a file instead of training from scratch
34     public bool loadFromFile = false;
35
36     // Initialization
37     void Start()
38     {
39         // Initialize the ANN with specific parameters: input nodes, output nodes, hidden layers, neurons per layer, and learning rate
40         ann = new ANN(5, 2, 1, 10, 0.5);
41
42         // Load pre-trained weights or start training
43         if (loadFromFile)
44         {
45             LoadWeightsFromFile();
46             trainingDone = true;
47         }
48         else
49             StartCoroutine(LoadTrainingSet());
50     }
51
52     // Display training progress and parameters on the GUI
53     void OnGUI()
54     {
55         GUI.Label(new Rect(25, 25, 250, 30), "SSE: " + lastSSE);
56         GUI.Label(new Rect(25, 40, 250, 30), "Alpha: " + ann.alpha);
57         GUI.Label(new Rect(25, 55, 250, 30), "Trained: " + trainingProgress);
58     }
59 }
```

```
Assets > ANNDrive_sc.cs > ...
8 public class ANNDrive : MonoBehaviour
60 // Coroutine to load the training dataset and train the ANN
61 IEnumerator LoadTrainingSet()
62 {
63     string path = Application.dataPath + "/trainingData2.txt";
64     string line;
65
66     if (File.Exists(path))
67     {
68         // Count the number of lines in the training dataset
69         int lineCount = File.ReadAllLines(path).Length;
70
71         // Open the dataset file for reading
72         StreamReader tdf = File.OpenText(path);
73
74         // Lists to store inputs and outputs for the ANN
75         List<double> calcOutputs = new List<double>();
76         List<double> inputs = new List<double>();
77         List<double> outputs = new List<double>();
78
79         for (int i = 0; i < epochs; i++)
80         {
81             sse = 0;
82             tdf.BaseStream.Position = 0; // Reset file pointer to the beginning
83
84             // Save the current weights for possible rollback
85             string currentWeights = ann.PrintWeights();
86
87             while ((line = tdf.ReadLine()) != null)
88             {
89                 string[] data = line.Split(';');
90
91                 // Ignore lines with zero translation and rotation values
92                 if (System.Convert.ToDouble(data[5]) != 0 && System.Convert.ToDouble(data[6]) != 0)
93                 {
94                     inputs.Clear();
95
96                     // Read inputs (sensor data from raycasting)
97                     inputs.Add(System.Convert.ToDouble(data[0]));
98                     inputs.Add(System.Convert.ToDouble(data[1]));
99                     inputs.Add(System.Convert.ToDouble(data[2]));
100                     inputs.Add(System.Convert.ToDouble(data[3]));
101                     inputs.Add(System.Convert.ToDouble(data[4]));
102
103                     // Normalize outputs (translation and rotation)
104                     outputs.Add(Map(0, 1, -1, 1, System.Convert.ToSingle(data[5])));
105                     outputs.Add(Map(0, 1, -1, 1, System.Convert.ToSingle(data[6])));
106
107                     // Train the ANN and calculate the error
108                     calcOutputs = ann.Train(inputs, outputs);
109                     float thisError = ((Mathf.Pow((float)(outputs[0] - calcOutputs[0]), 2) +
110                                     Mathf.Pow((float)(outputs[1] - calcOutputs[1]), 2))) / 2.0f;
111                     sse += thisError;
112                 }
113             }
114
115             // Update training progress and average error
116             trainingProgress = (float)i / epochs;
117             sse /= lineCount;
118
119             // Adjust learning rate and reload weights if necessary
120             if (lastSSE < sse)
121             {
122                 ann.LoadWeights(currentWeights);
123                 ann.alpha = Mathf.Clamp((float)ann.alpha - 0.001f, 0.01f, 0.9f);
124             }
125         }
126     }
127     else
128     {
129         ann.alpha = Mathf.Clamp((float)ann.alpha + 0.001f, 0.01f, 0.9f);
130         lastSSE = sse;
131     }
132     yield return null; // Wait for the next frame
133 }
134 }
135
136 trainingDone = true;
137
138 // Save the trained weights for future use
139 if (!loadFromFile)
140     SaveWeightsToFile();
141 }
142
143 // Save ANN weights to a file
144 void SaveWeightsToFile()
145 {
146     string path = Application.dataPath + "/weights.txt";
147     StreamWriter wf = File.CreateText(path);
148     wf.WriteLine(ann.PrintWeights());
149     wf.Close();
150 }
```



11. Sonuç

Yarış oyununda, oyuncu ve ANN kontrollü araçlar arasında veri toplanarak ANN modeli ile otonom sürüş gerçekleştirilmiştir. Modelin eğitim verisi, aracın çevresindeki engellere olan mesafeleri ve araç hareketini içermektedir. Elde edilen model, oyundaki engellere göre doğru hareketleri yapacak şekilde optimize edilmiştir.

12. Projenin Çalıştırılmış Hali

