



**İSTANBUL TEKNİK ÜNİVERSİTESİ**  
Elektrik Elektronik Fakültesi

**YAPAY SİNİR AĞLARI (EHB 420)**  
**2020-2021 Güz Öğretim Dönemi**  
**ÖDEV-1**

**Öğretim Üyesi: Prof. Dr. Neslihan Serap ŞENGÖR**  
**Dersin Yardımcısı: Araş. Gör. Halil Durmuş**

**Recep Salih YAZAR**  
**040170066**

**Ramazan Umut AKTAŞ**  
**040190762**

## GİRİŞ/AÇIKLAMALAR

İlk soruda veri kümemizi oluştururken doğrusal ayrıştırılabilir noktalar kümesini elde etmek için kendi karar düzlemimizi oluşturduk. 6 boyutlu uzaydaki 5 boyutlu bir hiper-düzlemin ifadesi aşağıdaki gibidir:

- $\underline{X} = [x^1 \ x^2 \ x^3 \ x^4 \ x^5]$ , koordinat vektörü
- $\underline{V} = [a \ b \ c \ d \ e]$ , en az bir elemanı sıfırdan farklı ve tüm elemanları reel sayılardan oluşan bir vektör
- $\langle x, u \rangle$ , öklidyen iç çarpım
- $F$ , bir reel sayı

olmak üzere;

$\langle \underline{X}, \underline{V} \rangle = F$  denklemini sağlayan tüm  $x^i \in \mathbb{R}^5$ ,  $i=1,2,3,4,5$  noktalar kümesine hiper-düzlem denir.  $F=0$  olduğu durumlarda, hiper-düzlem  $\mathbb{R}^6$  uzayını iki alt uzaya ayırır, hiper-düzlemin üzerinde kalan noktalar pozitif, altında kalan noktalar negatif yarı uzayın elemanlarıdır. Verilerimizi oluştururken noktalarımızın 20 tanesini belirlediğimiz karar düzleminin oluşturduğu pozitif yarı uzaydan, 20 tanesini de negatif yarı uzaydan seçtik. Pozitif yarı uzayda kalan noktaları 1 değerli sınıf ile, negatif yarı uzayda kalan noktaları -1 değerli sınıf ile belirttik. Hiper-düzlemimizi belirleyen  $\underline{V}$  ve  $F$  aşağıdaki gibidir:

$$\underline{V} = [-1 \ 3 \ 4 \ -2 \ 5]$$

$$F = 0$$

Eğitim kümemizi seçerken noktaları karar düzlemine uzaklığı yakın olanlardan seçtik. Bu koşulu sağlamak için  $\langle \underline{X}, \underline{V} \rangle$  değeri -10 ile 10 arasında olacak şekilde eğitim kümesi vektörleri oluşturuldu. Karar düzleminden uzak, ayrık noktaları seçmiş olsaydık, genlikte ayrık algılayıcının belirleyeceği karar düzlemi bu düzleme yakın olan noktaları ayıramayabilirdi. Eğitim ve test kümesi için iki sınıftan da yakın sayıda veri aldık. Eğitim kümesi için 13 tane 1 numaralı sınıftan 12 tane -1 numaralı sınıftan vektör aldık. Test kümesi için ise 7 tane 1 numaralı sınıftan 8 tane -1 numaralı sınıftan vektör kullandık. Eğitim ve test kümesinde ortak eleman yoktur.

Ağırlıkların ilk koşullarını tartışmak için ağırlık vektörünü ( $\underline{W}$ ) karar düzlemimizi belirleyen vektöre ( $\underline{V}$ ) benzer olan vektörler ve farklı olan vektörlerden seçerek karar düzleminin belirlenmesi için gereken iterasyon sayısını saydırdık. Benzerlik ölçütü olarak Öklid normunu kullandık.  $\|\underline{V} - \underline{W}\|_E$  normunun küçüldükçe benzerliğin artacağını kabul ettik. Aynı şekilde farklı şekilde sıralanan eğitim kümesi elemanları için de iterasyon sayısını saydırdık. Ayrıca farklı durumlarda elde edilen karar düzlemlerinin test kümesindeki kaç elemanı doğru olarak sınıflandırdığını rapora ekledik.

İlk sorunun b alt maddesinde veri kümesini oluşturmak için rastgele vektörler ürettik.

İkinci soruda veri setinin doğrusal ayrıştırılabilir olup olmadığını test etmek için veri noktalarını ve konveks gövdelerini çizdirdik. İki kümenin doğrusal ayrıştırılabilir olması için konveks gövdelerinin kesişmemesi gerekir. <sup>[1], [2]</sup>

Konveks gövde çizdirmek için Graham Scan Algoritması kullanıldı. <sup>[3], [4]</sup>

Ödev boyunca bias terimi her zaman vektörün son elemanı olarak alınmıştır.

3.Soruda eğitim ve test kümemizi  $x_1$  için  $[0, 1]$  aralığını,  $x_2$  için  $[0, \frac{\pi}{2}]$  aralığını 50 eşit parçaya bölerek ve bu parçaların kombinasyonlarını alarak oluşturduk Toplam veri kümemiz 2500 noktadan oluşmuştur. Verilen fonksiyonu daha iyi modelleyebilmek için eğitim kümesi elemanlarını homojen şekilde seçmeye özen gösterdik.

4.Soruda verilen fonksiyonu ADALINE ile ifade edebilmek için ağ yapısının giriş katmanının elemanlarına ek olarak  $x_1$ ,  $x_2$ ,  $x_1^2$ ,  $x_2^2$ ,  $x_1x_2$  olarak güncelledik. Böylece verilen fonksiyon ifadesindeki tüm olası  $x_i$ ,  $x_j$  kombinasyonları genlikte sürekli algılayıcıya girdi olarak verildi. Bu girdilerin katsayıları  $c_1$ ,  $c_2$ ,  $c_{11}$ ,  $c_{22}$ ,  $c$  ise ADALINE tarafından bulunacaktır.

Birinci soru için oluşturduğumuz veri kümesi, eğitim kümesi ve test kümesi aşağıdaki gibidir. Bu kümeyi oluşturmak için kullandığımız kod .py dosyası olarak eklenmiştir.

### Sınıf: 1

[-4 -3 3 -3 1],[-7 8 4 -2 9],[7 3 1 -8 1],[-8 1 2 2 6],[-9 4 9 6 8],[3 9 5 -5-9],  
[-6 -4 5 -4 2],[9 -4 4 -6 5],[-6 -1 9 -6 4],[-5 4 -10 -5 3],[-9 0 3 7 0],  
[-2 5 5 5 -5],[-7 -8 2 7 6],[-4 -6 -3 -5 5],[8 -5 6 -5 -1],[-1 -2 5 8 2],  
[2 -5 6 8 3],[-8 -2 3 -4 -3],[-8 6 -8 6 4],[9 8 1 -6 -5]

### Sınıf: -1

[-2 -2 6 3 -10],[7 0 -5 -3 -8],[9 -1 -6 -3 2],[2 -1 -6 3 -5],[9 0 -3 -5 -5],  
[-6 -3 4 8 -2],[-10 6 -5 -6 -4],[-6 1 -9 7 4],[-6 -10 0 0 2],[8 9 -3 7 1],  
[2 -1 8 -7 -10],[-7 -4 -4 5 6],[-3 4 -9 8 7],[5 5 0 -9 -6],[-7 -5 2 6 1],  
[-4 -1 -1 1 1],[-9 9 -9 -7 -4],[-7 2 -7 -10 -2],[-4 -1 -1 -2 -2],[-1 9 -1 -10 -10]

### Eğitim Kümesi:

[-7 8 4 -2 9],[3 9 5 -5-9],[-5 4 -10 -5 3],[-9 0 3 7 0],[-2 5 5 5 -5],[-7 -8 2 7 6],  
[-4 -6 -3 -5 5],[8 -5 6 -5 -1],[-1 -2 5 8 2],[2 -5 6 8 3],[-8 -2 3 -4 -3],  
[-8 6 -8 6 4],[9 8 1 -6 -5],[-10 6 -5 -6 -4],[8 9 -3 7 1],[2 -1 8 -7 -10],  
[-7 -4 -4 5 6],[-3 4 -9 8 7],[5 5 0 -9 -6],[-7 -5 2 6 1],[-4 -1 -1 1 1],  
[-9 9 -9 -7 -4],[-7 2 -7 -10 -2],[-4 -1 -1 -2 -2],[-1 9 -1 -10 -10]

### Test Kümesi:

[-4 -3 3 -3 1],[7 3 1 -8 1],[-8 1 2 2 6],[-9 4 9 6 8],[-6 -4 5 -4 2],[9 -4 4 -6 5],  
[-6 -1 9 -6 4],[-2 -2 6 3 -10],[7 0 -5 -3 -8],[9 -1 -6 -3 2],[2 -1 -6 3 -5],  
[9 0 -3 -5 -5],[-6 -3 4 8 -2],[-6 1 -9 7 4],[-6 -10 0 0 2]

## 1.Soru

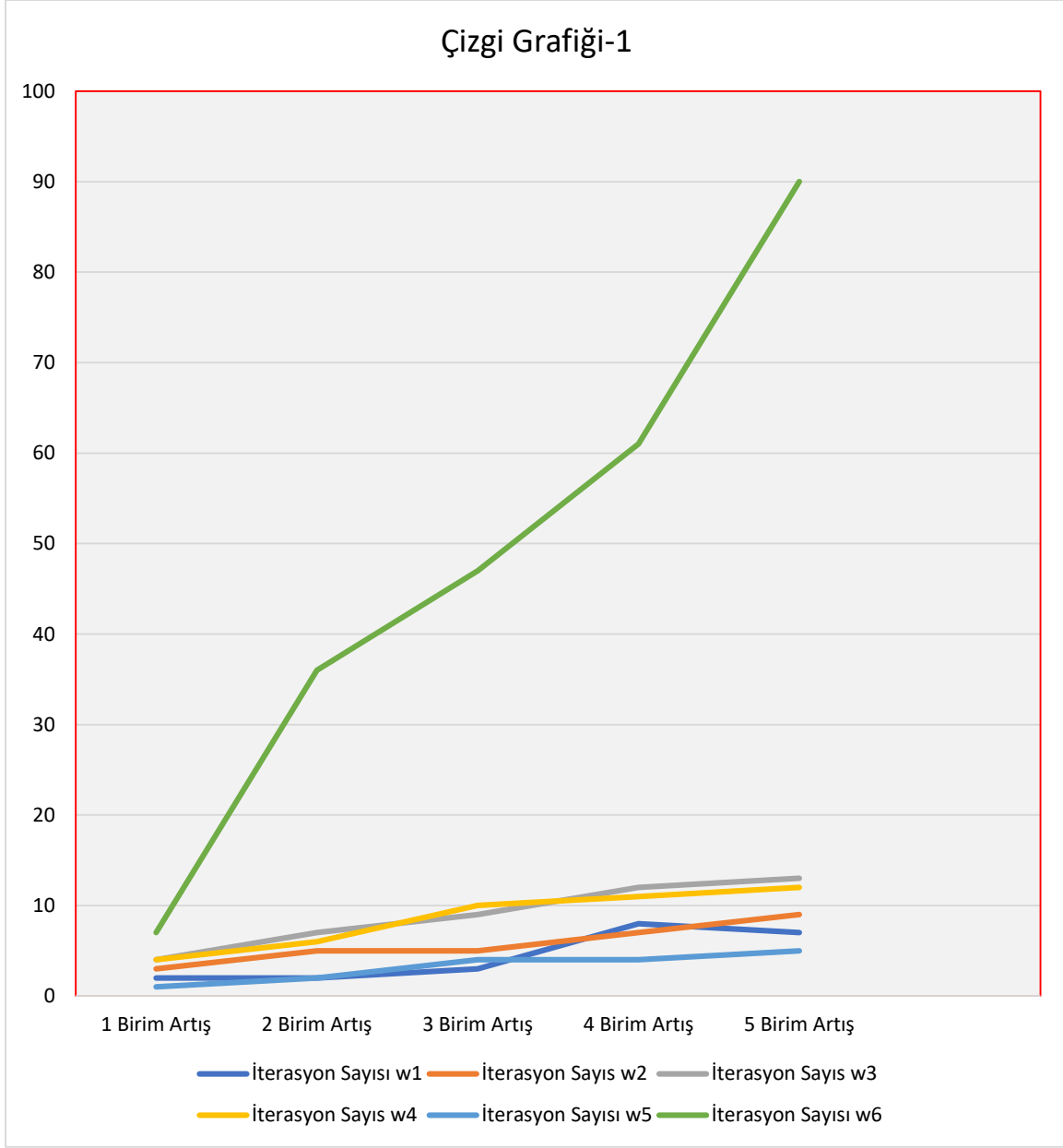
- Ağırlıkların ilk koşullarının eğitim sürecine etkisi

Veri kümesini belirlemek için oluşturduğumuz düzlemi belirleyen  $\underline{V} = [-1 \ 3 \ 4 \ -2 \ 5 \ 0]$  vektörüne uzaklıklarına göre sıralanan başlangıç ağırlıkları ile, bu ağırlıklar ile karar düzlemi belirlenene kadar geçen iterasyon sayısı ve bulunan karar düzleminin test kümesindeki kaç veriyi doğru sınıflandırdığını gösteren tablo (Tablo-1) aşağıdadır.  $\underline{V}$  vektörünün son elemanı olan 0, karar düzlemi orijinden geçtiği için sıfır alınmıştır. Eğer düzlem orijinden geçmek yerine başka bir noktadan geçseydi son eleman, iç çarpımın ( $\langle \underline{X}, \underline{V} \rangle$ ) sonucu olan skalerin negatifi (-F) olurdu. Bu süreçte öğrenme hızı 0.01 alınmıştır ve eğitim kümesinin sıralanışı sabit tutulmuştur.

T1	BAŞLANGIÇ AĞIRLIK VEKTÖRÜ	$\underline{V}$ VEKTÖRÜNE MESAFE	İTERASYON SAYISI	DOĞRU SINIFLANDIRILAN VERİ SAYISI
1	[0 3 4 -2 5 0]	1	2	15
2	[-1 4 4 -2 5 0]	1	3	15
3	[-1 3 4 -2 5 0]	1	4	15
4	[-1 3 4 -1 5 0]	1	4	15
5	[-1 3 4 -2 6 0]	1	1	15
6	[-1 3 4 -2 5 1]	1	7	15
7	[1 3 4 -2 5 0]	2	2	15
8	[-1 5 4 -2 5 0]	2	5	15
9	[-1 3 6 -2 5 0]	2	7	15
10	[-1 3 4 0 5 0]	2	6	15
11	[-1 3 4 -2 7 0]	2	2	15
12	[-1 3 4 -2 5 2]	2	36	15
13	[2 3 4 -2 5 0]	3	3	15
14	[-1 6 4 -2 5 0]	3	5	15
15	[-1 3 7 -2 5 0]	3	9	15
16	[-1 3 4 1 5 0]	3	10	15
17	[-1 3 4 -2 8 0]	3	4	15
18	[-1 3 4 -2 5 3]	3	47	15

19	[3 3 4 -2 5 0]	4	8	15
20	[-1 7 4 -2 5 0]	4	7	15
21	[-1 3 8 -2 5 0]	4	12	15
22	[-1 3 4 2 5 0]	4	11	15
23	[-1 3 4 -2 9 0]	4	4	15
24	[-1 3 4 -2 5 4]	4	61	15
25	[4 3 4 -2 5 0]	5	7	15
26	[-1 8 4 -2 5 0]	5	9	15
27	[-1 3 9 -2 5 0]	5	13	15
28	[-1 3 4 3 5 0]	5	12	15
29	[-1 3 4 -2 10 0]	5	5	15
30	[-1 3 4 -2 5 5]	5	90	15
31	[1 1 1 1 1 0]	$\sqrt{42}$	51	15
32	[0 0 0 0 0 0]	$\sqrt{55}$	35	15
33	[-1 -1 -1 -1 -1 0]	$\sqrt{78}$	118	15
34	[-1 3 4 -2 55 0]	50	38	15

Tablo-1 ilk 30 satırda belirlenen ağırlık vektörleri sadece düzlemi belirleyen vektörün 1 elemanı değiştirilerek elde edilmiştir. Eğitim kümesindeki vektörlerin son elemanı olan bias terimi ile çarpılan ağırlık vektörünün son elemanındaki artış yapılan iterasyon sayısındaki artışı ağırlık vektörünün diğer terimlerindeki değişime göre daha çok etkilemiştir. Tablo-1’de görüldüğü üzere ağırlık vektörünün son terimi 1 arttığında 7 iterasyon, 2 arttığında 36 iterasyon, 3 arttığında ise 47 iterasyon yapılmıştır. Buna karşılık diğer terimlerde yapılan artışlar 3 birime kadar maksimum 10 iterasyon gerektirmiştir.



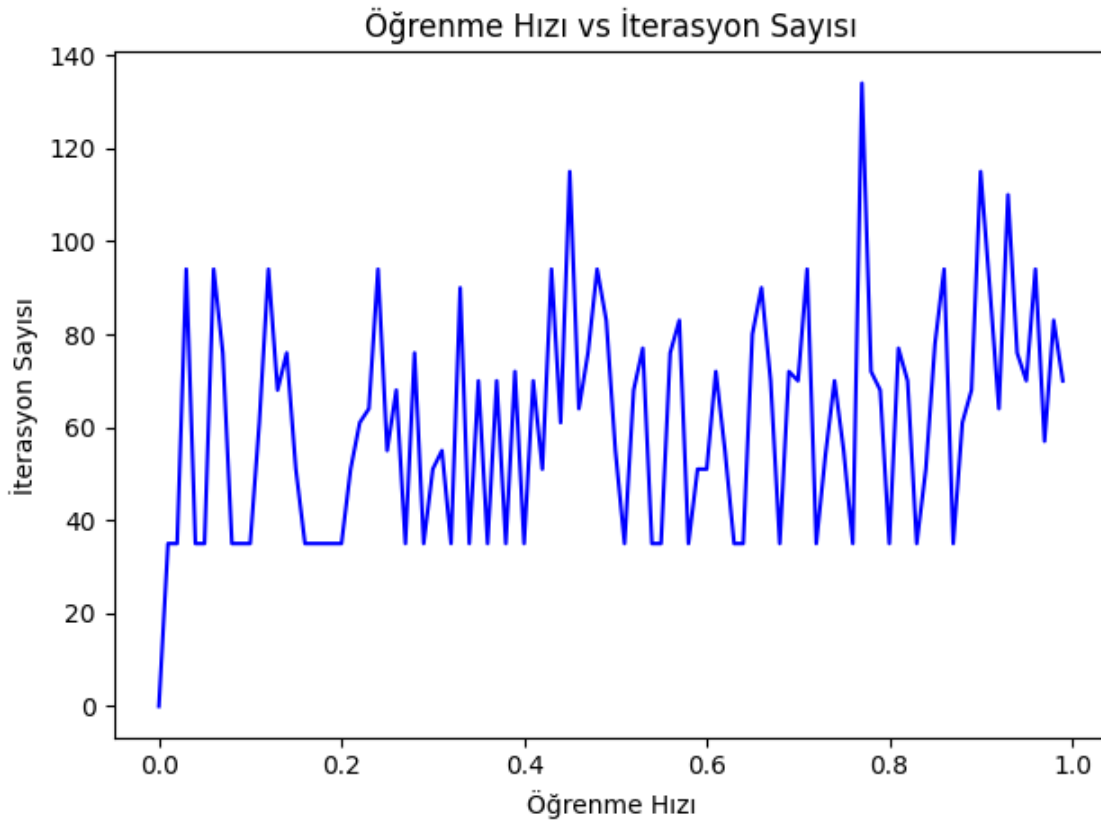
Çizgi grafiği-1’de düzlemi belirleyen vektörünün seçilen elemanına kendisi hariç diğer elemanlar sabit tutulup, seçilen elemanın her adımda bir birim artırılması ile elde edilen ağırlık vektörü ile elde edilen iterasyon sayısı temsil edilmektedir. Tablo-1’deki ilk 30 veriyi temsil etmektedir. Grafikten de görüldüğü üzere yapılan iterasyon sayısının ağırlık vektörünün son terimine bağlılığı daha yüksektir. Ayrıca ağırlık vektörünün düzlemi belirleyen vektöre olan uzaklığının artması ile iterasyon sayısı her zaman artmamıştır. Örneğin ağırlık vektörünün ilk elemanındaki 4 birimlik artışta 8 iterasyon yapılırken 5 birimlik artışta 7

iterasyon yapılmıştır. Tablo-1'deki 31,32,33,34 numaralı satırlardaki verinin de desteklediği üzere iterasyon sayısı iki vektörün arasındaki mesafe ile doğru orantılı olmak zorunda değildir. Ayrıca görüldüğü üzere iterasyon sayısındaki değişim ağırlık vektörünün elemanlarının değişiminden eşit olarak etkilenmemektedir. Elemanların iterasyon sayısını farklı hassasiyetlere etkilemektedirler. Son olarak, test kümemiz ağırlık vektörünün seçiminden bağımsız olarak her seferinde %100 doğrulukla sınıflandırılmıştır. Bu durum eğitim kümesindeki elemanlar doğru sınıflandırıldığı takdirde eğitim kümemizin de doğru sınıflandırıldığını destekler niteliktedir.



- **Öğrenme hızının eğitim sürecine etkisi**

Başlangıç ağırlık vektörü ve eğitim kümesi elemanlarının sıralanışı değiştirilmeden, sadece eğitim hızı değiştirilerek karar düzleminin belirlenmesi için gereken iterasyon sayısının grafiği aşağıdadır. Test kümesindeki elemanlar tüm öğrenme hızlarında %100 doğru sınıflandırılmıştır. Farklı öğrenme hızlarında kaç iterasyon yapıldığını görmek için 0 ile 1 arası 100 parçaya bölünmüş ve 0.01'den itibaren artış gösteren öğrenme hızlarına karşılık gelen iterasyon sayısı çizdirilmiştir.



Grafikte görüldüğü üzere öğrenme hızı ile iterasyon sayısı arasında dalgalı bir ilişki vardır. Belirli değerlerde yapılan iterasyon minimum değere ulaşmaktadır, salınımların periyodu eşit değildir. Öğrenme hızı değiştikçe bulunan karar düzlemleri değişmektedir ancak test kümesi her zaman doğru sınıflandırılmaktadır. Bu test kümesi seçiminin eğitim kümesi doğru sınıflandıkça test kümesinin de doğru sınıflanacak şekilde yapıldığını desteklemektedir.

- Eğitim kümesi elemanlarının farklı sıralanışının eğitim sürecine etkisi

Öğrenme hızı 0.01 ve başlangıç ağırlığı [0 0 0 0 0 0] alınarak yapılan hesapta eğitim kümesinin farklı sıralandığı 25 durumda yapılan iterasyon sayısı ve doğru sınıflandırılan veri sayısı aşağıdaki tabloda (Tablo-2) listelenmiştir. Bu başlangıç koşulları ile eğitim kümesinin sıralanışı değiştirilmeden karar düzleminin bulunması 35 iterasyon sürmüştür.

<b>TABLO-2</b>	<b>İTERASYON SAYISI</b>	<b>DOĞRU SINIFLANDIRILAN VERİ SAYISI</b>
1	47	15
2	105	15
3	88	15
4	14	15
5	64	15
6	54	15
7	75	15
8	104	15
9	55	15
10	48	15
11	80	15
12	75	15
13	84	15
14	32	15
15	83	15
16	26	15
17	73	15
18	89	15
19	27	15
20	56	15
21	61	15
22	100	15
23	40	15
24	41	15
25	33	15

Tablo-2' de görüldüğü üzere eğitim kümesinin sıralanışından bağımsız olarak test kümesinin tüm elemanları doğru olarak sınıflandırılmıştır. Bu durum test kümesinin eğitim kümesi doğru sınıflandırıldığı takdirde doğru sınıflanacak şekilde seçildiğini desteklemektedir.

Eğitim kümesinin farklı sıralanışı ile veriler ortalama 62 iterasyonda sınıflandırılmıştır. Sıralı eğitim kümesine göre 27 iterasyon artış yaşanmıştır. İterasyon sayısına ait verilerin standart sapması 26,0091 olarak hesaplanmıştır.

- **Lineer Ayırıştırılamayan Grup**

Doğrusal ayırıştırılamayan bir veri kümesi oluşturmak için 5 boyutlu vektörlerimizin elemanlarını ilk önce iki gruba ayırdık. Bir sınıfa ait vektörlerin tümünün elemanları çift sayılardan diğer sınıfa ait olanların ise tek sayılardan oluşuyor. Ayrıca seçilen herhangi bir vektörün 5 elemanının hepsi aynı. Son olarak vektörler oluşturulurken girdileri ardışık sıralanacak şekilde oluşturuldu.

Sınıf-1:  $[2i \ 2i \ 2i \ 2i \ 2i]$ ,  $i=0,1,2, 3, \dots, 10$

Sınıf-2:  $[2i+1 \ 2i+1 \ 2i+1 \ 2i+1 \ 2i+1]$ ,  $i=0,1,2, 3, \dots, 10$

Bu sayede vektörlerin elemanları 5 koordinat ekseninde de ayırıştırılamayan veri kümelerinden oluşacak şekilde düzenlenmiş oldu.

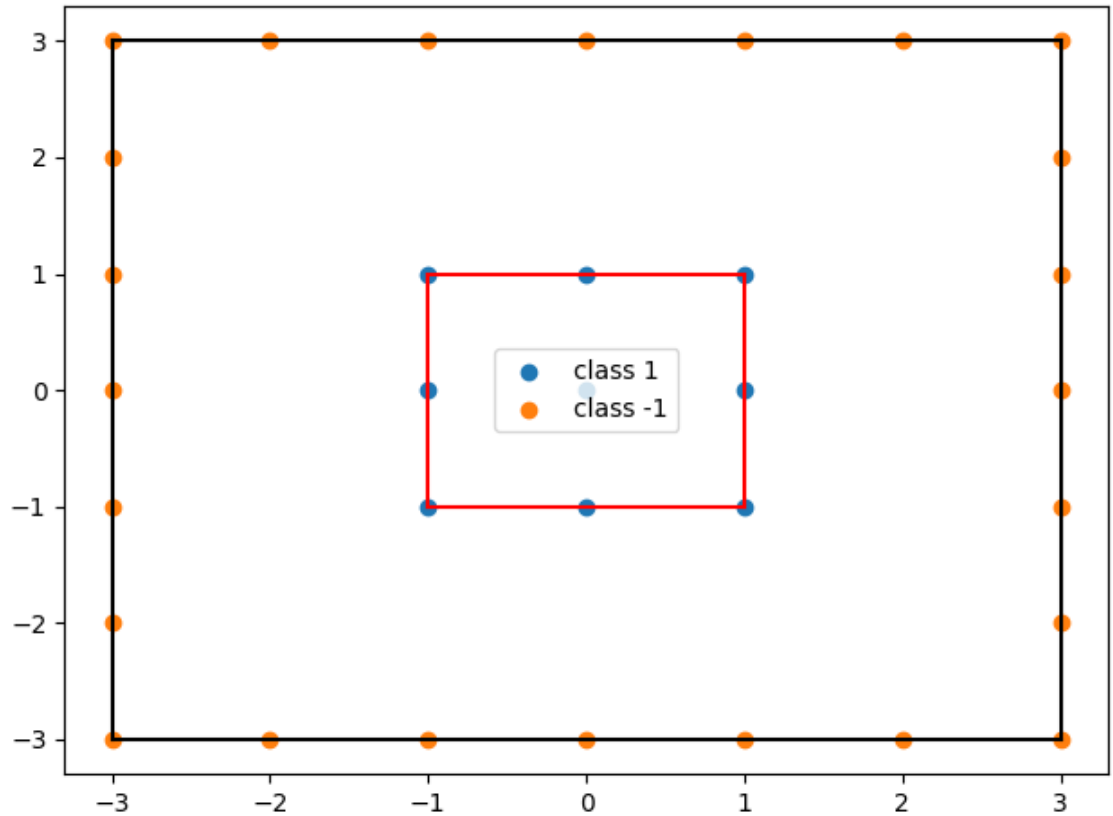
Genlikte ayırık algılayıcı veri kümesinin farklı sıralandığı, başlangıç ağırlığının ve öğrenme hızının değiştirildiği durumlarda 500, 1500 ve 5000 iterasyon sonucunda bir karar düzlemi bulamadı. Bu durum verilerin doğrusal ayırıştırılmaz olduğunu desteklemektedir, çünkü perceptron yakınsama teoremi gereği, doğrusal ayırıştırılabilen veriler için sonlu sayıda adımda verilen veri kümesini doğrusal ayırıştırarak bir karar düzlemi bulunmalıdır.

## 2.Soru

$$\left\{ \begin{pmatrix} 0 \\ -1 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ -1 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \begin{pmatrix} -1 \\ -1 \end{pmatrix}, \begin{pmatrix} -1 \\ 0 \end{pmatrix}, \begin{pmatrix} -1 \\ 1 \end{pmatrix}, \right. \\ \left. \begin{pmatrix} -3 \\ 3 \end{pmatrix}, \begin{pmatrix} -3 \\ 1 \end{pmatrix}, \begin{pmatrix} -3 \\ 0 \end{pmatrix}, \begin{pmatrix} -3 \\ -1 \end{pmatrix}, \begin{pmatrix} -3 \\ -3 \end{pmatrix}, \begin{pmatrix} -1 \\ 3 \end{pmatrix}, \begin{pmatrix} -1 \\ -3 \end{pmatrix}, \right. \\ \left. \begin{pmatrix} 0 \\ 3 \end{pmatrix}, \begin{pmatrix} 0 \\ -3 \end{pmatrix}, \begin{pmatrix} 1 \\ 3 \end{pmatrix}, \begin{pmatrix} 1 \\ -3 \end{pmatrix}, \begin{pmatrix} 3 \\ 3 \end{pmatrix}, \begin{pmatrix} 3 \\ 1 \end{pmatrix}, \begin{pmatrix} 3 \\ 0 \end{pmatrix}, \begin{pmatrix} 3 \\ -1 \end{pmatrix}, \right. \\ \left. \begin{pmatrix} 3 \\ -3 \end{pmatrix}, \begin{pmatrix} -2 \\ 3 \end{pmatrix}, \begin{pmatrix} -3 \\ 2 \end{pmatrix}, \begin{pmatrix} -3 \\ -2 \end{pmatrix}, \begin{pmatrix} -2 \\ -3 \end{pmatrix}, \begin{pmatrix} 2 \\ 3 \end{pmatrix}, \begin{pmatrix} 3 \\ 2 \end{pmatrix}, \begin{pmatrix} 3 \\ -2 \end{pmatrix}, \begin{pmatrix} 2 \\ -3 \end{pmatrix} \right\}$$

Yukarıdaki soruda verilen iki farklı sınıftan oluşan veri kümesi gösterilmiştir.

- **Veri kümesi lineer ayrıştırılabilir midir?** Aşağıdaki resimde iki sınıfa ait veri kümesi ve konveks gövdeleri çizdirilmiştir.



Görüldüğü gibi -1 numaralı sınıfın konveks gövdesinin içinde 1 numaralı sınıfın konveks gövdesi yer almaktadır. İki farklı veri setinin konveks gövdeleri kesişmektedir (geometrik olarak değil küme olarak). İki farklı veri kümesinin doğrusal ayrıştırılabilir olması için konveks gövdeleri kesişmemelidir, bu veri kümesi için bu şart sağlanmadığı için bu iki kümeyi lineer olarak ayrıştırıramayız.

- **Kümeleri doğrusal ayrıştırılabilir hale getirmek için yöntem önerisi**

Verilen kümeleri ara katmanında 3 nöron bulunan genlikte ayrık algılayıcı ile sınıflandırabiliriz. Ara katmandaki 3 nöron örüntünün boyutunu arttıracaktır. Örüntüye ilişkin vektörlerin koordinatları sırasıyla  $x_1$ ,  $x_2$  olsun. Ara katmandaki ilk 2 nöronun girdileri sırasıyla  $x_1$  ve  $x_2$  olacaktır. Bu nöronların yaptığı bir işlem yoktur, sadece girdi bilgisini depolarlar. 3. nöron ise  $x_1$  ve  $x_2$  girdilerini kullanarak bir  $x_3$  çıktısı üretecektir. Bu 3. Çıktı örüntünün boyutunu arttıran çıktıdır. Bu çıktı -1 numaralı sınıfa ait girdiler için pozitif bir değer, 1 numaralı sınıfa ait girdiler için ise negatif bir değer verecektir. Böylelikle 2 boyuttan 3 boyuta taşınan veri kümemiz  $x_3$  eksenine göre pozitif tarafta ve negatif tarafta kalacak şekilde 2 farklı sınıfa ayrılacak şekilde sınıflanmış olacaktır.  $X_3$  ekseninin kendisi bu boyutu arttırılmış örüntüyü doğrusal olarak ayrıştıran düzlemlerden birisi olacağı için bu örüntüyü doğrusal olarak ayrıştıran en az bir düzlem olacağından emin olabiliriz.

$x_3 = (x_1)^4 + (x_2)^4 - 55$  işlemini yerine getiren 3. nöron 1 numaralı sınıfa ait verilerin tümü için negatif bir değer, -1 numaralı sınıfa ait tüm veriler için ise pozitif bir değer üretir. Bu durumda genlikte ayrık algılayıcının ara katmanının çıkışı şu şekilde ifade edilebilir:

$$f(x_1, x_2) = [(x_1), (x_2), ((x_1)^4 + (x_2)^4 - 50)]$$

Bu kısmı ara katman ile gerçekledikten sonra eğitim ve test kümemizi seçerken  $z=0$  düzlemine yakın olan noktaları eğitim kümesine dahil ettik. Çünkü  $z=0$ 'a göre 2 sınıfa ayrılan noktalardan bu düzleme uzak olanları sınıflandıran bir karar düzlemi yakın olanları sınıflandıramayabilir.  $Z=0$ 'a yakın olan noktaları sınıflandıran düzlemin uzak noktaları sınıflandırması daha olasıdır.

Verilen 33 tane veriden 18 tanesi eğitim kümesine 15 tanesi ise test kümesine dahil edilmiştir. Kümeler aşağıda sırasıyla yazılmıştır.

### Eğitim Kümesi:

[0 -1 -49],[0 0 -50],[0 1 -49],[-1 0 -49],[1 0 -49],[-3 0 31],[3 0 31],[0 3 31],  
[0 -3 31],[-3 -1 32],[-1 3 32],[-1 -3 32],[-3 1 32],[1 3 32],[1 -3 32],[3 -1 32],  
[3 1 32]

### Test Kümesi:

[1 -1 -48],[1 1 -48],[-1 -1 -48],[-1 1 48],[-3 -3 112],[-3 3 112],[3 3 112],  
[3 -3 112],[-2 3 47],[-3 2 47],[-3 -2 47],[-2 -3 47],[2 3 47],[3 2 47],[3 -2 47],  
[2 -3 47]

Aşağıdaki tabloda (Tablo-3) farklı başlangıç ağırlık koşulları ile doğrusal ayrıştırılabilir hale getirilen noktaların kaç iterasyonda sınıflandırıldığı ve kaç tanesinin doğru sınıflandığı gösterilmiştir. Burada öğrenme hızı sabit (0.5) ve verilerin sıralanışı sabit bırakılmıştır.

TABLO-3	BAŞLANGIÇ AĞIRLIKLARI	İTERASYON SAYISI	DOĞRU SINIFLANDIRILAN VERİ SAYISI
1	[0 0 0 0]	2	16
2	[1 1 1 1]	2	16
3	[-1 -1 -1 -1]	1	16
4	[100 100 100 100]	2	16
5	[-100 -100 -100 -100]	1	16
6	[54 0 -27 98]	1	16
7	[-42 72 0 -33]	2	16
8	[0 19 -32 -1]	1	16
9	[1 -777 -1 3]	2	16
10	[-1 3 -1 -879]	2	16

Tablo-3’de görüldüğü üzere lineer ayrıştırma işleminden sonra elde edilen veriler, ağırlık vektörünün başlangıç koşuluna bağımlı olarak 1 veya 2 iterasyonda karar düzleminin bulunabileceği hale gelmiştir. Ağırlık

vektörünün çok büyük aralıklarda değiştiği hallerde dahi karar düzleminin 2 iterasyonda bulunması yapılan dönüşümün verileri kolayca doğrusal ayrıştırılabilir hale getirecek bir dönüşüm olduğunu göstermektedir.

Eğitim kümesinin farklı sıralanışı sonucu ve öğrenme hızının değerlerinin değiştirilmesi doğru sınıflandırılan veri sayısını etkilememiştir.

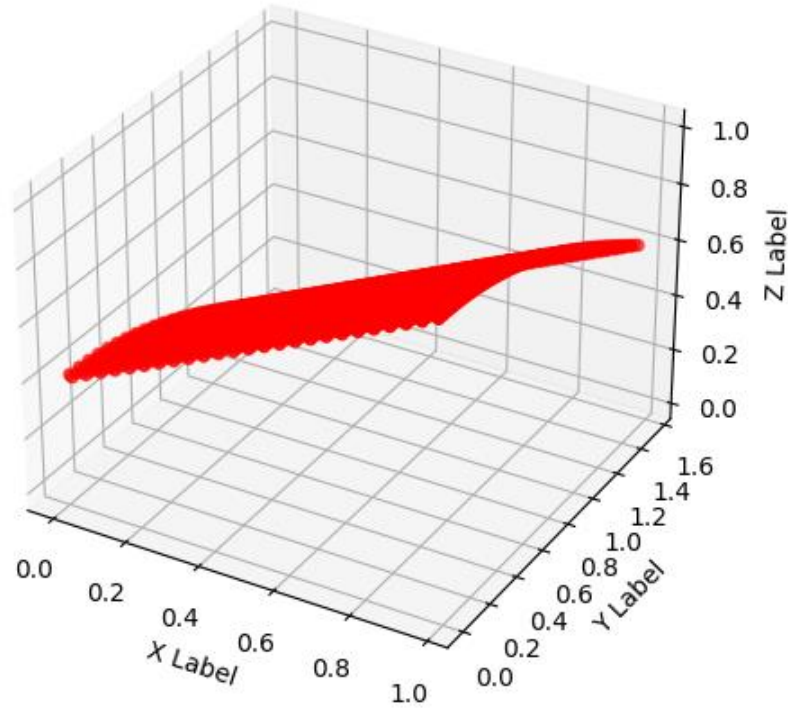


### 3.Soru

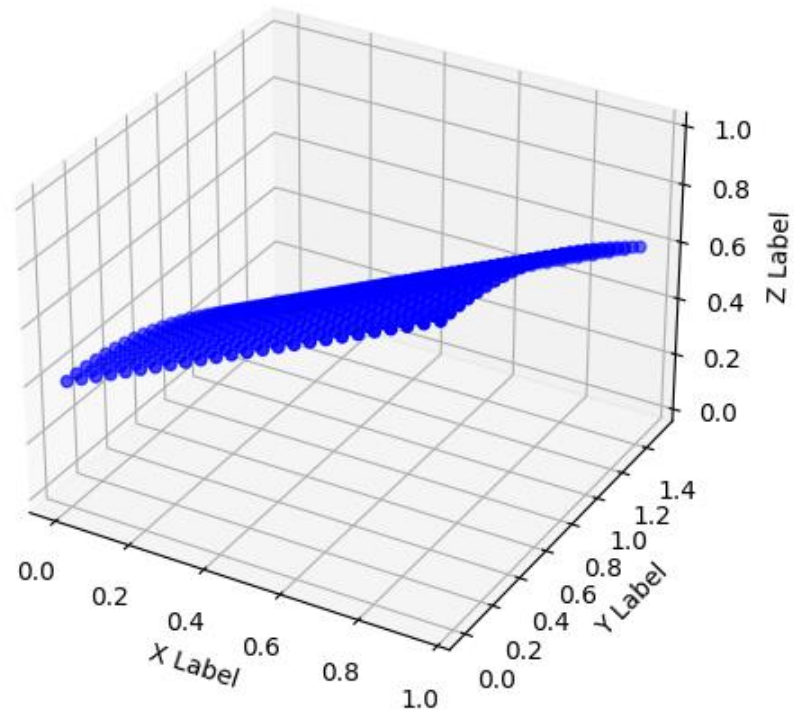
Verileri homojen bir şekilde seçebilmek için  $P(0.02*2i, \frac{\pi}{100}*2i)$ ,  $i=0, 1, 2, \dots, 25$  noktalarını seçtik. Böylece 0.02 birime bölünmüş  $x_1$  ve  $\frac{\pi}{100}$  birime bölünmüş  $x_2$  aralığından adım aralıklarının çift tamsayı katlarının kombinasyonlarına denk gelen 625 tane noktayı seçtik. Eğitim kümemizi ise geri kalan 1875 noktadan oluşmaktadır.

Aşağıdaki resimler veri kümemizi göstermektedir.

Train Data



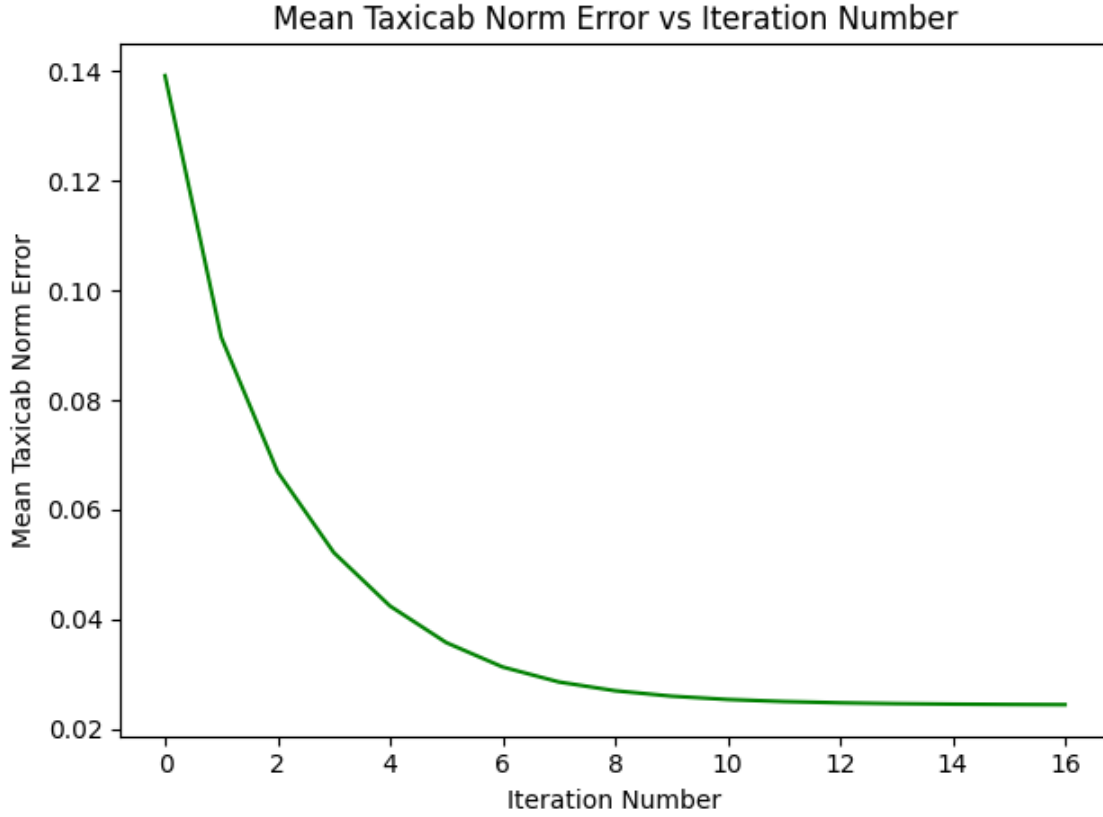
## Test Data



Durdurma kriteri olarak bir iterasyon içinde tüm veri noktaları için yapılan yapılan hatayı bir hata vektörü içine kaydettik. Hata vektörünün Manhattan normunu hesapladık ve vektörün girdi sayısına böldük. Böylece bir iterasyon içinde fonksiyonun eğitim kümesi noktaları için gerçek değerden ne kadar saptığının ortalamasını hesaplamış olduk. Durdurma kriteri olarak bu ortalama hatanın küçük olmasını istediğimiz değerler belirledik. Aşağıdaki tabloda (Tablo-4) belirli bir tolerans ile kaç noktanın doğru modellendiği, bunun için kaç iterasyon geçtiği ve bu değerlerin durdurma kriterinin değerine bağlılığı gösterilmiştir. Burada öğrenme hızı (0.04) sabit ve eğitim kümesinin 5 farklı şekilde sıralandığı durumun ortalaması alınmıştır. Başlangıç ağırlığı  $w=[0 \ 0 \ 0 \ 0]$  alınmıştır.

Durdurma Sınırı	İterasyon Sayısı	Doğru Sınıflandırılan Veri Sayısı	Tolerans
$E<0.035$	7	615.6	0.1
$E<0.035$	7	588	0.08
$E<0.035$	7	449	0.04
$E<0.03$	8	620.6/625	0.1
$E<0.03$	8	594.6/625	0.08
$E<0.03$	8	473.4/625	0.04
$E<0.025$	13	624/625	0.1
$E<0.025$	13	611.2/625	0.08
$E<0.025$	13	518.3/625	0.04
$E<0.0245$	16	625/625	0.1
$E<0.0245$	16	614.2/625	0.08
$E<0.0245$	17	524.8/625	0.04

Tablo-4’de görüldüğü üzere durdurma sınırının alabileceği üst değer küçüldükçe veriler daha yüksek doğruluk ile modellenebilmektedir. Buna karşılık yapılan iterasyon sayısı artmaktadır. Aşağıdaki grafikte yapılan iterasyon sayısı ile durdurma şartını belirleyen hatanın nasıl değiştiği tasvir edilmektedir.



Grafikte görüldüğü üzere iterasyon sayısı arttıkça hata belirli bir değere yakınsamaktadır. Durdurma sınırını 0.0233 yapınca iterasyon üst sınırı olan 5000 iterasyona ulaşıyoruz, dolayısıyla bu yakınsanan hata değerinin 0.0233 civarında olduğunu söyleyebiliriz. Bu değer genlikte ayrık algılayıcımızın bu şartlar altında olabilecek en iyi performansını belirler.

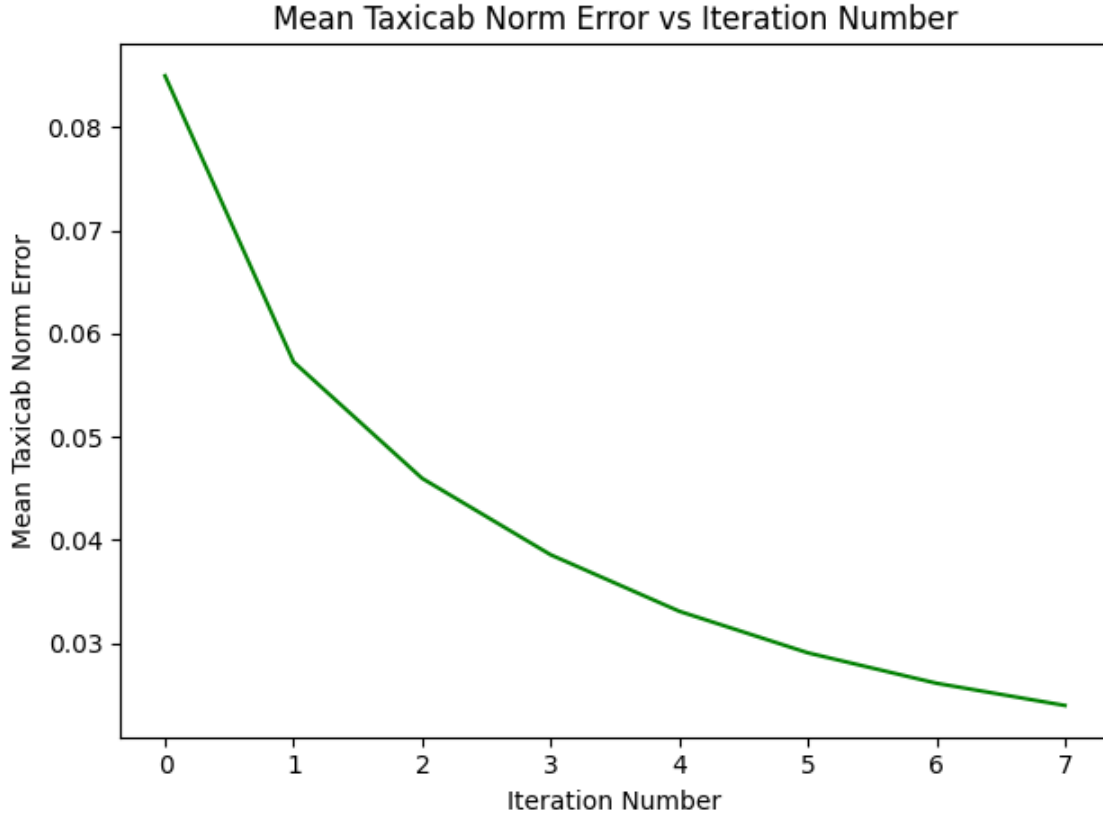
## 4.Soru

ADALINE’da yaptığımız değişiklik sonucunda sabit öğrenme hızında,  $w=[0\ 0\ 0\ 0]$  eğitim kümesi vektörlerini karıştırarak elde ettiğimiz 5 farklı sonucun ortalamasını alarak belirlenen tolerans değeri ile kaç tane test verisinin doğru sınıflandığını ve bunun için kaç iterasyon gerektiğini gösteren tablo (Tablo-5) aşağıda oluşturulmuştur. Verileri oluştururken 3.Soruda kullandığımız veri kümesinin aynısını kullandık.

Durdurma Sınırı	İterasyon Sayısı	Doğru Sınıflandırılan Veri Sayısı	Tolerans
$E<0.035$	3	615.4/625	0.1
$E<0.035$	3	587.8/625	0.08
$E<0.035$	3	453./625	0.04
$E<0.03$	4	623/625	0.1
$E<0.03$	4	604/625	0.08
$E<0.03$	4	490.8/625	0.04
$E<0.025$	5	624/625	0.1
$E<0.025$	5	613.6/625	0.08
$E<0.025$	5	509.2/625	0.04
$E<0.0245$	6	625/625	0.1
$E<0.0245$	6	617/625	0.08
$E<0.0245$	6	515/625	0.04

Bu yöntem ile modelleme yaptığımızda yüksek nispeten yüksek tolerans değerleri için doğruluk soru-3’dekine göre daha yüksektir. Tolerans değerini düşürdüğümüzde soru-3’deki metodun daha doğru modelleme yaptığını görürüz. Buna karşılık 3.soruda yapılan iterasyon sayısı daha fazladır. Burada kullandığımız modelin daha az hassas ama daha hızlı olduğunu söyleyebiliriz.

Aşağıdaki resimde hatanın iterasyon ile değişimi tasvir edilmiştir.



Resimde de görüldüğü üzere burada hata, iterasyon sayısı ile 3.sorudakine nazaran daha hızlı azalmaktadır.

## KAYNAKÇA

1. <http://web.mit.edu/dxh/www/convex.pdf>
2. <https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.121.6481&rep=rep1&type=pdf>
3. <https://www.youtube.com/watch?v=vPDPE66nhlo>
4. [https://en.wikipedia.org/wiki/Graham\\_scan](https://en.wikipedia.org/wiki/Graham_scan)