****

**T.C.**

**YILDIZ TEKNİK ÜNİVERSİTESİ**

**KİMYA METALURJİ FAKÜLTESİ**

**MATEMATİK MÜHENDİSLİĞİ**

**Bulanık Mantık Tabanlı Trafik Analizi**

**LİSANS TEZİ**

**1**

# ÖZET

**2**

İçindekiler

**ÖZET**……..……..……..……..……..……..……..……..……..……..……..……..……..………2

**1.GİRİŞ**……..……..……..……..……..……..……..……..……..……..……..……..……..…….4

1.1 Bulank Mantık Nedir ve Nasıl Ortaya Çıkmıştır……………………………………………...41.2 Trafik Analizinde Bulanık Mantık Tabanlı Modelleme…..…………………………………..5

1.3 Proje Kapsamı..………………………………………………………………………………..5

**2.ÖNBİLGİLER………………………………………………………………………………….6**

2.1 Bulanık Tabanlı Modellemede Kullanılacak Veri…………………………………………….6

2.2 Gerekli API Hizmetleri……………………………………………………………………......7

2.3 Veri’nin aktarılması ve saklanması.…………………………………………………………...7

**3.BULANIK MODELLEME**..…………………………………………………………………10

3.1 Önerilen Yaklaşım…………………………………………………………………………...10

3.2 Üyelik Fonksiyonu…………………………………………………………………………...10

**4.1 ÖĞRENME MODELİ………………………**…………...…………………………………12

4.1 Yapay Sinir Ağları.....………………………………………………………………………..12

4.2 Derin Öğrenme Nedir………………………………………………………………………..12

**3**

**1.GİRİŞ**

* 1. **Bulanık Mantık Nedir ve Nasıl Ortaya Çıkmıştır?**Bulanık mantık veya puslu mantık Lütfi Zade tarafından 1965 yılında yayınlanmış bir makalenin sonucunda ortaya çıkmıştır[1]. Yıllardır bildiğimiz klasik mantığın aksine kesinliği reddetmiş ve kısmi durumların ortaya çıkabileceğinden bahsederek bir sistem geliştirmiştir. Klasik mantıkta önermeler doğru ve yanlış olarak değerlendirilir. Lutfi Zade ise bu mantığa “Ne kadar doğru?” sorusu ile yaklaşmıştır. Çünkü bir önermenin “çok doğru, doğru, kısmen doğru, kısmen yanlış, yanlış, çok yanlış” gibi birçok ibareye sahip olabileceğini düşünür. Örneğin 40 derece klasik mantık için “sıcak”, aynı zamanda 80 derece de klasik mantık için “sıcak” ama bu durumları bulanık mantıkta incelersek 40 derece “sıcak” ibaresine sahip olurken 80 derece “çok çok sıcak” olmalıdır. İnsan, doğası gereği kesin değildir ve gerçek hayat da beyaz ve siyah değildir. Bu gri alanları tanımlanması için gerekli olan mantık sistemi Bulanık Mantıktır.

Bulanık mantığın en büyük üstünlüğü yukarıda bahsedildiği gibi insan diline çok yakın olmasıdır. Bulanık mantık sistemi oluşturulurken belirli fonksiyonlardan yararlanılır ve bu kullanılan fonksiyonlar, bizim girdilerimiz için 0 ile 1 arasında değerlerle tanımlanır. Atılan bu değerler üyelik olarak tanımlanır. Bulanık mantık sistemi oluşturulurken kurallar konuya vakıf kişiler tarafından belirlenmelidir. Bir kural ortaya atılmadan önce uzun testlere tabi tutulmalı ve bu testlerden elde edilen verilerin sonucunda bir karara varılmalıdır. Klasik mantığın aksine bozuk ve gürültülü verileri incelenmesine de olanak sağlar. Matematiksel modellemesinin oluşturulması kolaydır. Esnek bir sistem olduğundan mütevellit kuralların eklenmesi veyahut eksiltilmesi sistemi bozmamaktadır. Birçok avantajının yanı sıra dezavantajları da vardır. Bu sistemlere, sistematik olarak yaklaşmak çok sağlıklı değildir. Sistemin klasik mantığa nazaran daha basit olması gerekmektedir. Kompleks olduğu vakitlerde anlaşılması çok daha zordur. Sistematik yaklaşım ve modelleme çok tercih edilmemektedir.

Günümüz teknolojisinde gelişen makine öğrenmesi ve yapay zekâ ile bulanık mantık kullanımı da artmıştır. Tüketici odaklı geliştirilen fotoğraf makinesi veya video kayıt cihazlarından, dünya devi otomotiv markalarının şanzıman ve frenleme sistemlerine kadar her yerde kullanılmaktadır. Gelişen bilgisayarlarda büyük verilerin analiz edilmesi eskiye nazaran çok daha hızlı ve ekonomik olduğundan dolayı bir bulanık mantık sistemi oluşturulurken gerekli testlerin yapılması ve bu testlerin sonuçlarına erişimi çok daha kolaydır.

4

* 1. **Trafik Analizinde Bulanık Modelleme**

Bu proje günlük hayatta birçok insanın en büyük problemlerinden biri olan trafik yoğunluğunu analiz etme üzerine kurgulanmıştır. Ortalama bir Yıldız Teknik Üniversitesi öğrencisi gününün üç saatini trafikte geçirmektedir. Bu proje kapsamında trafiğe bir çözüm üretilmemektedir fakat bulanık mantık modellemesi sayesinde bir a noktasından bir b noktasına giderken hangi saatlerinin en verimli olacağını tahmin edebileceğiz. Ayrıca bu projenin bir diğer amacı ise güzergahın süresini hiç bilmeyen birisine yardımcı olabilmektir. Örneğin İzmir’de bir yolun hakkı 15 dakikaysa ve bu yol en yoğun trafik halinde 25 dakika sürüyorsa İzmirli bir vatandaş 25 dakikaya çok fazla diyecektir ve o yolu kullanmayacaktır. Aynı süre İstanbul’da yaşayan birisi için çok kısa gelecektir ve yolu kullanmak isteyecektir. Oysa bu kişi bu yol için 25 dakikanın fazla olacağından haberdar olursa farklı yollar kullanabilir veya çıkacağı saati değiştirebilir.

**1.3 Proje Kapsamı**

Bu proje en nihayetinde a noktasından b noktasına gitmenin en uygun zamanına dayanmaktadır. Proje iki kısma ayrılmaktadır.

Birinci kısımda lokasyonlar belirlenmiştir. Yapılan araştırmalar sonucunda YTÜ öğrencilerinin en çok kullandıkları güzergâh incelenmiş ve Altunizade çıkış noktası olarak belirlenmiştir. Bu durumda “a” noktası yani başlangıç konumu Altunizade Metro İstasyonu olmaktadır. Bitiş yani “b” noktamız ise Yıldız Teknik Üniversitesi Davutpaşa kampüsü seçilmiştir. İstanbul öngörülmesi güç bir şehir olduğundan dolayı tek bir veriden analiz edilmesi mümkün değildir. Proje esnasında kurulan sistem sayesinde haftanın yedi günü, yarım saat aralıklarla bu güzergâh incelenmiştir. Google API kullanılmıştır. Google, algoritmasına istinaden kullanıcıyı en hızlı şekilde götürmeyi hedeflemektedir bu nedenle trafiğin duracağı dakikalarda farklı kestirme yolları önermektedir. İnceleme sonucunda görünen odur ki, yol kilometre bazında azalsa dahi belli saatlerde ulaşım süresi çok fazla artmaktadır. En temelinde kullanıcının gideceği ortalama hız düşmektedir. Burada yoğun trafik ifadesinin tanımlanması gerekmektedir. Eğer "a” noktasından “b” noktasına seyahat halindeyken araç olması gereken – yasal hız sınırını geçmemek şartıyla- hızından çok daha yavaş ilerliyorsa o mevkide trafik yoğundur. İncelenen yedi günün ardından girdi olarak kullanılacak veriler toplanılmıştır. Bu veriler üzerinde işlemler yapıldıktan sonra veriler bulanık mantık sistemine girdi olarak girebilecek hale gelmiştir.

İkinci kısım ise mantık sisteminin kurulmasıdır. Bulanık mantık sistemi kurulacağı vakit verilerin en büyük, en küçük ve ortalama değerleri bulunmuştur. Bulanık mantık sistemi kullanıcının yararı olan kısmı 1 olarak tanımlar. Bu durumda verinin içerisinden aracın ortalama hızının en yüksek olduğu durumu 1 ve en düşük olduğu durumu 0 olarak tanımlayarak fonksiyon oluşturulmuştur. Oluşturulan fonksiyon grafiğe dönüştürüldüğünde her günün çok farklı değerler doğurduğunu gözle görülmüştür.

5

Proje başlangıcında tüm günlerin ortalama değerlerini alarak genel bir grafik hazırlanması amaçlanırken sonradan hafta sonu ve hafta içi olmak üzere iki parçaya ayrıştırılmasına karar verilmiştir. Bu kararın sebebi ise grafiğin çizimi esnasında değerler gerçeği yansıtmayacak ve kullanıcıya yanlış çıkarım yapmasına sebebiyet verecektir.

**2.ÖNBİLGİLER**

Proje kapsamında takip edilecek yol:

1) Ham veriyi toplama, düzenleme ve analize hazır hala getirme

2)Analize hazır veri üzerinde yapay zeka destekli işlemler yaparak bulanık mantık tabanlı bir derecelendirme sisteminin oluşturulması

3) Bulanık mantık tabanlı derecelendirme sistemini anlama, yorumlama ve sonuç çıkarma

4) Elde edilen sonuçların yapay sinir ağlarında kullanıma hazır haline getirilmesi

5) Son hale gelen veriyi makine öğrenme algoritmaları kullanılarak bir öğrenme algoritması geliştirilmesi

**2.1 Bulanık Tabanlı Modellemede Kullanılacak Olan Veri**

Projenin ilk kısmı, üzerinde çalışılan konu gereği ihtiyaç duyulan trafik verisinin

toplanması ile ilgilidir. Bu kısımda ihtiyaç duyulan parametreler şu şekildedir:

* A noktası,
* B noktası,
* A ve B noktası arasındaki
  + Mesafe,
  + Ulaşım Süresi ve
  + Ortalama Hız.

Hazırlanan örnek uygulamada bu seçimi yaparken iki noktanın da sık kullanılan yerler olmalarına dikkat edilmiş olup bu sebepten dolayı

* A noktası Altunizade Metro İstasyonu olarak,

6

* B noktası ise Yıldız Teknik Üniversitesi Davutpaşa Kampüsü olarak seçilmiştir.

Sonrasında ise projenin önemli aşamalarından olan seçilen bu iki nokta arasındaki

ihtiyaç duyulan parametreleri elde etme aşamasına geçilmiştir.

**2.2 Gerekli API hizmetleri**

Bu aşamada elde edilecek veriler trafik ile ilgili veriler olduğu için “Google Cloud

Platform” içerisinde yer alan “Google Maps” hizmetlerinden olan “Geocoding API” ve

“Distance Matrix API” isimli API’ler kullanılmıştır.Toplanan verilerin aktarılması v aklanılması için Google Spreadsheet API[4] kullanımı ile daha iyi bir görsel gösterim için günlük tablolar halinde yapılandırılmışlardır.

API (Application Programming Interface), bir uygulamanın, servisin veya platformun sahip olduğu yeteneklere izin verilen sınırlandırmalar dahilinde dışarıdan erişilebilmesini sağlayan bir arayüzdür. Bahsedilen API’ler ile çalışılırken sırasıyla

1. doğrulama,
2. sorgu atma ve
3. sorgu cevabı alma

olmak üzere 3 adımdan oluşan bir algoritma kurulmuştur.

**2.3 Verinin saklanması ve aktarılması**

Bu örnek uygulamanın yazılımı Python ile yazılmıştır. Yazılımda sırasıyla

Adım 1: Gerekli kütüphaneler çağırılmıştır.

* Requests: API’lere istek atılmasını ve cevap alınmasını sağlayan kütüphane
* Sched: Her saat başı fonksiyonu ateşlenmesini sağlayan kütüphane
* Datetime: Zaman ve tarihlerle ilgili işlemler yapabilmemizi sağlayan kütüphane
* Pandas: Veriyi işlemede, temizlemede ve makine öğrenmesi gibi alanlarda oldukça efektif şekilde kullanılan kütüphane

7

* Numpy: Diziler ve matrisler üzerinde çalışacak üst düzey matematiksel işlevler ekleyen kütüphane
* Urlencode: API’lere istek attığımız URL’lere parametre ekleme ve URL’leri düzenlemede kullanılan kütüphane
* Matplotlib: Çeşitli şekillerde görselleştirme imkanı sunan bir kütüphane
* SciKit-Learn: Eğri uydurma, regresyon veya tahmin gibi alanlarda kullanılan makine öğrenimi ile ilgili kütüphane
* SciKit-Fuzzy: Bulanık mantık algoritmaları içeren kütüphane

Adım 2: “Google Cloud Platform” servislerine bağlı API’lere istek atabilmemiz için gerekli güvenlik adımları takip edilmiştir. İlgili kod aşağıda verilmiştir.

**Text

Description automatically generated**

**Kod 2.3.1: Google Servis ve Tablo Güvenlik Bilgilerinin Kurulumu**

Adım 3: Geocoding API[2] kullanarak girdi olarak girilen adresin ilçesini, ilini ve koordinatlarını elde eden fonksiyonun inşası.

Adım 4: Distance Matrix API[3] kullanarak girdi olarak iki adet koordinat alan ve o iki

koordinat arası mesafeyi ve ulaşım süresini trafiği dikkate alarak hesaplayıp çıktı

olarak sunan algoritmanın inşası.

Adım 5: Oluşturulan listedeki değerler Google Sheets API[4] kullanılarak proje ekibi üyelerinin yetkisi olan bir google sheet sayfasına gün gün ve saat saat otomasyon ile aktarılmıştır. İlgili kod aşağıda verilmiştir.

Bu 5 adımın takip edilmesi sonucunda, sched kütüphanesi kullanılarak 2 hafta boyunca haftanın her gününde her saat başında veri elde etme işlemi tamamlanmıştır.

Bu sayede

8

* verilerin elde edilmesi,
* verilerin oluşturduğumuz Google Sheet sayfasına aktarılması,
* verilerin orada saklanması

sağlanmıştır.

**Table

Description automatically generated**

Resim 1.1**:** Çarşamba Günü toplanan saat, ortalama hız, ortalama mesafe ve süre verilerinin temsili

Veriler resim 1.1’de gözüken formatta elde edilip toplandıktan sonra projenin sıradaki adımına geçilmiştir. Sıradaki adımda Sci-Kit Fuzzy isimli kütüphane kullanılarak her gün için yukarıda görülen değerler baz alınarak üyelik fonksiyonları oluşturulmuştur.

9

**3.BULANIK MODELLEME**

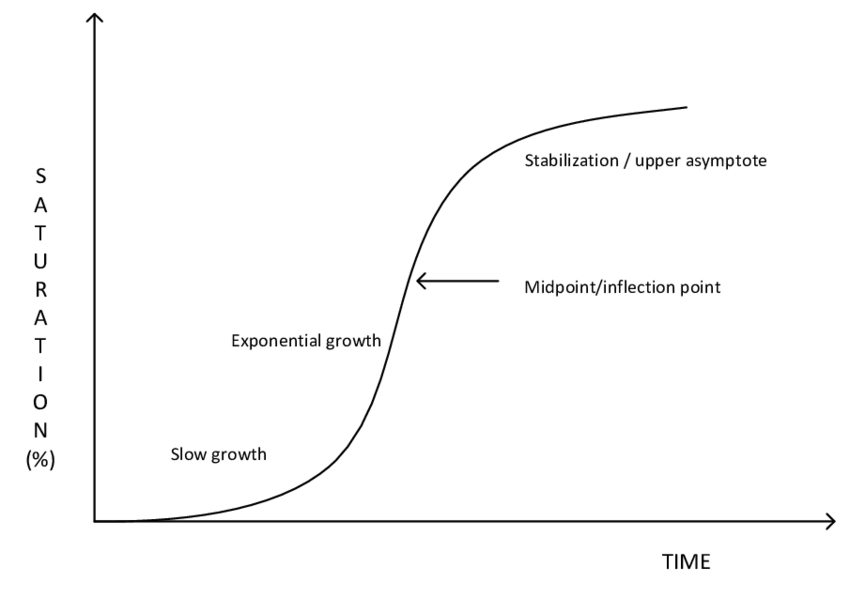
**3.1 Önerilen Yaklaşım**

Bu bölümde 2 hafta boyunca her saat dilimi için elde edilen verilerin sonucunda hangi yöntemlerle üyelik değerleri ve üyelik fonksiyonları belirleneceği kararlaştırılmıştır. Bu verilerin sonucunda belirli bi A noktası ile belirli bir B noktası araındaki mesafe, ulaşım süresi ve yol boyunca gidilen ortalama hız hesaplanmıştır.

Elde edilen ortalam hız değerleri üzerinden hesaplanacak üyelik değerlerinin oluşturulması için s-shape metodu önerilmiştir.

**3.2 S-Shape Metodu ( Sigmoid Growth Curve )**

S-shape metodunun genel amacı bir fonksiyonun verilen değerlere bakılarak belirli bir kural oluşturarak yan “S” harfi şeklinde bir grafik oluşturmaktır.



Şekil.3.1: Örnek S-shape grafiği ve grafiğin bölümleri

10

Şekil 3.1’de görüldüğü gibi bu grafiğin temel mantığı şudur:

Grafik başlangıçta yavaş yükselirken, devamında hızını gittikçe arttıracak şekilde büyür. Grafiğin bu evresinde grafik “J” şeklini alır. Grafiğin devamında grafik maksimum hızda büyür, ancak bir süre sonra grafiğin büyüme hızı azalır. Bu azalma büyümenin stabil olacak şekilde devam edene kadar devam eder. Grafik başlangıç noktasında 0 iken, grafiğin yükselme hızının en düşük olduğu durumda maksimum değerini görür.

S-shape metodu genel olarak başlangıç değerlerin minimum, son değerlerin maksimum olan veriler için uygundur. Aynı zamanda kullanımı ve grafiğin okunması kolay ve anlaşılır olduğundan s-shape metodu tercih edilebilir. Fakat her veri s-shape metodu için uygun olmayabilir. Bu tür durumlar dikkate alınarak uygun metod bulunmalıdır.

Üyelik değerlerinin atanması çalışmasında python üzerinden “scikit-fuzzy” kütüphanesinden yararlanılır. Bu kütüphaneyi kullanılarak üyelik değerleri hesaplanır. Bu hesaplamada 0-1 başlangıç değerlerini sırasıyla en düşük ortalama hız değerine ve en büyük ortalama hız değerine atanır. Arkasından bu iki değer arasındaki noktalar için s-shape metodu ile üyelik değerleri atanır.

Bu metodun matematiksel ifadesi Denklem 3.2’de gösterilmiştir.

Text

Description automatically generated

Denklem 3.2: S-shape metodunun matematiksel gösterimi

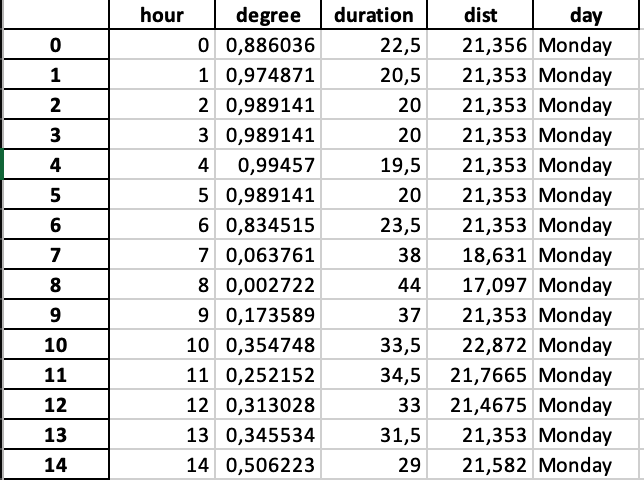
11

Denklem 3.2’den de anlaşılacağı üzere,”a” değeri en düşük ortalama hız değerine eşitken “b” değeri en yüksek ortalama hız değerine eşittir. Dolayısıyla en düşük ortalama hız değerinin üyelik derecesi “0” olurken en yüksek ortalama hız değerinin üyelik derecesi “1” değerini alacaktır. Bu iki değer arasında kalan diğer değerler ise Denklem 3.2’deki fonksiyonda denk gelen sayı aralığına göre işleme sokulurlar ve buna göre üyelik değerlerini alırlar.

**4.ÖĞRENME MODELİ**

Araştırmanın bu kısmında elde edilen üyelik değerlerini girdi olarak kullanacak bir öğrenme modeli gereklidir. Üyelik değerleri 0 ile 1 arasındaki kesirli değerler olduğundan klasik makine öğrenme teknikleri burada başarısız kalmaktadır çünkü klasik makine öğrenme modelleri veriden öğrenirken bu veriye bir sınıflandırma yapar. Bu sınıflandırma çoğu modellerde doğru veya yanlış, kırmızı ya da kırmızı değil şeklinde iki çeşit sınıflandırmadır. Kısaca anlatmak gerekirse klasik modeller sonucu 0 ya da 1 olarak bulurlar, bu ise araştırmanın konusuna terstir.

Bu araştırmada bulunmak istenilen nokta trafiğin var veya yok gibi basit bir sınıflandırma yapmak yerine o yol üzerindeki elde edilen veriler doğrultusunda trafik yoğunluğunun bulanık mantık tabanlı derecesinin öğrenebilmesidir. Bu yüzden seçilecek modelin vermesi gerek sonuç 0 veya 1 değil, verilen verilere göre trafik yoğunluğu derecesini bulmaktır ve bu değer her veri için 0 veya 1 arasındadır. Sonuç olarak modelin sunacağı değer herhangi bir sınıfa ait olma olasılığı değil, üyelik derecesidir.



Tablo 4.1: Modelde kullanılacak veri tablosu

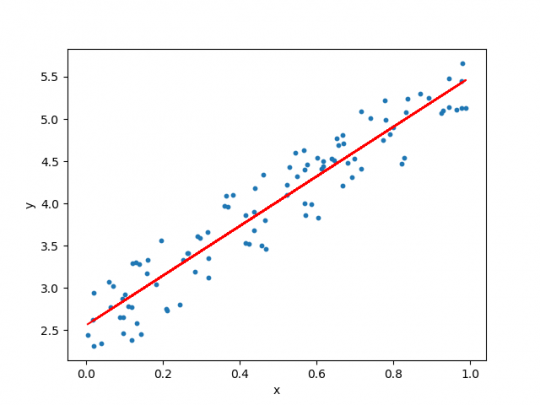
12

Tablo 4.1’de görülen bulanık üyelik değerleri ile oluşturulmuş yeni veri setinde süre, uzaklık, saat ve gün değerleri kullanılarak trafik yoğunluğunun üyelik değerini öğrenebilecek bir yöntem bulunmalıdır. Bunlarn yanı sıra tabloda olmayan saat değerleri için de üyelik değerleri bulunmalıdır. Bunun için ise önerilen yöntem eğri uydurmadır.

**4.1 Eğri Uydurma ( Curve Fitting )**

Eğri uydurma, sonlu bir dizi veri noktasına, en iyi uyan matematiksel fonksiyonu oluşturma işlemidir. Yerleştirilen eğriler, veri görselleştirme teknikleri kullanılarak matematiksel fonksiyondan çıkan değerlere en yakın sonucu bularak grafiği çıkarılır. Eğri uydurma işlemi için belli başlı yaklaşımlar vardır. Bu yaklaşımlar:

1. Regresyon: Hata oranı büyük değerlerin, her bir veri noktasından geçmeyen, verilerin genel eğilimini tek bir eğri ile gösterimidir.
2. İnterpolasyon: Hata oranı düşük değerlerin, iyi bilinen ayrık noktaların her birinden geçecek şekilde eğri uydurarak gösterimidir.



Şekil 4.1: Eğri Uydurma Örneği

Araştırma verilerine uygun olacak şekilde regresyon kullanılmıştır. Bunun nedeni Şekil 4.1’de görüldüğü gibi verinin bütününe bakarak genel bir grafik oluşturmaktır. İnterpolasyon oluşturmada hata payı daha az olmasına karşın, regresyonu değerlerin birbirlerine

13

yakın olduğu zaman daha uygun bir sonuç ortaya çıkar. Fakat regresyonun interpolasyona göre dezavantajı ise, birbirlerinden uzak değerlere sahip bir veride regresyonun vereceği hata payı çok yükselir ve bu olumlu sonuç vermez. Bu durumlar gözden geçirilerek regresyon yapılması uygun görülmüştür.

A picture containing diagram, line, plot, text

Description automatically generated

Şekil 4.2: Üyelik değerleri üzerinden uydurulmuş bir regresyon modeli

Eğri uydurma ile aradaki saat değerleri için de bir üyelik derecesi tahmini gerçekleştirilebilmiştir ama Tablo 4.1 veri setinden bir öğrenme algoritmasının kurulabilmesi için eğri uydurma metodu tek başına yetersiz kalacaktır ve bu yüzden bir yapay sinir ağı algoritmasının gereksinimine ihtiyaç duyulmuştur.

**14**

**4.2 Yapay Sinir Ağları Nedir?**

Yapay sinir ağları adından da anlaşılacağı gibi insanlardaki sinir hücrelerinin çalışma sistemine çok benzer bir şekilde çalışmaktadır. Klasik Von Neumann bilgisayar mimarisi ve yazılımlar, sayısal ve sembol işlemede çok başarılı ancak karmaşık algısal problemlerin çözünde yetersiz kalmaktadır. İnsan beyni ise sayısal ve sembol işlemede daha yavaş kalmasına rağmen karmaşık algısal işlemlerde, idrak etme ve deneyimlerle kazanılmış bilgileri kullanma konusunda çok hızlı ve başarılı durumdadır.

Algısal işlemlerde başarılı olmak için, Yapay Sinir Ağı, beyindeki biyolojik sinir ağlarının yapısını, öğrenme, hatırlama ve genelleme kabiliyetlerini taklit eder. Öğrenme işlemi örnekler kullanılarak gerçekleştirilir. Öğrenme sonrasında yapay sinir ağı örüntü tanıma/sınıflandırma (ses, yazı, yüz, şekil, nesne, vb.), kümeleme/gruplama, optimizasyon, tahmin, kontrol, karar, vb. işler için kullanılabilmektedir[6].

A diagram of a machine

Description automatically generated with low confidence

Şekil 4.2.1: Temsili Yapay Sinir Ağı[9]

**4.3 Model Seçimi**

Oluşturulacak modelden istenen çıktı sonucu tek katmanlı olmalıdır yani bir değer bulmalıdır. Bunun içinse eğri uydurma metodu önerilmişti ( Bkz: Bölüm 4.1 ). Önerilen metodu kullanabilmek için ve Tablo 4.1’deki veri setinden yola çıkarak en uygun yapay sinir ağ modeli, python sklearn neural network kütüphanesinden “MLP Regressor” olarak belirlenmiştir[7].

**4.3.1 MLP Regressor ( Multi-Layer Perceptron Regressor )**

Perceptron, girdi verilerindeki özellikleri tespit etmek amacıyla kullanılan bir yapay sinir ağı sistemidir. Perceptron öğrenme algoritması, 1957 yılında Frank Rosenblatt tarafından Cornell Havacılık Labaratuarında icat edilmiştir[8].

15

MLP Regressor eğri uydurma görevleri için kullanılan bir yapay sinir ağı tipidir. MLP Regressor, girdi öznitelikleri ve çıktılar arasındaki ilişki lineer olmadığı veya kompleks olmadığı zaman en iyi sonuçları vermektedir ve bu da Tablo 4.1’deki veri setinin MLP regressor için uygun olduğu anlamına gelmektedir.

**4.3.2 Model Aktivasyon Fonksiyonu Seçimi**

Aktivasyon fonksiyonları yapay sinir ağlarının öğrenme aşamasında her bir katmandaki her bir nöronda hesaplanan çıktı değerlerini girdi değeri olarak kullanan bir fonksiyondur ve birçok tipi mevcuttur. MLP Regressor modelinde kullanılabilen aktivasyon fonksiyonları aşağıdaki gibidir:

1. Tanh: Hiperbolik tanh fonksiyonu: f(x) = tanh(x)
2. ReLu: Düzeltilmiş lineer birim fonksiyonu: f(x) = max(0, 1)
3. Logistic: Mantıksal sigmoid fonksiyonu: f(x) = 1 / (1 + exp(-x))
4. Identity: Lineer fonksiyon: f(x) = x

Yukarıda listelenen aktivasyon fonksiyonlarının girdi değeri olan x şu şekilde hesaplanır:

Girdi katmanındaki her bir nörondan gelen girdi değerleri önceden belirlenen ( genellikle rastgele olarak ) ve her bir girdiye karşılık gelen ağırlık değerleri ile çarpılır ve bu çarpımın sonucu bias değeri ile toplanır.

Girdi Matrisi \* (Ağırlık Matrisi)T + bias = x,

Çıktı = Faktivasyon(x)

Aktivasyon fonksiyonundan geri dönen çıktı değerleri bir sonraki katmanda kullanılır.

**REFERANSLAR:  
  
[1] Fuzzy Logic Introduction - M. Hellmann - Universite de Rennes** Erişim:02.03.2023

**[2] Google Maps Platform.” Overview | Geocoding API | Google** Developers”.Erişim:02.03.2023.https://developers.google.com/maps/documentation/geocoding/.

**[3] Google Maps Platform.” Overview | Distance Matrix API | Google** Developers”.Erişim:02.03.2023.https://developers.google.com/maps/documentation/distance- matrix/.

**[4] Google Cloud Platform.” Overview | Sheets API | Google** Developers”.Erişim:02.03.2023.https://developers.google.com/sheets/api/guides/concepts.

**[5]** **Mathworks.”S-shaped Membership Function** – MATLAB smf”.Erişim:12.04.2023.

https://www.mathworks.com/help/fuzzy/smf.html

**[6] Yapay Sinir Ağları ve Tek Katmanlı Ağlarda Öğrenme –**https://www.linkedin.com/pulse/yapay-sinir-a%C4%9Flar%C4%B1-ve-tek-katmanl%C4%B1-a%C4%9Flarda-%C3%B6%C4%9Frenme-tanju-do%C4%9Fan/ Erişim:06.05.2023

**[7] Sklearn Neural Network MLPRegressor Kütüphanesi -**

[**https://scikit-learn.org/stable/modules/generated/sklearn.neural\_network.MLPRegressor.html**](https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPRegressor.html)Erişim:07.05.2023

**[8] Sefa İşci – Machine Learning Engineer**

[**https://medium.com/kaveai/percepton-1-550f83895555**](https://medium.com/kaveai/percepton-1-550f83895555)Erişim:07.05.2023

**[9] Şekil 4.2 kaynağı:** [**https://ebrary.net/htm/img/29/825/41.png**](https://ebrary.net/htm/img/29/825/41.png)Erişim:07.05.2023

**[10] Önişleme:** [**https://scikit-learn.org/stable/modules/preprocessing.html#normalization**](https://scikit-learn.org/stable/modules/preprocessing.html#normalization)Erişim:29.05.2023