

BULANIK MANTIĐIN YAPAY ZEKA UYGULAMALARI

Alara BAĐCI, 19052032, Matematik MühendisliĐi

Denizhan KELEř, 18052002, Matematik MühendisliĐi

Barıř KOÇYİĐİT, 18052041, Matematik MühendisliĐi

AltuĐ USCA, 19052623, Matematik MühendisliĐi

Sude Naz DoĐukan, 19052068, Matematik MühendisliĐi

ÖZET: Bu makalede bulanık mantık ve yapay zeka hakkında bilgi verilmiř ve bulanık mantığın bir yapay zeka tekniĐi olduĐu gösterilmek istenmiřtir.

Bu tekniĐe uygun literatür taraması yapılmıř ve çeřitli uygulamalardan bahsedilmiřtir. Bulanık mantık mimarisi önerilen yaklařım olarak kullanılmıř ve ayrıntılı olarak anlatılmıřtır. Önerilen yaklařımın performansı literatürden farklı sayısal örnekler dikkate alınarak gösterilmiřtir.

Anahtar kelimeler: Bulanık mantık, yapay zeka, üyelik fonksiyonu, makine öğrenimi, bulanık sistemler.

BULANIK MANTIĞIN YAPAY ZEKA UYGULAMALARI

I.GİRİŞ

Günlük hayatımızda, durumun doğru mu yanlış mı olduğunu belirleyemediğimiz durumlarla karşılaşabiliriz. Bulanık, belirsiz veya belirsiz bir şeyi ifade eder. “Yapay Zekada Bulanık Mantık”, muhakeme için değerli bir esneklik sağlar. Bulanık mantık insan gibi düşünmeyi esas almış ve bunları matematiksel fonksiyonlara çevirerek işlem yapan bir daldır. Bulanık mantığın en büyük özelliği de ikili Aristo mantığı yerine “bulanık küme” teorisine dayanan matematiksel bir disiplin olmasıdır. Bazı kavramları kesin sonuçlarla ifade edebilirken bazılarını ise kesin sonuçlarla ifade edemeyiz. İşte bu kesin yargılar arasında kalan değerleri kullanabilmemiz ve makinaların değerlendirmesi için bulanık küme mantığına yani bulanık mantığa ihtiyaç duyuyoruz.

Yapay zekâ alışılmış tanımı ile “Bilgisayarın daha zeki davranmasını sağlayan bir yaklaşımdır.” Farklı bir tanım olarak “Yapay zekâ makinaların daha zeki davranmalarına olanak veren uygulamadır.” Yapay zekâ kullanan akıllı bir makinanın kendisini ve çevresini doğru algılaması ve göstermesi, bu bilgileri kodlaması ve kodlanmış bilgiyi çözmesi, mantıksal çıkarım uygulaması ve bilgiye kolay erişim için bilgileri sıralaması gerekmektedir. İleriye yönelik tahmin

yapabilme yapay zekâ ürünlerinden beklenen bir özelliktir. Yapılan bir hatanın farkına varması ve düzeltilmesi yine yapay zekâ ürünlerde olması gereken özelliktir.

Karmaşık ve zor problemleri sonsuz ihtimal arasından yaklaşık çözümlerle ortadan kaldırdığı için hayli kullanışlı bir yapıdan bahsetmekteyiz ve iş sadece gelecek için kurduğumuz teknolojilerde bitmemektedir. Füzelerden örnek verecek olursak, füzelerde bulanık mantık uçuş kontrol mekanizması olarak iş görmektedir. Bir füze uçuşa geçtiğinde sürdürmesi gereken bir rotaya sahip ve hava akımı, rüzgâr gibi etkenlerle maksimum birkaç santim bu rotadan kayar. Fakat yanlış bir hesap olduysa onu yarı yolda yeniden yönlendirmek gerekecek ve belki de 50 derecelik bir açı değişikliği zorunlu olacaktır. Böylesi bir durumda, bulanık mantıktan faydalanan bir programlamaya ihtiyaç söz konusudur.

Bulanık kümelerin önemli bir özelliği, sayısal değerlere sahip sınıflarla katı olmayan bir şekilde ara yüz oluşturmak için bir çerçeve sağlamaktır. Sınıflandırma problemlerinde bulanık sınıfların kullanılması, bir akıl yürütme aşamasının başlangıcında sınır durumlarının keyfi olarak sınıflandırılması ihtiyacını ortadan kaldırır. Sayısal veriler, sembolik akıl yürütme makinelerini beslemek için dilsel olarak etiketlenmiş bulanık kümeler aracılığıyla özetlenebilir. Bulanık kümelerin tipik olarak yapay zekaya getirdiği şey, muhakeme

cihazlarında dereceliliği yakalamak için matematiksel bir çerçevedir. Böyle bir derecelilik teorisi, birçok insanın inanma eğiliminin aksine, sayısal olması gerekmez, ancak tamamen sıralı (kafes tabanlı) olabilir. Ayrıca, derecelendirilmişlik çeşitli biçimler alabilir: önermeler arasındaki benzerlik, belirsizlik seviyeleri ve tercih dereceleri. Bize göre, insanlar bu çeşitli derecelendirme türlerini matematiksel olarak sağlam ve hesaplama açısından izlenebilir bir şekilde ele alma yeteneğine inanmaya başladıklarında, bulanık mantığın AI araştırmasının bir parçası olarak tam olarak kabul edilme şansı olacaktır.

Bulanık mantık, bir bulanık kontrolcü olarak ilk kez 1970 yılların ortasında, Londra 'ki Queen Mary College'de Prof. Ebrahim H. Hamdani tarafından bir buhar makinesinde uygulanmıştır. [1] Ticarî olarak ise ilk defa, 1980 yılında, Danimarka'daki bir çimento fabrikasının fırınına kontrol etmede kullanılmıştır. Çimentoda kullanılan kimyasal maddeler arzu edilen sıcaklığın üzerinde bir derecede yakılırsa, ortaya çıkan katı madde ufalanamayacak kadar sert olur. Bu sıcaklığın altında bir derecede yakmak ise kaliteyi düşürür. Bulanık mantık ile hazırlanan bir dizge, bilgisayar desteğinde, duyarlı alıcılardan, ısı ve maddelere ait bilgileri alır ve geri-besleme mekanizmasıyla değişkenleri kontrol ederek, bu ayarlama işini çok hassas ölçümlerle gerçekleştirir ve önemli oranda enerji tasarrufu temin etmiştir. [3] Dünyanın en gelişmiş metrosu olarak kabul edilen Japonya'daki Sendai metrosunda,

yaklaşık 14 km boyunca 16 istasyonda duran tren, o kadar yumuşak hareket etmektedir ki, ayaktaki yolcular sadece hafifçe sallanmaktadır. Bu dizgenin temelinde "bulanık mantık" vardır. Ondaki etki-tepki süresi insanlarınkinden üç kat daha kısadır. Ayrıca %10 yakıt tasarrufu sağlar ve şimdiye kadar hiçbir tehlikeye yol açmamıştır.[4] Bulanık mantık dizgesine dayanan fotoğraf makineleri, otomatik odaklama yapan klasik makinelerden bile daha net bir görüntü sunmaktadır. Fotokopi makineleri ise bulanık mantıkla çok daha kaliteli kopyalar çıkarmaktadırlar, çünkü odanın sıcaklığı, nemi ve orijinal kâğıttaki karakter yoğunluğuna göre değişen resim kalitesi gibi faktörler hesaplanarak mükemmele yakın hale getirilmektedir.[5] Bulanık mantığın önemli bir uygulama alanı da küresel bilgisayar ağı/internettir. İnternette arama sorunuyla ilgili olarak bazı şirketler "yüksek görünürlük" sunan bulanık mantığı kullanmaktadır. Çünkü internet uygulamaları, hareketli geçişlerden daha çok görünürlüğe sahiptir.[6] Gelişmiş Petrol Geri Kazanım taramalarında bir yapay zekâ tekniği olarak bulanık mantık kullanılmıştır. Petrol viskozitesi, rezervuar derinliği, kaya geçirgenliği, petrol doygunluğu gibi kriterler kullanılarak sahip oldukları üyelik dereceleriyle en uygun EOR yöntemi seçilmesinde faydalıdır.[7] Reaktör kontrolünde yapay zekâ muhtemeline bulanık mantık uygulamaları vardır. Yapay zekada yaklaşık akıl yürütme yöntemleri, nükleer santrallerin kontrolü için karar destek araçları oluşturmak için

kullanılabilir. Bir nükleer reaktörün kontrol odası operatörü, belirli bir anda birden fazla alarmın çaldığı durumlara periyodik olarak maruz kalır. Bu operatörler bir şeyler ters giderse tesisin ana bileşenlerinin nasıl çalıştığını bilir ve uygun eylem planını belirleyebilir. Yine de bir kriz durumunda operatör yorgunluğunun yanı sıra anormal reaktör koşulları sırasında operatörlere yardımcı olmak için uzman sistemlerin tasarımına ve uygulanmasına ihtiyaç doğmuştur.[8] Öğrencilerin bireysel özelliklerinin belirlenmesinde yine bulanık mantık, yapay zekâ tekniği olarak kullanılabilir. Günümüzde hızla gelişen bilim ve teknoloji, sosyal, kültürel ve siyasal oluşumlar bireylerin yetenekleri ve bireysel farklılıkları doğrultusunda eğitilmelerini gerektirmektedir. Bireyin zekasının türü ve öğrenme stili genellikle öğrenmedeki bireysel farklılıkları dikkate almak için kullanılır. Bu özelliklerin tespitinde kullanılan ölçeklerin değerlendirilmesi eğitimciler tarafından yapılmaktadır. Ancak hızla gelişmekte olan yapay zekâ teknolojileri birçok alanda ve çeşitli amaçlar için kullanılabildiği gibi bu ölçeklerin değerlendirilmesinde de kullanılabilirler.[9] Dünya üzerinde video oyun oynayan birçok insan vardır. Oyun yapay zekâsı, oyunları zekâ yanılsaması ile güçlendirmekle ilgilenen video oyunu geliştirme dalıdır. Game AI, basit sonlu durum makinelerinden son teknoloji evrimsel algoritmalara kadar daha geniş yapay zekâ alanından birçok tekniği ödünç alır. Bu teknikler arasında bulanık mantık, iyi bir video oyunu yapay

zekâ geliştiricisinin cephaneliğinde bulunması gereken araçlardan biridir, çünkü formülasyonunun basitliği ve ifade gücü ile birleşir. Oyunlarda oyuncunun gerçeğe yakın deneyim yaşayabilmesi için “NPC-Non Player Character” yani bilgisayar kontrolündeki karakterlerin ve ajanların doğala yakın davranış sergilemesi gerekmektedir. Uzun bir süredir NPC’lerin oyunlardaki davranışlarını ve oyuncu ile etkileşimini düzenlemek için sonlu durum makineleri kullanılmaktadır. Ulaşılmak istenen hedef bu karakterlerin veya ajanların insan doğasına özgü bir şekilde davranışlarının daha az kestirilebilir düzeye çekilmesidir. Bu aşamada devreye bulanık mantığın gücü girmektedir. Bulanık mantığın sonlu durum makinelerinde kullanımı NPC’lerin oyundaki duygu ve davranışlarını daha doğal hale getirebilmektedir. Bu da oyuncunun oyunu her oynayışında farklı deneyimler yaşamasını sağlayarak oynayışı arttırmaktadır.[10]

II.ÖN BİLGİLER

Tanım 1: Yapay zekâ; bilgisayar ve makineleri kullanarak insan zihninin problem çözme ve karar verme yeteneğini taklit etmek amacıyla ortaya çıkarılmıştır.

Tanım 2: Makine öğrenimi; insanların öğrenme şeklini taklit etmek için veri ve algoritmaların kullanımına odaklanan ve doğruluğunu kademeli olarak arttıran bir yapay zekâ ve bilgisayar bilimi dalıdır.

Tanım 3: Hesaplamalı Zekâ (CI); biyolojik ve dilsel olarak desteklenmiş

hesaplama paradigmalarının teorisi, tasarımı, uygulaması ve geliştirilmesidir. Genel olarak CI üç kategoriye ayrılır: Sinir Ağları, Bulanık Sistemler ve Evrimsel Hesaplama.

Tanım 4: Yapay Sinir Ağları; insan beyninin davranışını yansıtarak bilgisayar programlarının yapay zekâ, makine öğrenimi ve derin öğrenme alanlarındaki kalıpları tanımasına ve ortak sorunları çözmesine olanak tanır.

Tanım 5: Üyelik fonksiyonu için; bir verinin, kümeye olan aidiyetini 0 ile 1 arasında değerlendiren fonksiyondur. Üyelik fonksiyonunun çıktısı, üyelik derecesi olarak tanımlanır.

Tanım 6: Uzman sistemler için; normalde insan uzmanlığı gerektiren özel bir alandaki sorunları çözmek için yapay zekâ yöntemlerini kullanan sistemlerdir. Bulanık uzman sistemler ise, Boole mantığı yerine bulanık mantığı kullanan bir uzman sistemdir. Diğer bir ifadeyle, bir bulanık uzman sistem, veriler hakkında akıl yürütmek için kullanılan üyelik fonksiyonları ve kuralların toplamıdır.

Tanım 7: İkili mantık için; bir önermenin sadece doğru ya da yanlış doğruluk değerine sahip olduğu mantık sistemlerine ikili değerli mantık sistemleri denir. Doğruluk değerine ilişkin türlü biçimlerdeki “belirsiz”likleri konu edinen, böylece iki değerli sistemin dışına çıkarak onu genişleten sistemlere ise çok değerli mantık sistemleri denir.

III.BULANIK MANTIK MİMARİSİ

Bulanık mantık mimarisi 4 farklı katmandan oluşur.

1) Kabuller (Rule Bases)

Karar verme sistemini kontrol etmek için uzmanlar tarafından sunulan tüm kuralları ve If- Then koşullarını içerir. Bulanık teorideki son güncelleme, bulanık kontrollerin tasarımı ve ayarlanması için etkili yöntemler sağlar. Genellikle, bu gelişmeler bulanık kuralların sayısını azaltır.

2) Bulanıklaştırma (Fuzzification)

Bulanıklaştırma katmanı, kesin sayıların bulanık kümelerle dönüştürüldüğü adımdır. Kesin küme, aynı özelliklere sahip bir öğeler kümesidir. Belirli bir mantığa göre, bir eleman kümeye ait olabilir veya olmayabilir. Kesin kümeler ikili mantığa dayalıdır (Evet veya hayır.). Burada, daha önceden tanımlanmış üyelik fonksiyonları aracılığıyla tanımlanan dilsel terimler ve fiziksel değerler aşağıda belirtilen 5 farklı kategoriye ayrılır.

- Büyük pozitif (LP)
- Orta derecede pozitif (MP)
- Küçük (S)
- Orta derecede olumsuz (MN)
- Büyük negatif (LN)

Bulanıklaştırma, iki farklı yöntemle sağlanabilir:

a) Destek Bulanıklaştırma(s-bulanıklaştırma):

$$\tilde{A} = \mu_1 Q(x_1) + \mu_2 Q(x_2) + \dots + \mu_n Q(x_n)$$

Burada $Q(X_i)$ bulanıklaştırma çekirdeği (kernel) olarak adlandırılır ve μ 'nin sabit tutulması esasına dayanır.

b) Derece Bulanıklaştırma(g-bulanıklaştırma):

Burada x_i sabittir ve μ_i bir bulanık küme olarak ifade edilir.

Bulanıklaştırma işlemi için pek çok üyelik fonksiyonu kullanılsa da Gauss, üçgensel veya yamuksal üyelik fonksiyonları sıkça tercih edilir.

- Gauss Üyelik Fonksiyonu:

$$F(x; \sigma, c) = e^{-\frac{(x-c)^2}{2\sigma^2}}$$

Burada c ortalama ve σ Gauss fonksiyonunun standart sapmasıdır.

- Üçgensel Üyelik Fonksiyonu:

Üçgen bir üyelik fonksiyonu, aşağıdaki gibi üç parametre $\{a, b, c\}$ ile belirtilir:

$$\text{triangle}(x; a, b, c) = \begin{cases} 0, & x \leq a. \\ \frac{x-a}{b-a}, & a \leq x \leq b. \\ \frac{c-x}{c-b}, & b \leq x \leq c. \\ 0, & c \leq x. \end{cases}$$

Bu fonksiyon şu şekilde de ifade edilebilir:

$$\text{triangle}(x; a, b, c) = \max\left(\min\left(\frac{x-a}{b-a}, \frac{c-x}{c-b}\right), 0\right)$$

- Yamuksal Üyelik Fonksiyonu:

Bir yamuksal üyelik fonksiyonu, aşağıdaki gibi dört parametre $\{a, b, c, d\}$ ile belirtilir:

$$\text{trapezoid}(x; a, b, c, d) = \begin{cases} 0, & x \leq a. \\ \frac{x-a}{b-a}, & a \leq x \leq b. \\ 1, & b \leq x \leq c. \\ \frac{d-x}{d-c}, & c \leq x \leq d. \\ 0, & d \leq x. \end{cases}$$

Bu fonksiyon şu şekilde de ifade edilebilir:

$$\text{trapezoid}(x; a, b, c, d) = \max\left(\min\left(\frac{x-a}{b-a}, 1, \frac{d-x}{d-c}\right), 0\right).$$

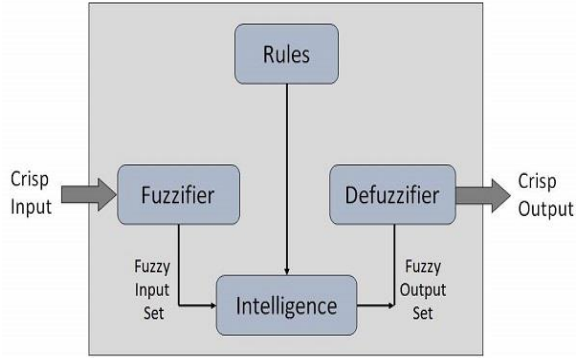
3) Çıkarım Mekanizması (Inference Engine)

Çıkarım mekanizması, giriş değerlerinin ve kuralların ne kadar eşleştiğini belirler. Kurallar, alınan giriş değerlerine göre uygulanır. Daha sonra kurallar, kontrol eylemlerini geliştirmek için kullanılır. Çıkarım mekanizması ve kabuller katmanı birlikte bir Bulanık Mantık sisteminde denetleyici olarak adlandırılır. Başka bir deyişle; çıkarım mekanizması, girdiler ve IF-THEN kuralları üzerinde bulanık çıkarımlar yaparak insan akıl yürütme sürecini simüle eder.

4) Durulaştırma (Defuzzification)

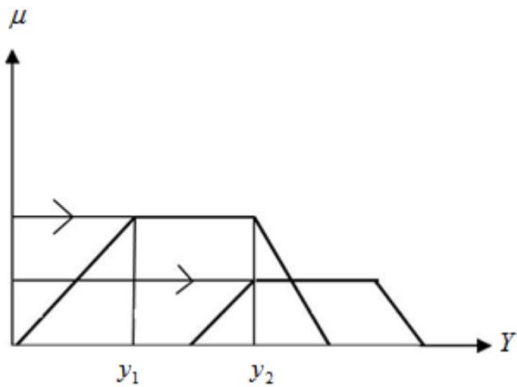
Bulanıklaştırmanın ters işlemidir. Burada bulanık değerler haritalanarak kesin değerlere dönüştürülür. Bu işlemi gerçekleştirmek için farklı durulaştırma yöntemleri olsa da girdiye göre en verimli olan seçilir. Bu, maksimum üyelik ilkesi (üyelik değerinin en yüksek olduğu noktaya karşılık gelen değer) ağırlıklı ortalama yöntemi ve merkezi nokta yöntemi gibi yöntemlerin kullanıldığı karmaşık bir süreçtir. Yapay zekadaki bulanık mantık, insan muhakemesini taklit etmeye yardımcı olsa da bu sistemlerin kurulabilmesi için uzman rehberliğine ihtiyacı vardır. Bu, sistemi daha iyi anlayan uzmanların deneyimine güvenmenizi sağlar. Bulanık

mantık algoritmalarının yürütülmesini geliştirmek için de kullanılabilir.



Durulaştırma İşleminde Kullanılan Yöntemler:

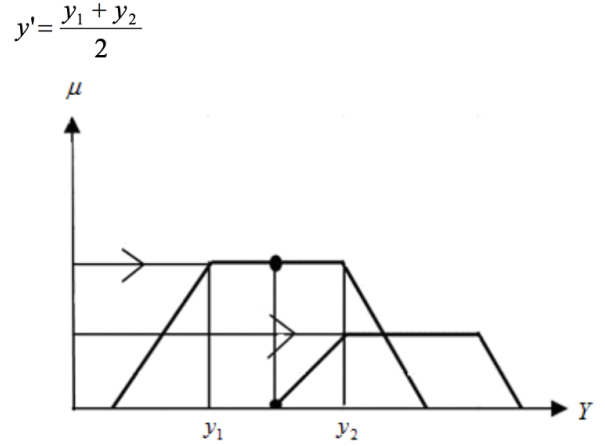
- a) Maksimum Metodu: Maksimum metodunda çıkışa ait keskin değerin elde edilmesi için sadece işlenen kurallar içinde en büyük yüksekliğe sahip (en büyük üyelik derecesine sahip) değer seçilir.



(Şekil 3.1)

Şekil 3.1 de görüleceği üzere burada çıkışa gönderilecek değer; $[y_1, y_2]$ kapalı aralığıdır.

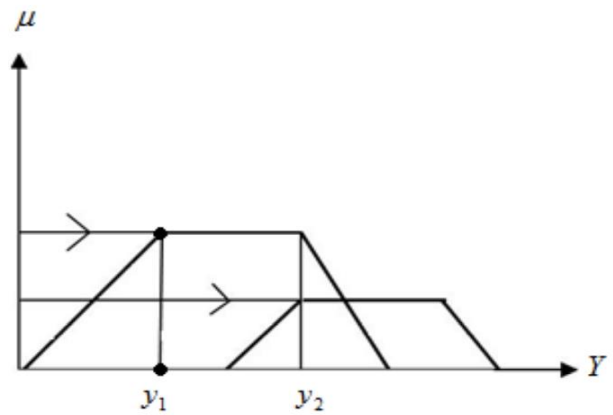
- a) Maksimum Ortası Metodu: Bu metod, çıkış değerini bulmak için; en büyük yüksekliğe sahip bulanık çıkış kümesinde maksimum yüksekliğin sınırlarını belirleyen $[y_1, y_2]$ sınır değerinin ortalamasını alır.



(Şekil 3.2)

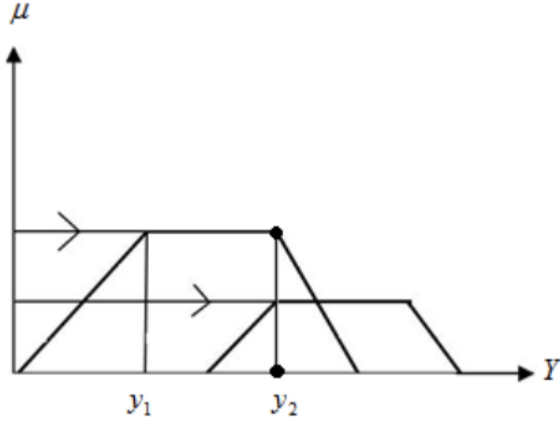
Şekil 3.2 de görüleceği üzere burada çıkışa gönderilecek değer; $[y_1, y_2]$ kapalı aralığının orta noktasıdır.

- b) Sol Kenar Noktası Metodu: Bu metotta çıkış keskin değeri olarak alt aralık sınır değeri seçilir.



(Şekil 3.3)

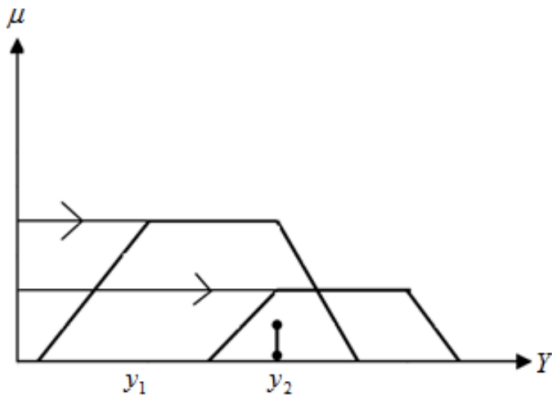
- c) Sağ Kenar Noktası Metodu: Bu metotta çıkış keskin değeri olarak üst aralık sınır değeri seçilir.



(Şekil 3.4)

- d) Ağırlık Merkezi Metodu: Bu metotta, çıkış keskin değeri olarak; tetiklenen kurallardan gelen üyelik değerlerinin altlarındaki alanların toplamının ağırlık merkezinin yatay eksen değeri olarak ele alınır

$$y' = \frac{\int y * \mu(y) * dy}{\int \mu(y) * dy}$$



(Şekil 3.5)

IV.UYGULAMALAR

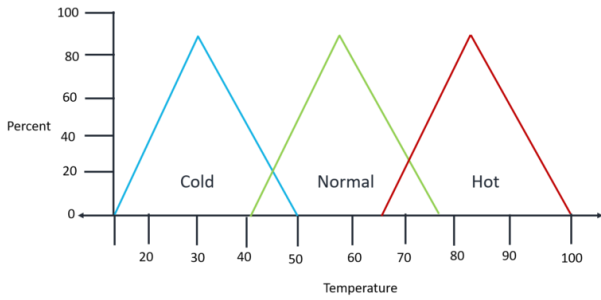
Bulanık mantık sisteminin tasarımı, her girdi ve çıktı için birer üyelik fonksiyonları dizisi ile başlar. Daha sonra kesin bir çıktı değeri elde etmek için üyelik fonksiyonlarına bir dizi kural uygulanır. Sırasıyla örneklere bakalım ve bulanık mantığın çalışma mekanizmasını anlayalım.

1) Klima Mekanizması

Bu örnekte klima sistemlerinde kullanılan mantık ele alınmaktadır. Eğer istenilen sıcaklık ile oda sıcaklığı arasında bir fark varsa, klima, sıcaklığı istenen değere ayarlamak için bulanık mantığı kullanır. Bu örnekteki amaç; girdimiz olan sıcaklık derecesine karşılık olarak, makinanın fan hızını ayarlamaktır.

Öncelikli olarak dilsel terimleri (veya değişkenleri) tanımlamalıyız. İkili mantık sisteminde sıcaklık sadece iki ana kategoriye ayrılabilir: sıcak ve soğuk. Bulanık mantık sisteminde ise farklı sıcaklık kategorileri bulunabilir ve bunları tanımlamak için dilsel terimler kullanabiliriz. Bu sistem özelinde kullanılan dilsel terimlerden bazıları çok soğuk, sıcak, çok sıcak, ılık, soğuk ve çok sıcaktır. Sıcaklık, yukarıda belirtilen dilsel terimlerden oluşan bir bulanık küme olarak adlandırılabilir.

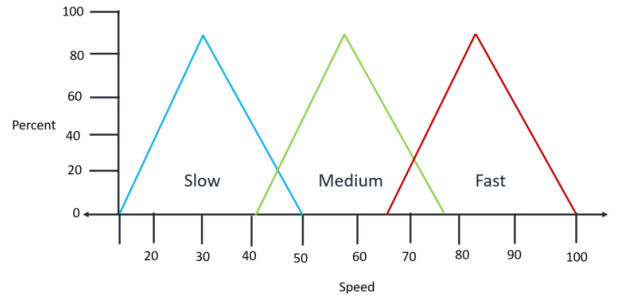
Burada sıcaklık giriř, fan hızı ıkıřtır. Her bir girdi iin bir dizi yelik fonksiyonu oluřturmamız gerekir. yelik fonksiyonu, bulanık deėiřken kmelerinin basit birer grafik temsilidir. İnceleyeceėimiz rnek iin; soėuk, sıcak ve ılık (normal) olmak zere  bulanık kme kullanacaėız. Daha sonra, bu  sıcaklık kmelerinin her biri iin birer yelik fonksiyonu oluřturacaėız.



(řekil 4.1)

Burada yelik deėerlerimiz; 0 ile 1 arasında deėiřmektedir ve gensel yelik fonksiyonu formuna uymaktadır. Soėuk' a ait yelik fonksiyonu; 0 ile 50 derece arasında tanımlanırken, ılık'a ait yelik fonksiyonu 40 ile 75 derece arasında tanımlanmakta, sıcak ise, 65 ile 100 derece arasında tanımlanmaktadır.

Bir sonraki adımda, ıktı iin; yavaş, orta ve hızlı olmak zere  bulanık kme kullanacaėız. Girdi dizisinde olduėu gibi burada da her bir ıktı dizisi iin yelik fonksiyonu yaratılır.

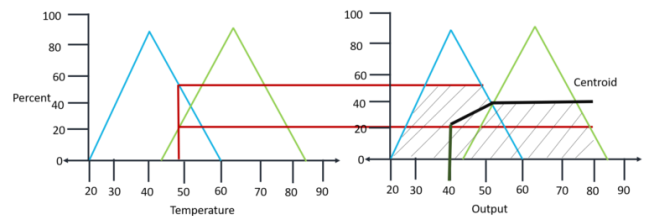


(řekil 4.2)

Burada yelik deėerlerimiz; 0 ile 1 arasında deėiřmektedir ve gensel yelik fonksiyonu formuna uymaktadır. Yavaş'a ait yelik fonksiyonu; 0 ile 50 hız deėeri arasında tanımlanırken, ortayaait yelik fonksiyonu 40 ile 75 derece arasında tanımlanmakta, hızlı ise, 65 ile 100 hız deėeri arasında tanımlanmaktadır.

yelik fonksiyonlarımız tanımlandıėına gre, yelik fonksiyonlarının son sisteme nasıl uygulanacaėını tanımlayacak kuralları oluřturabiliriz. Bu sistem iin  kural oluřturacaėız.

- Sıcak İSE Hızlı
- ılık İSE Orta
- Soėuk İSE Yavaş



(řekil 4.3)

Bu kurallar, sistemi yrtmek amacıyla; kesin ıktı deėeri retebilmek iin yelik

fonksiyonlarına uygulanır. Grafikten görüleceği üzere, 52 derecede üyelik fonksiyonları kesişir. Burada kesişim her iki fonksiyonda da gerçekleştiği için iki kuralı da uyguluyoruz. Daha sonra kesişim noktasının yüksekliğinde çıktı fonksiyonunu keseriz.

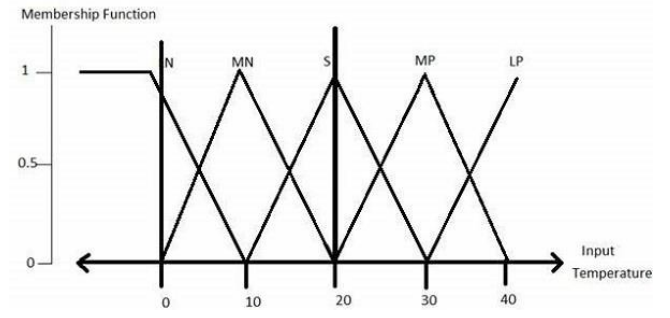
2) Sıcaklık Kontrolü

Bölüm 4, örnek 1 de anlatılan uygulamaya benzer olarak yeni bir örnek daha inceleyip, bulanık mantığın algoritmasını ve basamaklarını daha net anlayalım.

Örnek 1 de olduğu gibi klima sistemini ele alırsak; girdimiz yine ortamın sıcaklığı olacaktır, fakat bu sefer çıktımız fan hızı yerine olması istenilen(beklenen) ortam sıcaklığı olacaktır.

Birinci adım olarak, Bulanıklaştırma (Fuzzification) için, dilsel değişkenler tanımlanmalıdır. Bu örnek özelinde dilsel değişkenler kümemiz; Sıcaklık (t) = {çok soğuk, soğuk, ılık, çok sıcak, sıcak} 'dır. Bu kümenin her bir üyesi dilsel bir terimdir ve genel sıcaklık değerlerinin bir kısmını kapsar.

İkinci adım olarak dilsel değişkenlerin üyelik fonksiyonları oluşturulmalıdır. Sıcaklık değişkeninin üyelik fonksiyonu gösterildiği gibidir



(Şekil 4.4)

Üçüncü adım olarak; kabuller katmanı için (Rule Base) hedef/beklenen oda sıcaklığına karşı, odanın sıcaklığı değerlerini içeren matris oluşturulur.

RoomTemp. /Target	Very_Cold	Cold	Warm	Hot	Very_Hot
Very_Cold	No_Change	Heat	Heat	Heat	Heat
Cold	Cool	No_Change	Heat	Heat	Heat
Warm	Cool	Cool	No_Change	Heat	Heat
Hot	Cool	Cool	Cool	No_Change	Heat
Very_Hot	Cool	Cool	Cool	Cool	No_Change

(Şekil 4.5)

İlk sütun odanın mevcut sıcaklığını gösterirken; ilk satır odanın hedef sıcaklığını gösterir. Orta kısımda kalan 5x5 boyutundaki kesişim matrisi iseyapılması gereken sıcaklık değişim değişkenini ifade eder.

Üçüncü adımın hemen devamında if-then kuralları belirlenir.

Sr. No.	Condition	Action
1	IF temperature=(Cold OR Very_Cold) AND target=Warm THEN	Heat
2	IF temperature=(Hot OR Very_Hot) AND target=Warm THEN	Cool
3	IF (temperature=Warm) AND (target=Warm) THEN	No_Change

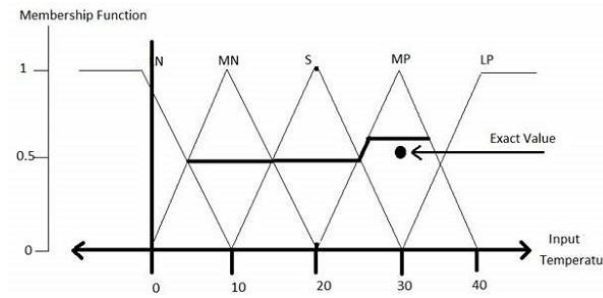
(Şekil 4.6)

1. EĞER sıcaklık soğuk VEYA çok soğuk İSE VE hedef odanın ılık olması İSE klima sıcak moda geçmelidir.
2. EĞER sıcaklık sıcak VE çok sıcak İSE VE hedef odanın ılık olması İSE klima soğuk moda geçmelidir.

3. EĞER sıcaklık ılık İSE VE hedef odanın ılık olması İSE klima modunda değişim yoktur.

Dördüncü adım olarak Çıkarım Mekanizması (Inference Engine) katmanı için; bulanık küme işlemleri yapılır. VEYA işlemi için MAX, VE işlemi için MIN işlemleri yapılarak kurallar değerlendirilir. Tam bir sonuç için tüm değerlendirme sonuçları birleştirilir. Ortaya çıkacak sonuç, bulanık bir değerdir.

Son olarak; Durulaştırma (Defuzzification) katmanı için; durulaştırma işlemi, çıkış değerinin üyelik fonksiyonuna göre yapılır ve net bir sonuç elde edilir.



(Şekil 4.7)

3) Bulanık Kontrol Sistemleri: Bahşiş Problemi

Bir restoranda vereceğimiz bahşiş miktarını seçmemize yardımcı olacak bir bulanık kontrol sistemi oluşturalım. Bahşiş verirken, yemek ve hizmet kalitesini 0 ile 10 arasında puanlayıp, bu planlamayı 0 ile %25 arasında bir bahşiş bırakmak için kullanırız.

Modellemeye problemi formülleştirerek başlayalım:

1.Girişler

1.1Hizmet Kalitesi

- Evren (net değer aralığı): 0 ile 10 arasında, bekleme personelinin hizmeti ne kadar iyiydi?
- Bulanık Küme (bulanık değer aralığı): Kötü, kabul edilebilir, şaşırtıcı

1.2Yemek Kalitesi

- Evren: 0 ile 10 arasında, yemek ne kadar lezzetliydi?
- Bulanık Küme: Kötü, idare eder, harika

2.Sonuçlar (Çıktılar)

2.1Bahşiş

- Evren: %0 ile %25 arasında ne kadar bahşiş bırakmalıyız?
- Bulanık Küme: Düşük, orta, yüksek

3.Kurallar

- Eğer hizmet kalitesi ya da yemek kalitesi iyi ise, bahşiş yüksek olmalı.
- Eğer hizmet kalitesi ortalama ise, bahşiş de ortalama olmalı.
- Eğer hizmet kalitesi ya da yemek kalitesi kötü ise, bahşiş düşük olmalı.

4.Kullanım

- Eğer kontrol mekanizmasına Hizmet için 9.8 ve kalite için 6,5 puan verirsek, ödememiz gereken bahşiş %20,2 olacaktır.

Skufuzzy Controller API Kullanarak Bahşis Kontrolcüsü Oluşturmak

Bu problemi modellerken skufuzzy controller API kullanacağız. İlk olarak kullanacağımız kütüphaneleri ekliyoruz, ardından bulanık değerleri tanımlayarak başlıyoruz. Numpy kütüphanesinden ‘arrange’ metodu ile değişkenlerimizin aralıklarını tanımlıyoruz. Daha sonrasında belirlediğimiz dilsel bahşis değerlerinin her biri için, üyelik fonksiyonları oluşturuyoruz. ‘trimf’ metodu, triangular membership function yani üçgen üyelik fonksiyonu anlamına gelmektedir. Fonksiyonun parametreleri üçgenin tanım aralığını, başlangıç ve bitiş noktalarını belirten birer listeden oluşmaktadır.

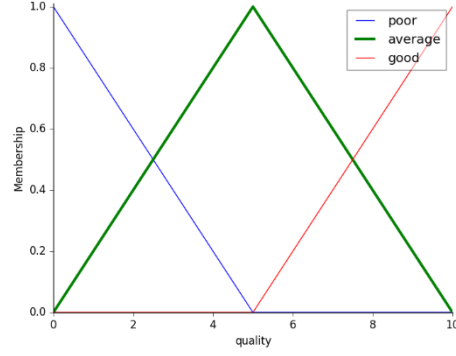
```
1 import numpy as np
2 import skfuzzy as fuzz
3 from skfuzzy import control as ctrl
4
5
6 quality = ctrl.Antecedent(np.arange(0, 11, 1), 'quality')
7 service = ctrl.Antecedent(np.arange(0, 11, 1), 'service')
8 tip = ctrl.Consequent(np.arange(0, 26, 1), 'tip')
9
10 quality.automf(3)
11 service.automf(3)
12
13 tip['low'] = fuzz.trimf(tip.universe, [0, 0, 13])
14 tip['medium'] = fuzz.trimf(tip.universe, [0, 13, 25])
15 tip['high'] = fuzz.trimf(tip.universe, [13, 25, 25])
```

(Şekil 4.8)

Üyeliğin nasıl görüldüğünü daha iyi anlayabilmek için ‘view’ komutunu kullanabiliriz.

```
quality['average'].view()
```

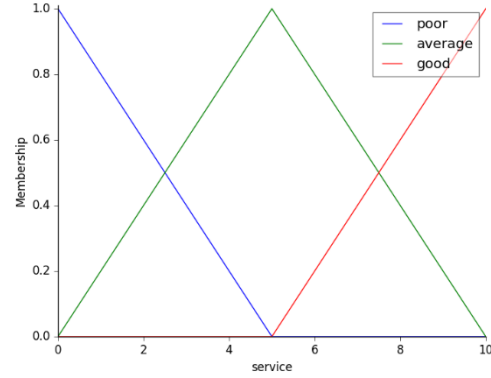
(Şekil 4.9)



(Şekil 4.10)

```
service.view()
```

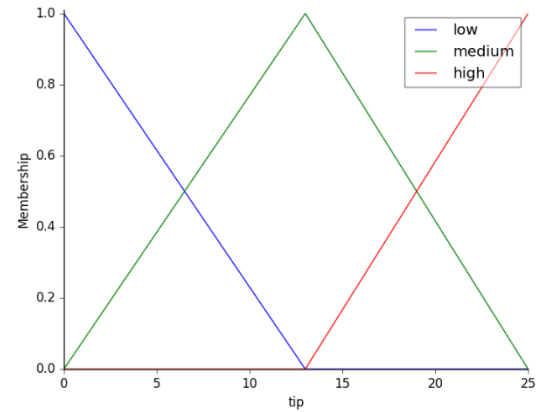
(Şekil 4.11)



(Şekil 4.12)

```
tip.view()
```

(Şekil 4.13)



(Şekil 4.14)

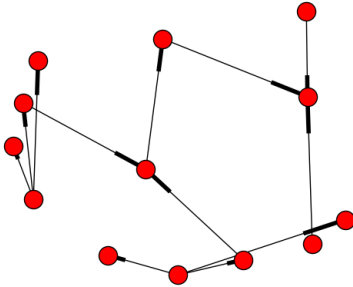
Bulanık Kurallar

Bu bölümde şekillerdeki üçgenleri kullanışlı hale getirebilmek için girdi ve çıktı değişkenleri arasındaki bulanık ilişkiyi, yukarıda belirlediğimiz kurallar çerçevesinde tanımlayacağız.

Çoğu insan belirlediğimiz kurallara katılsa da nihayetinde bu kurallar bulanıktırlar. Kesin olmayan kuralları tanımlanmış ve eyleme geçirilebilir bir ipucuyla eşleştirmek zordur. Bu, bulanık mantığın üstün olduğu türden bir görevdir.

```
rule1 = ctrl.Rule(quality['poor'] | service['poor'], tip[
rule2 = ctrl.Rule(service['average'], tip['medium'])
rule3 = ctrl.Rule(service['good'] | quality['good'], tip[
rule1.view()]
```

(Şekil 4.15)



(Şekil 4.16)

Kontrol Sisteminin Oluşturulması ve Simülasyonu

Kurallarımızı tanımladıktan sonra, basit bir kontrol sistemi oluşturuyoruz.

```
tipping_ctrl = ctrl.ControlSystem([rule1, rule2, rule3])
```

(Şekil 4.17)

Kontrol sistemini simüle etmek için ControlSystemSimulation oluşturuyoruz. Bu objenin, belirli koşullara uygulanan denetleyicimizi temsil ettiğini düşünelim. Bahşış için bu, yerel bir restoranda bahşış vermek olabilir. Eğer başka bir restoranda vereceğimiz bahşışı hesaplamak isteseydik, giriş değerlerimiz farklı olacağından yeni bir ControlSystemSimulation oluşturmak zorunda kalırdık.

```
tipping = ctrl.ControlSystemSimulation(tipping_ctrl)
```

(Şekil 4.18)

Artık girdileri belirleyerek ve hesaplama yöntemini çağırarak kontrol sistemimizi simüle edebiliriz. Kaliteye 10 üzerinden 6.5 ve hizmete 10 9,8 puan verdiğimiz varsayalım.

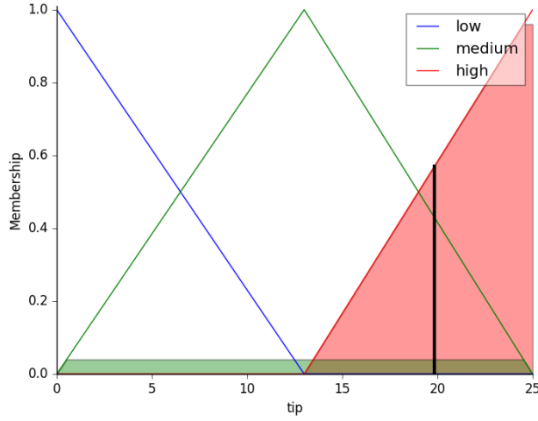
```
tipping.input['quality'] = 6.5
tipping.input['service'] = 9.8
tipping.compute()
```

(Şekil 4.19)

Hesaplama işlemi tamamlandıktan sonra durulaştırma işlemine geçiyoruz. 'tipping.output' komutu çıkış yani sonuç kümemizin son halidir. 'tip.view' komutu ile çıkış kümemizi grafiksel olarak görebiliyoruz.

```
print tipping.output['tip']
tip.view(sim=tipping)
```

(Şekil 4.20)



(Şekil 4.21)

Grafikten de görebileceğimiz gibi girdiğimiz değerlere karşılık gelen bahşiş %20.24'dür.

V.SONUÇ

Bu makalede, bulanık mantık sistemlerinin, oldukça geniş bir uygulama alanına sahip olan yapay zekâ konusunun neresinde yer aldığı anlatılmak istenmiştir. Paylaşılan bilgiler aslında klasik mantığın makine öğrenmesi, derin öğrenme ve uyarılara yapay zekanın yegâne ilhan kaynağı olan insan akli benzerliğinde tepki vermek konusunda yetersiz kaldığı vurgulanmaktadır. Bulanık mantık pek çok matematiksel çalışma alanına göre son derece yeni keşfedilmiş olmakla beraber günümüzde hala gelişmekte olan bir araştırma sahasıdır. Bu özelliği ve barındırdığı olasılıklar nedeniyle yeni gelişmelere her daim açık olması yapay zekanın amaçlarıyla bire-bir örtüşmektedir.

Bulanık mantık kullanımının yapay zekâ alanına vadettiklerinden kısaca bahsetmek

gerekirse öne çıkan avantajları şöyle sıralanabilir: Veri yetersizliği halinde bile sistemin devamlılığı, işgal edilen bellek alanının minimize edilmesi, kolay kurulan algoritmaları sebebiyle düşük maliyetli olması, insan muhakemesini esas aldığından amaca neredeyse kusursuz bir şekilde hizmet etmesi ve bu durumların beraberinde getireceği tüm tercih edilebilir nitelikler.

Ancak makalenin oluşturulması sırasında yapılan kaynak taramaları bazı dezavantajların bulanık mantığın tercih edilebilirliğini azalttığını göstermiştir. Bunlar ise şu şekilde listelenebilir: Verilerin güvenilirliğinin şüpheli olması, pek çok alternatif barındırdığından bir o kadar da çözüm yoluna sahip olması ve zaman zaman kafa karışıklığına yol açması, çok sayıda güncelleme gerektirmesi ve en önemlisi insan uzmanlığıyla yapılandırılmadıkları sürece sistemlerin işlevsiz olması.

Bulanık mantıkla yapay zekâ arasındaki derin ilişkiye literatürdeki bilgiler yardımıyla ışık tutmayı amaçlayan bu makale burada sonlansa da kısa sürede pek çok yeni ilişkinin kurulacağı öngörülmektedir.

KAYNAKÇA

[1]ARI, Mehmet Emin; *Stabilization of An Inverted Pendulum By Using A Fuzzy Controller*, A Master's Thesis, Middle East Technical University, 1995, Ankara

[2] ALAN, Atila Yusuf, *Nispi Mantik (Fuzzy Logic)*, International Seminar Group, Ludwigshaven, Germany (January 5, 2003)

[3] FREKSA, Christian; Kruse, Rudolf; de Mántaras, Ramon López; “*Interview with Prof. Lotfi A. Zadeh*, *Künstliche Intelligenz*, Band 15, Heft 3, 2001 40-43 Bibtex

[4] Nageh, M., El Ela, M. A., El Tayeb, E. S., & Sayyoub, H. (2015, September). *Application of using fuzzy logic as an artificial intelligence technique in the screening criteria of the EOR technologies*. In SPE North Africa Technical Conference and Exhibition. OnePetro.

[5] Guth, M. A. (1989). *Some uses and limitations of fuzzy logic in artificial intelligence reasoning for reactor control*. Nuclear Engineering and Design, 113(1), 99-109.

[6] Kazu, İ. Y., & Özdemir, O. (2009). *Öğrencilerin bireysel özelliklerinin yapay zeka ile belirlenmesi (Bulanık mantık örneği)*. Akademik Bilişim, 11-13.

[7.1] Pirovano, M. (2012). *The use of fuzzy logic for artificial intelligence in games*. University of Milano, Milano.

[7.2] KELEŞ, A. (2018). *DİJİTAL OYUNLARDA SONLU DURUM MAKİNELERİ VE BULANIK MANİK*. Electronic Turkish Studies, 13(29).

1) Artificial Intelligence- Fuzzy Logic Systems. (n.d.). Retrieved December 18, 2021, from https://www.tutorialspoint.com/artificial_intelligence/artificial_intelligence_fuzzy_logic_systems.htm

2) *What is fuzzy logic in AI and what are its applications?* Edureka. (2019, December 10). Retrieved December 18, 2021, from <https://www.edureka.co/blog/fuzzy-logic-ai/>

3) Ataşoğlu, A. (2020, May 25). *Python ile Bulanık Mantık Modellemesi*. Medium. Retrieved December 18, 2021, from <https://medium.com/deep-learning-turkiye/python-ile-bulanik-modellemesi-74459dc27308>

4) *Fuzzy Logic, Robotics and future of Artificial Intelligence*. Software Testing Help. (2021, November 29). Retrieved December 18, 2021, from <https://www.softwaretestinghelp.com/fuzzy-logic-robotics-in-ai/>

5) Puy, G. (2021, December 18). *Sosyal Hayatta iletişimin unsurları*. Ana Sayfa. Retrieved December 18, 2021, from <https://www.iienstitu.com/blog/bulanik-mantik-nedir-nasil-uygulanir>

6) *Fuzzy logic in artificial intelligence: Architecture, applications, Advantages & Disadvantages*. upGrad blog. (2021, December 14). Retrieved December 18, 2021, from <https://www.upgrad.com/blog/fuzzy-login-in-artificial-intelligence/>

7) *Fuzzification*. Fuzzification - an overview | ScienceDirect Topics. (n.d.). Retrieved December 18, 2021, from <https://www.sciencedirect.com/topics/engineering/fuzzification>

8) Rizvi, S., Mitchell, J., Razaque, A., Rizvi, M. R., & Williams, I. (2020, July 29). *A fuzzy inference system (FIS) to evaluate the security readiness of Cloud Service Providers - Journal of Cloud Computing*. SpringerOpen. Retrieved December 18, 2021, from <https://journalofcloudcomputing.springeropen.com/articles/10.1186/s13677-020-00192-9>

9) *What is computational intelligence?* IEEE Computational Intelligence Society. (n.d.). Retrieved December 18, 2021, from <https://cis.ieee.org/about/what-is-ci>

10) *Neural networks*. IBM. (n.d.). Retrieved December 18, 2021, from <https://www.ibm.com/topics/neural-networks>

11) By: IBM Cloud Education. (n.d.). *What is machine learning?* IBM. Retrieved December 18, 2021, from <https://www.ibm.com/cloud/learn/machine-learning>

12) *What is Artificial Intelligence (AI)?* IBM. (n.d.). Retrieved December 18, 2021, from https://www.ibm.com/topics/artificial-intelligence?mhsrc=ibmsearch_a&mhq=artificial+intelligence

13) Özdemir, O., & Kalınkara, Y. (2020). Bulanık Mantık: 2000-2020 yılları Arası Tez ve makale çalışmalarına Yönelik Bir içerik analizi. *Acta INFOLOGICA*, 4(2). <https://doi.org/10.26650/acin.762872>

14) *Iksadyayinevi.com*. (n.d.). Retrieved December 18, 2021, from <https://iksadyayinevi.com/wp-content/uploads/2020/02/BULANIK-MANTIK-Y%C3%96NTEM%C4%B0-VE-UYGULAMALARI.pdf>

15) *IV. Ünite: SembolİK Mantık: D-felsefedersligi.com*. (n.d.). Retrieved December 18, 2021, from <http://www.felsefedersligi.com/FileUpload/op30412/File/cokdegerliman.pdf>

16) *Different types of membership functions*. (n.d.). Retrieved December 18, 2021, from https://www.philadelphia.edu.jo/academics/qhamarshah/uploads/Lecture%2018_Different%20Types%20of%20Membership%20Functions%201.pdf